



T.C.
EGE ÜNİVERSİTESİ
Eğitim Bilimleri Enstitüsü



**METİN TABANLI PROGRAMLAMA ÖĞRETİMİNDE
METAFOR KULLANIMININ ETKİLERİ: BİR
KARMA YÖNTEM ARAŞTIRMASI**

Doktora Tezi

ERKAN GEÇİTLİ

EĞİTİM BİLİMLERİ ANABİLİM DALI

İzmir
Haziran, 2024

T.C.
EGE ÜNİVERSİTESİ
Eğitim Bilimleri Enstitüsü

METİN TABANLI PROGRAMLAMA ÖĞRETİMİNDE METAFOR
KULLANIMININ ETKİLERİ: BİR KARMA YÖNTEM
ARAŞTIRMASI

Doktora Tezi

Erkan GEÇİTLİ

Tez Danışmanı

Doç. Dr. Bünyamin YURDAKUL

Jüri Üyeleri

Prof. Dr. Gülsen ÜNVER

Prof. Dr. Bahar BARAN

Prof. Dr. Alper BAŞBAY

Prof. Dr. İlke Evin GENCEL

Eğitim Bilimleri Anabilim Dalı
Eğitim Programları ve Öğretim Doktora Programı

İzmir
Haziran, 2024

EGE ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ
ETİK KURALLARA UYGUNLUK BEYANI

Ege Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum “*Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması*” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversitede veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

Erkan GEÇİTLİ



EGE ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ

Tez Savunma Sınavı Jüri Tutanağı

006 – Enstitü Yönetim Kurulu’nun 05/05/2023 tarihli – 16/17 sayılı kararı



Öğrencinin	
Adı Soyadı	Erkan GEÇİTLİ
Numarası	94180000002
Anabilim Dalı	Eğitim Bilimleri
Programı	Eğitim Programları ve Öğretim - Yüksek Lisans Derecesi ile Doktora Programı
Derecesi	☐ Yüksek Lisans ☒ Doktora

Tezin	
Türkçe Başlığı	Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması
İngilizce Başlığı	Effects of Using Metaphors in Text-Based Programming Instruction: A Mixed Method Study
Değişiklik Varsa*	
Türkçe Başlığı	
İngilizce Başlığı	
Tez Danışmanı	Doç. Dr. Bünyamin YURDAKUL

Tez Savunma Sınavının	
Tarihi	27/06/2024
Saati	13.30
Yapılış Biçimi	☒ Yüz Yüze ☐ Çevrimiçi (Enstitü Yönetim Kurulu’nun 30/05/2024 tarih ve 20/3 sayılı kararı)

Ön Karar	
Ege Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliğinin ilgili maddeleri uyarınca; Jüri üyeleri tarafından hazırlanan ve ekte sunulan kişisel raporlar, intihal yazılım programında belirlenen “Benzerlik Kapsamlı Raporu” incelenmiş, ayrıntılı şekilde tartışılmış ve adayın tez savunma sınavına alınıp alınmayacağı ile ilgili olarak oybirliği / oyçokluğu ile aşağıdaki karar alınmıştır.	
☒	Savunma sınavına girmesi uygundur.
☐	Savunma sınavına girmesi <u>uygun değildir</u> .
☐	İntihal vardır.
Yukarıdaki kararınız “İntihal vardır” şeklinde ise, Ege Üniversitesi Lisansüstü Eğitim-Öğretim Yönetmeliği’nin Madde 8’in 4. bendinde belirtilen; “Rapordaki verilerde gerçek bir intihalin tespiti halinde gerekçesi ile birlikte karar verilmek üzere tez enstitü yönetim kuruluna gönderilir.” maddesi gereği “Son Kararınız” olarak değerlendirileceğinden aşağıdaki “Son Karar” alanında bir işaretlemede bulunmayınız. Değerli jüri üyemiz, kararınızın gerekçelerini yazmanızı rica ederiz.	

Son Karar	
Yukarıda verilen ön kararın “tezin savunulabileceği” yönünde olması itibarıyla aday tez savunma sınavına alınmış ve sonuçta tezle ilgili olarak oybirliği / oyçokluğu ile aşağıdaki nihai karar alınmıştır:	
☒	Başarılıdır (Kabul)
☐	Başarısızdır (Ret)
☐	Düzeltilmelidir (Tezli yüksek lisans için en çok 3 ay, Doktora için en çok 6 düzeltme verilebilir.)
Değerli jüri üyelerimiz kararınız özellikle “Başarısızdır (Ret)” ya da “Düzeltilmelidir” şeklinde ise, bu kutucuğa kararınızın alınma biçiminin ayrıntısı ile gerekçelerini yazmanızı rica ederiz.	

Jüri Başkanı**	Doç. Dr. Bünyamin YURDAKUL (Danışman)	İmza
Üniversite	Ege Üniversitesi	
Anabilim / Bilim Dalı	Eğitim Bilimleri / Eğitim Programları ve Öğretim	
Jüri Üyesi	Prof. Dr. Gülsen ÜNVER	İmza
Üniversite	Ege Üniversitesi	
Anabilim / Bilim Dalı	Eğitim Bilimleri / Eğitim Programları ve Öğretim	
Jüri Üyesi	Prof. Dr. Bahar BARAN	İmza
Üniversite	Dokuz Eylül Üniversitesi	
Anabilim / Bilim Dalı	Bilgisayar ve Öğretim Teknolojileri Eğitimi	
Jüri Üyesi	Prof. Dr. Alper BAŞBAY	İmza
Üniversite	Ege Üniversitesi	
Anabilim / Bilim Dalı	Eğitim Bilimleri / Eğitim Programları ve Öğretim	
Jüri Üyesi	Prof. Dr. İlke Evin GENÇEL	İmza
Üniversite	İzmir Demokrasi Üniversitesi	
Anabilim / Bilim Dalı	Eğitim Bilimleri / Eğitim Programları ve Öğretim	

* Savunma sınavı sonunda, tez başlığında bir değişikliğe karar verilmesi durumunda tezin Türkçe ve İngilizce başlıkları yazılmalıdır.

** Lisansüstü tez savunma sınavlarında danışman öğretim üyesi Jüri Başkanı olabileceği gibi, jürinin kendi arasında seçeceği bir üye de jüri başkanı olabilir.

• Bu tutanağa jüri üyelerinin kişisel raporları da eklenerek üç (3) işgünü içerisinde Anabilim Dalı Başkanlığı üst yazısıyla Enstitü Müdürlüğüne gönderilmelidir.

ÖNSÖZ

Metin tabanlı programlama öğretiminde metafor kullanımının etkilerini karma yöntem aracılığıyla incelemek öğretmen kimliğimle uygulamaya yönelik deneyimlerimde, doktora öğrencisi kimliğimle araştırma becerilerimin gelişiminde oldukça katkısı oldu. Bu süreçte yaşadığım her zorlukta ulaşabildiğim, sorunlarımın çözümü için yaratıcı çözüm önerileri sunan ve desteğini her an hissettiğim yüksek lisans ve doktora danışmanım değerli hocam Doç. Dr. Bünyamin Yurdakul'a sonsuz teşekkürlerimi sunuyorum.

Araştırmalarımda anlatmak istediklerimi uzun ve karmaşık cümleler kurmadan daha sade, daha etkili ve daha bilimsel bir dille anlatabilmem için bana örnek olan, yoğun çalışma zamanlarında bile zorluklarımı aşmamda bana vakit ayırmaktan çekinmeyen değerli hocam Prof. Dr. Gülsen Ünver'e, Tez İzleme Komitesinde değerli görüşleriyle tezimin şekillenmesinde yardımcı olan değerli hocam Prof. Dr. Bahar Baran'a ve Doç. Dr. Elif Buğra Kuzu Demir'e şükranlarımı sunarım.

Araştırmalarım sırasında görüşlerine ihtiyaç duyduğum uzmanlara, denel işlemim sırasında bana yardımdan çekinmeyen Bilişim Teknolojileri alanında öğretmenlik yapan arkadaşlarıma, araştırmaya katılmayı kabul eden tüm öğrencilere ve okullarda araştırma yapmam için izin veren okul müdürlerine ve Millî Eğitim Bakanlığına teşekkür ederim.

Ege Üniversitesi, Eğitim Programları ve Öğretim bilim dalında görev yapan tüm hocalarıma bu araştırmayı yürütebilmem için gerekli yeterlikleri kazanmamı sağladıkları için ayrıca teşekkür ederim. Araştırma verilerimin analizindeki desteklerinden dolayı Eğitimde Ölçme ve Değerlendirme bilim dalından Prof. Dr. Tahsin Oğuz Başokçu'ya ve Prof. Dr. Tuncay Öğretmen'e teşekkürlerimi sunarım.

Çalışmalarım sırasında yanımda olan ve beni destekleyen sevgili eşime, anneme ve her daim gülüşüyle çalışmam için gerekli enerjiyi sağlayan kızım Masal'a teşekkür ederim.

Erkan GEÇİTLİ



*Gökkuşağı kadar renkli, kuşlar kadar özgür
sevgili kızım Masal'a...*

İÇİNDEKİLER

ETİK KURALLARA UYGUNLUK BEYANI	iii
TEZ SAVUNMA SINAVI JÜRİ TUTANAĞI	iv
ÖNSÖZ.....	v
İÇİNDEKİLER	vii
TABLolar LİSTESİ.....	x
ŞEKİLLER LİSTESİ.....	xii
ÖZ.....	xiii
ABSTRACT	xv
I. BÖLÜM.....	1
GİRİŞ	1
1.1. Problem Durumu	1
1.2. Araştırmanın Amacı ve Önemi.....	9
1.3. Problem Cümlesi ve Alt Problemler	11
1.4. Sayıtlılar	12
1.5. Sınırlılıklar.....	13
1.6. Tanımlar	13
II. BÖLÜM	14
KURAMSAL ALANYAZIN VE İLGİLİ ARAŞTIRMALAR	14
2.1. Programlama Öğretimi	14
2.1.1. Programlama Öğretimi Yaklaşımları.....	16
2.1.2. Programlama Becerilerinin Ölçülmesi.....	22
2.1.3. Türkiye’de Programlama Öğretimi.....	27
2.2. Metafor ve Eğitimde Kullanımı	29
2.2.1. Metafor	29
2.2.2. Kavramsal Metafor Kuramı	31
2.2.3. Eğitimde Metafor Kullanımı.....	33
2.3. İlgili Araştırmalar	35
2.3.1. Programlama Kavramlarının Öğretiminde Metafor Kullanımı	35
2.3.2. Bilgi işlemsel Düşünme Becerisini Geliştirmede Metafor Kullanımı	39
2.3.3. Programlama Becerilerini Geliştirmede Metafor Kullanımı	41
III. BÖLÜM.....	45
YÖNTEM.....	45
3.1. Araştırmanın Deseni.....	45
3.2. Çalışma Grubu.....	46

3.2.1. Grupların Denkliğine İlişkin Bilgiler.....	47
3.3. Denel İşlem.....	49
3.3.1. Denel İşlem Materyali	49
3.3.2. Denel İşlem Ortamı.....	52
3.3.3. Kontrol Grubu Öğrenme-Öğretme Süreci	53
3.4. Veri Toplama Araçları.....	54
3.4.1. Nicel Veri Toplama Araçları	55
3.4.2. Nitel Veri Toplama Araçları.....	60
3.5. Veri Toplama Süreci	62
3.6. Verilerin Çözümlemesi ve Yorumlanması	66
3.6.1. Nicel Verilerin Çözümlemesi ve Yorumlanması.....	66
3.6.2. Nitel Verilerin Çözümlemesi ve Yorumlanması.....	67
3.6.2.1. Verileri Yönetme ve Düzenleme	67
3.6.2.2. Ortaya Çıkan Fikirleri Okuma ve Not Etme	69
3.6.2.3. Kodları Tanımlama ve Temalar Halinde Sınıflandırma	69
3.6.2.4. Yorumları Geliştirme ve Değerlendirme	69
3.6.2.5. Verileri Temsil Etme ve Görselleştirme	70
3.7. Geçerlik ve Güvenirlik	71
3.7.1. Geçerlik	71
3.7.2. Güvenirlik.....	77
3.8. Araştırmacının Rolü ve Etik.....	78
IV. BÖLÜM	80
BULGULAR.....	80
4.1. Programlama Kavramlarını Öğrenme	80
4.1.1. Deney ve Kontrol Grubu Karşılaştırmaları.....	80
4.1.2. Programlama Kavramlarını Öğrenmeye Yönelik Nitel Bulgular	85
4.1.2.1. Kavramlar	85
4.1.2.2. Öğrenme Süreci	88
4.1.3. Programlama Kavramlarını Öğrenme Sürecine Ait Nicel ve Nitel Kanıtların İlişkisi	91
4.2. Programlama Becerileri.....	92
4.2.1. Deney ve Kontrol Grubu Karşılaştırmaları.....	93
4.2.2. Programlama Becerilerine Yönelik Nitel Bulgular	94
4.2.2.1. Öğretim Süreci.....	96
4.2.2.2. Programlamaya Yönelik Tutum.....	103
4.2.3. Programlama Becerisi Açısından Nitel ve Nicel Kanıtların İlişkisi	103

4.3. Bilgi İşlemsel Düşünme Öz Yeterliği	105
4.3.1. Deney ve Kontrol Grubu Karşılaştırmaları.....	106
4.3.2. Bilgi işlemsel Düşünme Öz Yeterliğine Yönelik Nitel Bulgular.....	109
4.3.2.1. Bilgi işlemsel Düşünme Bileşenleri.....	109
4.3.2.2. Programlamayla İlişki.....	110
4.3.3. Bilgi İşlemsel Düşünme Öz Yeterliği Açısından Nitel ve Nicel Kanıtların İlişkisi	111
V. BÖLÜM.....	113
SONUÇ, TARTIŞMA VE ÖNERİLER	113
5.1. Sonuç ve Tartışma	113
5.1.1. Metafora Dayalı Öğretimin Programlama Kavramlarını Öğrenmeye Etkisi ...	113
5.1.2. Metafora Dayalı Öğretimin Programlamaya Etkisi	116
5.1.3. Metafora Dayalı Öğretimin Bilgi İşlemsel Düşünmeye Etkisi	119
5.2. Öneriler.....	121
5.2.1. Programlama Öğretimine Yönelik Öneriler	121
5.2.2. İleride Yapılacak Araştırmalar için Öneriler	123
KAYNAKÇA	126
EKLER.....	149
Ek-1. “Pastadan Programa” Ders Planı	149
Ek-2. Kavram Haritası Geliştirme ve Değerlendirme (KHGD) Formu	155
Ek-3. Şifrelenmiş Mesajlar Oluşturma Uygulaması.....	158
Ek-4. Elektronik Ürün Geliştirme Süreci Puanlama Yönergesi.....	159
Ek-5. Denel İşlem Öncesi Öğretmen Görüşme Formu	161
Ek-6. Denel İşlem Sonrası Öğretmen Görüşme Formu	163
Ek-7. Denel İşlem Sonrası Öğrenci Görüşme Formu.....	165
Ek-8. Gözlem Formu.....	167
Ek-9. Süreç Değerlendirme Formu	170
Ek-10. İl Milli Eğitim Müdürlüğünden Alınan Araştırma İzni	171
Ek-11. Etik Kurul Onayı	172
Ek-12. Araştırmanın 2021-2022 Eğitim-Öğretim Yılında da Uygulanabileceğine Dair İl Milli Eğitim Müdürlüğü İzni	173
Ek-13. Araştırmada Gerçekleşen Değişikliklere Yönelik Etik Kurul Onayı	174

TABLÖLAR LİSTESİ

Tablo 1. Deney ve Kontrol Gruplarında Yer alan Öğrenci Sayıları ve Şubelere Göre Dağılımı	46
Tablo 2. KHGD Formu Ön test Puanlarına Göre Deney ve Kontrol Grubu Giriş Özellikleri.....	48
Tablo 3. BİDÖ Ölçeği Ön Test Puanları Deney ve Kontrol Grubu Giriş Özellikleri	48
Tablo 4. Denel İşlem Materyalini Oluşturan Ders Planlarına Ait Bilgiler	51
Tablo 5. KHGD Formu Puanlayıcılar Arası Uyum Düzeyini Gösteren Kappa İstatistiği Sonuçları	56
Tablo 6. EÜGS Puanlama Yönergesi Puanlayıcılar Arası Uyum Düzeyini Gösteren Kappa İstatistiği Sonuçları	59
Tablo 7. Programlama Ünitesinde Yer Alan Kavramları Belirleme Formu Aracılığıyla Alan Uzmanları Tarafından Belirlenen Kavramların Listesi	63
Tablo 8. Karma Yöntem Deneyisel Desen Geçerlik Tehditleri ve Tehditleri En az İndirme Stratejileri (Creswell ve Plano Clark, 2018).....	71
Tablo 9. Deney ve Kontrol Grubunun Programlamanın Temelleri Ortalama ve Standart Sapma Değerleri.....	80
Tablo 10. Deney ve Kontrol Grubunun Program Kontrolüne İlişkin Ortalama ve Standart Sapma Değerleri	81
Tablo 11. Program Kontrolü Ön test-Son test Puanları ANOVA Sonuçları	82
Tablo 12. Deney ve Kontrol Grubunun Veri Yapılarına İlişkin Ortalama ve Standart Sapma Değerleri.....	83
Tablo 13. Veri Yapıları Ön test-Son test Puanları ANOVA Sonuçları.....	83
Tablo 14. Deney ve Kontrol Grubu Dosya İşlemlerine İlişkin Ortalama ve Standart Sapma Değerleri.....	84
Tablo 15. KHGD Formu Dosya İşlemleri Ön test ve Son test Puanları ANOVA Sonuçları	84
Tablo 16. Deney ve Kontrol Grubu Algoritma ve Akış Şeması Alt Performans Görevi Puanlarının Mann-Whitney U testi Sonuçları	93
Tablo 17. Programlama Becerilerine Yönelik Deney ve Kontrol Grubu Puanlarının t Testi Sonuçları	94

Tablo 18. Deney ve Kontrol Grubu BİDÖ Ölçeği Problem Çözme Faktörü Ortalama ve Standart Sapma Değerleri.....	106
Tablo 19. BİDÖ Ölçeği Problem Çözme Faktörü Ön test ve Son test Puanları ANOVA Sonuçları	107
Tablo 20. Deney ve Kontrol Grubu BİDÖ Ölçeği Eleştirel Düşünme ve İşbirlikli Öğrenme Faktörü Ortalama ve Standart Sapma Değerleri	107
Tablo 21. Deney ve Kontrol Grubu BİDÖ Farklı Düşünme Faktörü Ortalama ve Standart Sapma Değerleri	108
Tablo 22. Deney ve Kontrol Grubu BİDÖ Ölçeği Algoritmik Düşünme Faktörü Ortalama ve Standart Sapma Değerleri.....	108



ŞEKİLLER LİSTESİ

Şekil 1. Araştırmada kullanılan karma yöntem deneysel deseni.....	45
Şekil 2. Deney ve kontrol grubu bilişim teknolojileri sınıflarının yerleşim düzeni...	53
Şekil 3. Veri toplama süreci	65
Şekil 4. Rampin ve Rampin (2021) tarafından geliştirilen yazılım aracılığıyla verilerin analizine ve düzenlenmesine bir kesit.....	68
Şekil 5. Verilerin görselleştirilmesine yönelik bir kesit.....	70
Şekil 6. Kavram öğrenme ile ilişkili tema, alt tema ve kodlar	86
Şekil 7. Programlama becerileri ile ilişkilendirilen tema, alt tema ve kodlar	95
Şekil 8. Bilgi işlemsel düşünme öz yeterliği kategorisini oluşturan tema, alt tema ve kodlar.....	109

ÖZ

Bu araştırmanın temel amacı, kavramsal metafor kuramına dayalı metin tabanlı programlama öğretiminin öğrencilerin programlama kavramlarını kazanmaya, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliklerine etkilerini incelemektir.

Karma yöntem deneysel deseni ile yürütülen araştırma, metin tabanlı programlama ile ilk kez karşılaşan 9. sınıf öğrencileri ve öğretmenleriyle yürütülmüştür. İki anadolu lisesinin birinde yer alan üç şube deney grubu ($n=66$) diğer lisenin üç şubesi ise ($n=69$) kontrol grubu olarak seçkisiz atanmıştır. Deney grubunda 18 hafta boyunca metafora dayalı öğretim gerçekleştirilmiş, kontrol grubu şubelerine öğretim süreçlerinde herhangi bir müdahalede bulunulmamıştır. Programlama kavramlarını kazanma durumu grupların deney öncesi ve sonrası hazırladıkları kavram haritalarının analitik bir puanlama yönergesi kullanılarak, bilgi işlemsel düşünme öz yeterliği ise bir ölçek yardımıyla nicel olarak belirlenmiştir. Grupların programlama becerileri, geliştirilen bir senaryo temelinde süreçte hazırlanan elektronik ürünlerin deney sonunda analitik puanlama yönergesi aracılığıyla nicel değerlendirilmiştir. Yanı sıra deney öncesinde deney ve kontrol grubu öğretmenleriyle görüşmeler, deney sırasında gözlemler ve öğrencilerin süreç değerlendirme formları, deney sonunda da grupların öğretmen ve öğrencileriyle görüşmeler ile nitel veri toplanmıştır. Nicel veriler için karışık ölçümler için ANOVA, ilişkisiz ölçümlerde t testi ve Mann-Whitney U testi kullanılmıştır. Nitel veriler için içerik analizi uygulanmıştır.

Araştırma sonuçları, metafora dayalı öğretiminin metin tabanlı programlamada yer alan karar kontrol yapıları, veri yapıları ve dosya işlemlerine yönelik anlaşılması zor ve soyut kavramları öğretmede etkili olduğunu, programlamaya giriş kavramlarının öğretiminde ise fark yaratmadığını göstermiştir. Metafora dayalı öğretim, bilgisayar bilimi dersini eğlenceli ve anlaşılır hale getirerek derse ilgiyi arttırmış, kod yazmayı kolaylaştırarak dersten erken kopmalarının önüne geçmiştir. Öğrencilerin programlama becerilerine katkı sağlayarak elektronik ürün geliştirmelerine yardımcı olmuştur. Metafora dayalı öğretim, öğrencilerin bilgi işlemsel düşünmenin problem çözme bileşenine yönelik öz yeterliklerinde olumlu değişim yaratırken, eleştirel düşünme ve işbirlikli öğrenme, farklı düşünme ile

algoritmik düşünme boyutlarında anlamlı bir fark oluşturmamıştır. Araştırmada, metafora dayalı öğretimin öğrencilerin kavram oluşturmalarında ve kazanmalarında, onların programlama becerilerini ve bilgi işlemsel düşünme öz yeterliklerini geliştirmede etkili bir bilişsel araç olarak kullanılabileceği önerilmektedir.

Anahtar Kelimeler: Bilgi işlemsel düşünme öz yeterliği, kavram kazanma, kavramsal metafor, programlama becerileri, programlama öğretimi



ABSTRACT

The main purpose of this study is to examine the effects of text-based programming instruction on students' acquisition of programming concepts, programming skills, and computational thinking self-efficacy using conceptual metaphor theory.

The study, which was conducted using a mixed-methods experimental design, was conducted with 9th grade students and teachers who were experiencing text-based programming for the first time. Three classes in one of two anatolian high schools were randomly assigned as the experimental group ($n=66$) and three classes in the other high school ($n=69$) as the control group. The experimental group received 18 weeks of metaphor-based instruction, while the control group classes received no intervention during the instructional process. Acquisition of programming concepts was determined quantitatively by scoring the concept maps prepared by the groups before and after the treatment using an analytic rubric, and computational thinking self-efficacy was measured quantitatively using a scale. The groups' programming skills were evaluated quantitatively by means of an analytic rubric at the end of the treatment on the electronic products produced in the process, based on a developed scenario. In addition, qualitative data were collected through interviews with the teachers of the groups before the treatments, observations and students' process evaluation forms during the treatments, and interviews with the teachers and students of the groups at the end of the treatments. For quantitative data, ANOVA for mixed measures, independent-samples t-test and Mann-Whitney U test were used. Content analysis was used for the qualitative data.

The results of the study showed that metaphor-based instruction was effective in teaching difficult-to-understand and abstract concepts such as decision control structures, data structures and file operations in text-based programming, while it made no difference in teaching introductory programming concepts. Metaphor-based teaching increased interest in the course by making the computer science course fun and understandable, and prevented early withdrawal from the course by making it easier to write code. It helped students to develop electronic products by contributing to their programming skills. While metaphor-based instruction created a positive

change in the problem solving self-efficacy of computational thinking, it did not create a significant difference in the dimensions of critical thinking and collaborative learning, divergent thinking and algorithmic thinking. The research suggests that metaphor-based instruction can be used as an effective cognitive tool in students' concept formation and acquisition, in developing their programming skills and computational thinking self-efficacy.

Key words: Computational thinking self-efficacy, concept acquisition, conceptual metaphor, programming skills, teaching programming



I. BÖLÜM

GİRİŞ

Araştırmanın bu bölümünde problem durumu, araştırmanın amacı ve önemi, problem cümlesi ve alt problemler, sınırlılıklar ve tanımlar yer almaktadır.

1.1. Problem Durumu

Teknolojik gelişmeler toplumda değişim ve dönüşümlere neden olarak toplumu oluşturan bireylerin edinmeleri beklenen niteliklerin yeniden gözden geçirilmesini gerekli kılmaktadır. Avrupa Birliği Dijital Eğitim Eylem Planı (2021-2027) dijital eğitim ekosistemini desteklemek ve paydaşların dijital beceri ve yeterliklerini geliştirmek için eylem planları sunarak üye devletleri dijital çağa taşımayı amaçlamaktadır (European Commission, 2020). Uluslararası Eğitimde Teknoloji Topluluğu (International Society for Technology in Education [ISTE], 2019) bireylerin bilgiye ulaşmada teknolojiyi etkin kullanmalarının, stratejik bir yol izleyerek sorunlara teknolojik çözümler üretebilmelerinin ve bütün bunları işbirlikli olarak gerçekleştirebilmelerinin önemine vurgu yapmaktadır. Kalelioğlu'na (2015) göre öğretmenlerin, yöneticilerin ve diğer paydaşların belirtilen nitelikleri kazanmaları eğitim sisteminin bilgisayar bilimine yeterince önem vermesiyle mümkün olabilir. Böylelikle bilgisayar bilimi aracılığıyla toplumda meydana gelen değişim ve dönüşümlere hazırlıklı dijital vatandaşlar yetiştirilebilir.

Bilgisayar bilimi, dünyanın farklı yerlerinde bilişim teknolojileri, bilgi işlem ve dijital teknolojiler gibi isimlerle anılmakta, çoğu zaman ayrı bir eğitim programı olarak eğitim sistemlerinde yerini almaktadır (Fluck ve diğ., 2016). Bilgisayar bilimi, bilgisayar pedagojisini içeren (Passey, 2017) ve bilgisayar alanının temelini oluşturan akademik bir disiplindir (Webb ve diğ., 2017). İlk kez 1953'te Cambridge'de ders olarak yer almış, 1962'de de Purdue Üniversitesinde bölüm olarak açılmıştır. Bugüne kadar yenilenen teknolojiler ve sürekli müzakere gerektiren içeriği nedeniyle eğitim programları alanında tartışmalara neden olmuştur (Fluck ve diğ., 2016). Eğitim programlarının teknolojik gelişmelere uyum sağlaması için çok sık güncellenmesi içeriğin güncel tutulmasına katkı sağlamakta fakat öğretmenlerin program hakkındaki mesleki gelişim çalışmaları, programa yönelik araştırmalar ve programın uzun vadede değerlendirilmesi açılarından birtakım zorlukları beraberinde getirmektedir.

Her eğitim programında olduğu gibi bilgisayar bilimi eğitim programlarında hedefler, içerik, etkinlikler, öğrenci katılımı, öğretmen uygulamaları, pedagoji ve sınıf normları kritik öneme sahiptir (Goode ve diğ., 2014). Bu nedenle bilgisayar bilimi eğitim programlarının geleceği için bilgisayar bilimi öğretmenlerinin bu bileşenlere yönelik mesleki gelişimleri desteklenmeli, bilgisayar bilimi alanında esnek ve sürdürülebilir eğitim programları geliştirilmeli (Ketelhut ve diğ., 2020) ve bu programları uygulayacak iyi eğitilmiş ve kendine güvenen öğretmenler yetiştirilmelidir (Goode ve diğ., 2021). Ülkeler, bilgisayar bilimi öğretmenlerinin öğretim gerçekleştirdikleri kurumlarda tek başlarına olması, mesleki gelişim çalışmalarına yeterince ulaşamaması ve eğitim sistemlerinde bilgisayar bilimi dersine yeterince önem verilmemesi gibi zorlukları aşmak ve bilgisayar bilimi eğitime erişimi genişletmek amacıyla öğretmenlere yönelik düzenlenen mesleki gelişim programlarının sayısını artırmaktadır (Ni ve diğ., 2023). Mesleki gelişim programlarındaki artış bilgisayar bilimi dersi öğretmenlerinin güncellenen öğretim programlarıyla uyumlarını kolaylaştırabilir.

Öğretmenlerin eğitim programlarındaki değişimlere hazırlıklı olmaları sadece yeni alan bilgileriyle donatılmalarıyla ilgili değil, bu bilgileri öğrencilerle nasıl paylaşacaklarıyla da yakın ilişkilidir (Sentance ve Csizmadia, 2017). Nitekim son yıllarda öğretmenlerin bilgisayar bilimi alan bilgilerini arttırmaya ve bu alanda yer alan kavramların öğrencilere nasıl kazandırılacağına ilişkin çalışmalara artan bir talep vardır (Falkner ve diğ., 2019) ABD’de, bilgisayar bilimi alanındaki iş taleplerini karşılamada yetersiz kalınması nedeniyle bilgisayar bilimi öğretmenlerinin sayılarının ve niteliklerinin artırılmasına yönelik çalışmaları (Hamlen ve diğ., 2018) ve Birleşik Krallığın bilgisayar bilimi öğretmenlerine yönelik hem hizmet öncesi öğretmen eğitime hem de öğretmelerin sürekli mesleki gelişimlerine odaklanması (Sentance ve diğ., 2012) örnek gösterilebilir.

Eğitim programları içeriğinin teknolojik gelişmelerden hızlı bir şekilde etkilenmesi, bilgisayar bilimi öğretmenlerinin alan bilgisine yönelik içeriği dikkatli bir şekilde takip etmesini gerekli kılmaktadır. Son yıllarda bilgisayar bilimi alanında yazılım endüstrisinin ülke ekonomilerine katkısı nedeniyle özellikle programlama konuları dikkat çekmekte ve bilgisayar biliminin önemli bir içeriği haline gelmektedir.

Abesadze ve Nozadze'ye (2020) göre programlamanın yıllar önce özel yetenekli insanlar tarafından öğrenilmesi gerekliliği fikri, yerini 21. yüzyıl öğrencileri için önemli bir yetenek olarak görülmesine ve birçok eğitim programının önemli bir bileşeni haline gelmesine bırakmıştır. Estonya, Güney Kore, İngiltere, Amerika Birleşik Devletleri, Fransa, Finlandiya, Polonya ve Norveç gibi gelişmiş ülkeler yeni nesillerin eğitimi için güncel programlama konularını da kapsayan eğitim programları geliştirmiştir (Balanskat ve Engelhardt, 2015; Chou ve diğ., 2019; Gülbahar ve Kalelioğlu, 2018). Gelişmiş ülkeler bilgisayar bilimi eğitim programlarıyla programlama öğretimini daha geniş kitlelere ulaştırmayı hedeflemiştir.

Türkiye'de bilgisayar bilimine ait içerikler dönemin ihtiyaçları dikkate alınarak bilgisayar okuryazarlığına yönelik başlamıştır. 1998 yılında yürürlüğe giren İlköğretim Seçmeli Bilgisayar (4-8. sınıflar) Dersi Öğretim Programını 2006 yılında İlköğretim Bilgisayar Dersi (1-8. sınıflar) Öğretim Programı, 2012 yılında Ortaokul ve İmam Hatip Ortaokulu Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. sınıflar) Öğretim Programı, 2018 yılında ortaokul 5. ve 6. sınıflar ile 7. ve 8. sınıflar için ayrı ayrı yayımlanan Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı izlemiştir. Gülbahar ve Kalelioğlu'na (2017) göre 1998-2011 yılları arasında programlarda bilgi ve iletişim konularına odaklanılırken, 2012 yılından sonra problem çözme ve programlama konularına ağırlık verilmiştir. Eğitim programlarında öğrencilerden beklenen yeterliklerin okuryazarlık düzeyinden programlama becerisine doğru evrildiği görülmüştür.

2018 yılında lise düzeyinde uygulanmaya başlanan Bilgisayar Bilimi (Kur 1-2) Öğretim Programında, Ortaokul Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programıyla dikey kaynaşık problem çözme ve programlama konuları yer almaktadır. Kur-1'deki programlama ünitesi, toplam kurun yüzde 49'unu oluşturmakta ve daha çok bilgi işlemsel düşünme ve programlama becerilerinin geliştirilmesine odaklanmaktadır. Kur-2 ise öğrencilerin robot programlama, web tabanlı programlama ve mobil programlama gibi alanlardan birini veya birkaçını seçerek ürün geliştirmelerine yöneliktir. Öğretim programında programlamayla birlikte bilgi işlemsel düşünme, eleştirel düşünme, algoritmik düşünme ve karar verme becerilerine ayrıntılı olarak yer verilmektedir. Gülbahar ve Kalelioğlu (2017), bu becerilerden bilgi

işlemsel düşünme becerisini 2020’li yılların önemli ve dikkat çekici becerileri arasında görmektedir.

Wing’e (2006) göre bilgi işlemsel düşünme; problem çözme, sistemler tasarlama ve insan davranışlarının bilgisayar biliminin temel kavramları kullanılarak anlaşılmasıdır. Korkmaz ve diğerlerine (2017) göre bilgi işlemsel düşünme, yazma, okuma ve matematik gibi temel becerilerdendir ve sadece bilgisayar bilimciler için değil, herkes için temel bir beceridir. Son yıllarda oldukça dikkat çeken bu beceri, programlama öğretiminde sıklıkla (ör. Atman Uslu, 2018; Fagerlund ve diğ., 2021; Oluk ve diğ., 2018; Tikva ve Tambouris, 2021) kullanılmaktadır. Ancak García-Peñalvo ve Cruz-Benito’ya (2016) göre birçok ülke bilgisayar bilimi derslerinde öğrencilerin bilgi işlemsel düşünme, mantıksal düşünme ve problem çözme becerilerini geliştirmek yerine programlama öğretimi üzerine odaklanmaktadır. Bu durum programlama öğretiminin ülke ekonomisine yazılım sektörü aracılığıyla katkı sağlaması ve programlamanın öğretim programlarında yer alan diğer becerilere oranla daha somut ve kolay ölçülebilir çıktılar üretmesi ile açıklanabilir.

Programlama, dijital dünyada yaratmanın, sorunları çözmenin ve fikirleri hayata geçirmenin bir yoludur (Nouri ve diğ., 2020). Bu nedenle, öğrencilerin programlamada başarılı olabilmeleri üst düzey düşünme becerilerine sahip olmalarına bağlıdır (Santos ve diğ., 2020). Alanyazındaki birçok araştırma (ör. Bennedsen ve Caspersen, 2019; Santos ve diğ., 2020; Yıldız Durak, 2018) programlama öğrenmenin zorluğuna işaret etmekte; yaşanan zorlukların nedenleri programlamayla ilk kez karşılaşan bireylerin soyut kavram ve süreçleri anlamakta zorlanmaları (Funcke, 2014; Olsson ve diğ., 2015), öğretmenlerin programlama öğretim yeterliliklerindeki sınırlılıklar (Pérez-Marín ve diğ., 2020; Qian ve Lehman, 2017) ve karmaşık yapıdaki programlama dillerinin geleneksel öğretim yöntemleriyle öğretilmeye çalışılması (Erümit ve diğ., 2019; Gaspar ve Langevin, 2007) olarak sıralanmaktadır.

Öğretmenler, programlamanın bilgisayar bilimi dersi öğretim programlarında yoğun biçimde yer almaya başlamasıyla birlikte programlamanın soyut kavram ve süreçlerini öğrencilerle paylaşmak ve bu kavramlar aracılığıyla uygulamalar geliştirmelerini desteklemek için çeşitli yollar aramaktadırlar. Programlama öğrenmeye yeni başlayan öğrenciler programlama kavramlarını zihinlerinde

modellemede sorun yaşadıklarından (Derus ve Ali, 2012; Funcke, 2014) ve kavram setlerini eşzamanlı olarak öğrenmeleri yoğun bilişsel süreç gerektirdiğinden (Shuhidan, 2012; Wang ve diğ., 2009) programlamaya yönelik olumsuz tutum geliştirebilmektedir. Programlama öğretiminde öğrencilerin programlamaya yönelik olumlu tutum geliştirmelerini sağlamak ve bilgisayar bilimlerine ilgilerini arttırmak amacıyla araç (ör. Kalelioğlu, 2015; Shin ve diğ., 2013) ve öğrenen odaklı (ör. Bers, 2019; López-Pernas ve diğ., 2019) birçok yöntem geliştirilmiştir. Türkiye’de programlama öğretiminde blok tabanlı, metin tabanlı ya da fiziksel programlama gibi araç odaklı yaklaşımlar ve proje tabanlı öğrenme gibi öğrenen odaklı yaklaşımlar kullanılmaktadır (Kert, 2018).

Alanyazında, bilgisayar biliminde programlamanın nasıl öğretilceğine (Jumankuziev, 2023; Konan, 2020; Pérez-Marín ve diğ., 2018) ve programlama araçlarına ve öğretimine ilişkin birçok araştırma (ör. Kukul ve Gökçearslan, 2014; Koulouri ve diğ., 2015) yer almaktadır. Programlama öğretiminin pek çok beceriye olumlu katkısına rağmen araştırmaların daha çok öğretimsel araçlar üzerinde yoğunlaştığı (ör. Aydoğdu, 2020; Numanoğlu ve Keser, 2017; Salleh ve diğ., 2013) görülmektedir. Erümit ve diğerlerine (2019) ve Popat ve Starkey’e (2019) göre programlama öğretiminin sistematik bir şekilde gerçekleşmesi için gerekli pedagojik araştırmalar ise oldukça sınırlıdır. Oysaki pedagojik araştırmalar, bilgisayar bilimi öğretmenlerinin anlaşılması güç programlama kavramlarını daha anlaşılır hale getirmelerine, öğrencilerin programlamaya yönelik olumlu tutum geliştirmelerine ve program yazma becerilerine katkıda bulunabilir.

Programlama kavramlarının öğretimine yönelik pedagojik araştırmalar (ör. Cao ve diğ., 2016; Forišek ve Steinová, 2012; Hui ve Umar, 2011; Moape ve diğ., 2018) incelendiğinde metaforların programlamanın soyut kavram ve süreçlerin anlaşılmasında yaşanan zorlukların üstesinden gelmek için sıklıkla kullanıldığı görülmüştür. Metaforların psikoloji ve dilbilim alanında önemli bir yeri vardır (Leary, 1990; Şeyihoğlu ve Genç, 2011; Tuleja, 2017). Metaforlar, etrafımızdaki dünyayı algıladığımız mercekleler gibi davranarak (Wegner ve Nückes, 2015) soyut bir konuyu daha somut, tanıdık terimlerle anlamamızı sağlar ve onlara biçim verirler (Gibson ve Zellmer-Bruhn, 2001; Morgan, 2006). Bir metafor genellikle birbiriyle ilişkili

olmayan iki fikri veya nesneyi aralarındaki benzerliđi vurgulamak iin birbirine bađlar (Wallace, 2001). Sadece retorik bir ara deđil, gerekliđe iliřkin algılarımızı ve anlayışımızı yapılandırabilen bir gttr (Kvecses, 2010; Lakoff ve Johnson, 1999). En basit haliyle metaforlar, aıklayıcı yapıları tanıdık bir deneyim alanından, anlaşılması veya yeniden yapılandırılması gereken bařka bir alana taşırlar. Krippendorff'a (1993) gre metaforlar iki alan arasında bazı yapısal benzerlikleri grmeyi gerektirmekle birlikte herhangi bir yapısal benzerliđin ok tesinde organize olurlar. Bu kapsayıcı yapısal zellikleri, metaforları edebiyat alanının dıřına taşıyarak daha geniř kapsamda kullanılmasına olanak sađlar.

Metaforların alanlar arasındaki taşıyıcı zelliđi birok bilim dalının dikkatini ekmiř (Zeren ve Yapıcı, 2014), zellikle sosyal bilimler alanında kuramların geliřtirilmesinde olduka yararlı bir ara olarak kullanılmıřtır (Koohang ve Harman, 2005). rneđin bađlam ve kltr, anlamı ve gerekliđi aıklamak iin ortaya ıkan yapılandırmacılık (Doolittle, 2014) temelde bir đrenme metaforudur ve bilgi edinmeyi bir inřa srecine benzetir (Ernest, 1994; Fox, 2001). Yapılandırmacılıđın temel metaforlarından bir diđeri de đrenmeyi destekleyen iliřkiler inřa etmektir (Lane, 2018; Thomas ve McRobbie, 2001). Yapılandırmacı đretmenler, sınıf etkinliklerini yapılandırırken đrenmenin gerekleřmesi iin gerekli zamanı ve materyalleri đrencilere sađlayarak đrencilerin kavramlar ve kuram arasında iliřkiler kurmalarına yardımcı olurlar (Brooks ve Brooks, 1999; Campbell-Larsen, 2012; Hung, 2002). Yapılandırma srecinde kavramlar ve kuramlar arasındaki iliřki, etkin anlam oluřturmanın nemli bir rneđidir (Moape ve diđ., 2018; Tay, 2016) ve kavramsal metafor kuramı bunu olduka iyi aıklar.

Kavramsal metafor kuramı, Lakoff ve Johnson'ın (1980) *Metaforlar: Hayat, Anlam ve Dil* kitabında ayrıntılı olarak aıklanarak ađdař yaklařım metaforlarının yaygınlařmasında olduka etkili olmuřtur. Klasik yaklařımlar metaforları dilde kullanılan bir ifade biimi olarak dřnce ve gnlk yařamdan ayrı tutarken, ađdař metafor kuramları metaforları soyut bir durumu aıklamak iin uygun bir zihinsel ara olarak grmřtr (zelmacı ve akır, 2017). ađdař metafor kuramlarından kavramsal metafor kuramına gre insanın dřnme sreci ođunlukla metaforiktir. Metaforlar sadece dil ile ilgili olarak dřnlse de aslında dřnce ve eylem ile

ilişkilidir. Metaforun özü aslında bir tür şeyi başka bir tür şeye göre anlamak ve tecrübe etmektir. Örneğin “Tartışma savaştır.” kavramsal metaforunda olduğu gibi tartışma sırasında yapılan birçok şeyi (tartışmayı kazanma/kaybetme, tartışılan kişiyi bir hasım gibi görme, onun mevzilerine saldırma, stratejiler kullanma vb.) savaş kavramı bir yapıya kavuşturur (Lakoff ve Johnson, 1980). Kavramsal metafor kuramı, bir olgu ya da kavramın sadece dil ile değil, günlük yaşamda edinilen deneyimler aracılığıyla da şekillenebileceğini açıklayarak yaşıntının kavram oluşturmadaki yerine vurgu yapmıştır.

Kavramsal metaforlar, bilimsel kavramların öğretiminde de büyük ilgi görmüş (Campbell-Larsen, 2012) biyoloji (Lancor, 2014; Niebert ve Gropengiesser, 2015), kimya (Beger ve Jakel, 2015; Brist, 2012) ve matematik (Font ve diğ., 2010; Uysal, 2016) gibi bilim alanlarına ait kavramların öğretiminde yaygın olarak kullanılmıştır. Son yıllarda bilgisayar biliminde algoritma (Forişek ve Steinová, 2012), programlama (Esper ve diğ., 2013; Pérez-Marín ve diğ., 2020; Pérez-Marín ve diğ., 2018), dinamik bellek (Jiménez-Peris ve diğ., 1997) ve matrisler (Milner, 2010) gibi bazı özel kavramların öğretiminde de metaforlardan yararlanılmıştır. Özellikle programlamayı öğrenmede soyut kavramların doğru anlaşılması oldukça önemli olduğundan (Lahtinen ve diğ., 2005) metaforlar bilgisayar bilimi derslerinde öğretmenlerin zor ve soyut kavramları daha anlaşılır hale getirmelerine yardım ederek (Pérez-Marín ve diğ., 2018) programlama öğretiminde kullanılabilmektedir.

Son yıllarda programlama öğretiminde yaşanan pedagojik tartışmalara birçok araştırma (ör. Cao ve diğ., 2016; Funcke, 2014; Hui ve Umar, 2011; Pérez-Marín ve diğ., 2018; Pérez-Marín ve diğ., 2020) metafora dayalı öğretim ile katkıda bulunmakta, bilgisayar bilimi alanında belli bir kavram için seçilen metaforun uygunluğunu kuramsal temelleri dikkate alarak incelemektedir (Moape ve diğ., 2018). Öğretimde uygunsuz kullanılan metaforlar, öğrencinin zihninde yapılanan modele zarar verebilir (Funcke, 2014). Bu nedenle bir kavrama ait uygun zihinsel şemalar oluşturmak için kavramın öğretiminde kuramsal temele dayanan metaforlar kullanmak önemlidir.

Türkiye’de metafora ilişkin araştırmalarda bir kavram ya da eyleme ilişkin algıların ortaya konulmaya çalışıldığı ya da metaforların veri toplama aracı olarak

kullanıldığı görülmektedir. Metaforların öğretimde kullanımına verilebilecek örnekler ise oldukça sınırlıdır. Programlama öğretiminde ise programlamaya ait kavramların metafor analizini konu edinen çok kısıtlı araştırma (Gökoğlu, 2017) bulunmakla birlikte programlama öğretiminde metafor kullanımına ilişkin araştırmaya rastlanmamıştır.

Bilgisayar bilimi eğitim programları çoğu zaman programlama gibi yapısal ve mantıksal zorluklar barındıran bir içeriğe (Çakır, 2019; Heintz ve diğ., 2016) sahip olduğundan lise eğitim programlarında yer alan birçok dersten farklıdır. Bu zorlayıcı durum birçok lise öğrencisinin derse olan tutumlarını olumsuz etkilemektedir (Major ve diğ., 2011; Papadakis ve Kalogiannakis, 2017). Özellikle metin tabanlı programlamaya yeni başlayan lise öğrencilerinin başarısızlık oranlarının yüksek olmasını ve öğrencilerin başaramayacaklarına olan inançlarını konu alan kapsamlı araştırmalara (ör. Cheah, 2020; Qian ve Lehman, 2017) rağmen sorun devam etmektedir. Öğrencilerin karşılaştıkları bu mantıksal zorlukların ortaya konması, nedenlerinin belirlenmesi ve çözüm yolları aranması bu nedenle oldukça önem kazanmaktadır.

Öğrencilerin başaramayacaklarına ilişkin inançlarını ortaya çıkaran en temel nedenlerden biri programlamanın yapısında birçok soyut kavramı barındırması (Cheah, 2020), diğeri ise öğrencilerden bu kavramları öğrenmenin yanında üst düzey birçok beceriyi bir arada kullanmalarının beklenmesidir (Noh ve Lee, 2020). Buna dayanarak bu araştırmada, ilk olarak araştırmacı tarafından metafora dayalı öğretimi temele alan ders planları hazırlanmış ve deney grubu bilgisayar bilimi öğretmenine hali hazırdaki yazılı programdaki programlamanın soyut kavramlarının metafora dayalı öğretimi için uygulama süreçlerinin planlanması, yürütülmesi ve değerlendirilmesi konusunda eğitim verilmiştir. Sonrasında metafora dayalı öğretimle yürütülen denencel bilgisayar bilimi dersinde bu kavramları öğrenen öğrencilerin, söz konusu kavramlara karşılık gelen kodlarla nasıl bir programlama süreci geçirdikleri, denencel sürecin onların programlama becerilerini nasıl etkilediği, geliştirdikleri ürünlerde ve ürün geliştirme süreçlerinde programlama becerilerinin nasıl değiştiği ve son olarak bilgi işlemsel düşünme öz yeterliğinin nasıl bir değişim gösterdiği incelenmiştir.

1.2. Araştırmanın Amacı ve Önemi

Bu araştırmanın temel amacı, kavramsal metafor kuramına dayalı metin tabanlı programlama öğretiminin öğrencilerin programlama kavramlarını kazanmaya, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliklerine etkilerini incelemektir.

Eğitim programı hedefleri, ülkelerin eğitim politikalarıyla yakın ilişkilidir ve hedeflerin bu politikaları desteklemesi beklenir. Hedefler, eğitim programlarının hazırlanması, uygulanması ve değerlendirilmesine kaynak sağlayan ekonomi politikalarından etkilenir ve gelişmiş ülkeler ekonomilerine katkı sağlayacak nicelikte ve nitelikte bireylerin yetiştirilmesini eğitim sistemlerinden talep eder. Son yıllarda yazılım sektörünün ekonomiye katkısı, bu alanda yetişmiş insan gücüne gereksinimi ortaya koymaktadır. Bu gereksinim, programlama becerisinin kazandırılmasında öğretim programının bir bileşeni olarak eklenmesi, ayrı bir disiplin olarak öğretilmesi, yatay kaynaşıklık ilkesi çerçevesinde diğer öğretim programlarıyla bütünleştirilmesi ya da ilgili öğretim programlarına sızdırılması gibi yaklaşımları ortaya çıkarmıştır. Bu araştırmada metafora dayalı öğretim, öğrencilerin programlama kavramlarını öğrenmelerini ve elektronik ürün geliştirmelerini destekleyen pedagojik bir araç olarak önerilmekte ve böylelikle hem açıklanan yaklaşımlarla programlama öğretimine hem de dolaylı olarak ülkelerin ihtiyaç duyduğu yetişmiş insan gücüne katkı sağlanabileceği düşünülmektedir.

Programlama becerisi genellikle bilgi işlemsel düşünme, problem çözme ve yaratıcı düşünme gibi üst düzey düşünme becerileriyle birlikte ele alınmaktadır. Birçok ülke eğitim programlarında programlamaya üst düzey düşünme becerileriyle birlikte yer vermekte ancak programlama alanının sürekli değişen ve gelişen dinamik yapısı nedeniyle program çalışmaları zorlaşmaktadır. Çözüm olarak yürütülen program güncelleme çalışmaları ise öğretmenlerin programlama öğretimine yönelik mesleki gelişimlerini sürekli desteklemeyi gerekli kılmaktadır. Bu araştırmada programlama becerilerine etkisi incelenen etkili bir zihinsel araç olarak metafor kullanımına ilişkin sonuçlar, programlama becerisi ve sıralanan üst düzey düşünme becerilerinin karşılıklı ve birlikte gelişimini desteklemede kullanılabilir. Bunun için metafora dayalı öğretim, mesleki gelişim programlarında yer alabilir ya da

öğretmenlere programlama öğretimi için bir seçenek olarak sunulabilir. Ayrıca, bilgi işlemsel düşünmenin öğrencilerin kavram öğrenme ve elektronik ürün geliştirme süreçleriyle yakın ilişkili olduğu kabul edildiğinde, bu araştırmadan elde edilen metafora dayalı öğretimin bilgi işlemsel düşünmeye etkisine yönelik sonuçlarla önceki araştırma sonuçlarının karşılaştırılması mümkün olabilir.

Bilgisayar bilimi öğretmenleri programlamaya yönelik hedefleri kazandırmada araç odaklı yaklaşımları temele alabilmektedir. Son yıllarda araç sayısındaki çokluk nedeniyle, bu yaklaşımın yerine programlama becerilerinin kazandırılması öne çıkmaktadır. Programlama araçlarından bağımsız olarak öğrencilerden programlama becerilerine sahip olmaları beklenmektedir. Öğretmenlerin mesleki gelişim çalışmaları bu değişimden etkilenerek araç odaklı mesleki eğitim yerini öğrenen odaklı pedagojik yaklaşımlara bırakmaktadır. Metafora dayalı öğretim, öğrenen odaklı pedagojik yaklaşımlardan biri olarak programlama öğretimine yönelik öğretmenlere düzenlenecek mesleki gelişim çalışmalarında yer alabilir.

Programlama, genel olarak kod yazma süreci olarak algılsa da aslında sürecin planlanması, kodlanması ve değerlendirilmesi aşamalarını da içerir. Yapısında çoğu zaman anlaşılması zor birçok soyut kavram yer aldığından öğrenme sürecinin başındaki öğrenciler programlamaya olumsuz tutum sergileyebilir. Öğretmenler, öğrenme sürecinde öğrencileri etkin kılmak ve bu soyut kavramları daha anlaşılır hale getirmek için birçok yöntem kullanabilir. Öğrencilerin programlamanın soyut kavramlarını anlamlı bir şekilde yapılandırmalarının yolu, kendi bilişsel ilişkisel ağlarını oluşturmalarıyla mümkün olabilir. Böylelikle öğrenciler zihinlerinde iyi yapılandırılmış kavramlar oluşturarak bu kavramlara karşılık gelen kodlarda uygulamalar geliştirebilir ve programlamaya karşı olumlu tutum sergileyebilir.

Araştırmada, programlama kavramlarını öğretim süreçleri bağımsız değişken olarak alınmıştır. Öğrencilerin programlama kavramlarını öğrenirken ilişkisel ağlarını oluşturmalarına yardımcı olmak için dersin öğretim programına bağlı kalınarak metafora dayalı öğretim süreçleri planlanmış, uygulanmış ve öğrencilerin kavram öğrenmelerine etkileri incelenerek değerlendirilmiştir. Metafora dayalı öğretimin programlama kavramlarını öğrenmeye etkilerinin nitel ve nicel kanıtlarla incelenmesi,

öğrencilerin programlamaya karşı tutumlarında ve dersten erken kopmaların önüne geçmede olası etmenleri irdelemeye yardımcı olabilir.

Programlama kavramlarını etkin ve kalıcı bir şekilde öğrenen öğrencilerin aynı zamanda programlama kavramlarına karşılık gelen kodları kullanarak daha nitelikli elektronik ürünler geliştirmesi beklenebilir. Kavram öğrenme ve programlama becerisi arasındaki bu ilişki, araştırmada metafora dayalı gerçekleştirilen öğrenme-öğretme süreçlerinin programlama becerisi üzerinde nasıl bir etki yaratacağı konusunda merak uyandırmış, öğrencilerin elektronik ürün geliştirme süreci üzerinden bu verilere ulaşmak hedeflenmiştir. Bu nedenle araştırma, metafora dayalı öğretimin öğrencilerin programlama becerilerini geliştirmeye katkısını ortaya koyması açısından önemlidir.

Alanyazın taraması, Türkiye’de programlama öğretiminde metafor kullanımının yeterince incelenmediğini ve programlama öğretimi dışındaki alanlarda yürütülen araştırmaların ise sınırlı sayıda olduğunu göstermiştir. Metafora dayalı öğretimin planlanması, uygulanması ve değerlendirilmesine yönelik geniş ve derin bakış açısı oluşturabilecek araştırma eksikliği, lise düzeyinde gerçekleştirilen bu araştırmayı önemli kılmaktadır. Ayrıca, kuramsal temelinde kavramsal metafor kuramı yer alan bu araştırma, programlama öğretiminde kullanılan pedagojik yaklaşım çalışmaları arasında yer bulabilir. Bu araştırma, programlamanın soyut ve anlaşılması zor kavramlarını daha anlaşılır hale getirmede alternatif bir yöntem önererek öğretmenlere, metafora dayalı öğretimle ilgilenen araştırmacılara ve öğretim programı tasarılarının hazırlanması, uygulanması ve değerlendirilmesinde rol alan program geliştirme ekibine, program değerlendiricilere ve karar alıcılara katkılarının olacağı düşünülmektedir.

1.3. Problem Cümlesi ve Alt Problemler

Araştırmanın temel problem cümlesi “Metin tabanlı programlama öğretiminde metafor kullanımının lise 9. sınıf öğrencilerinin programlama kavramlarını öğrenmelerine, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliklerine etkisine ilişkin nicel ve nitel kanıtlar nelerdir?” biçiminde ifade edilmiştir. Araştırmanın bu temel problemine yanıt aramak için aşağıdaki alt problemler oluşturulmuştur:

1. Metin tabanlı programlama öğretiminde metafor kullanımı, programlama kavramlarını öğrenmeyi nasıl etkilemektedir?
 - a. Deney ve kontrol gruplarındaki öğrencilerin programlama kavramlarına ilişkin ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır?
 - b. Programlama kavramlarını öğrenme sürecini ve kavramsal değişimi ortaya koyan nitel kanıtlar nelerdir?
 - c. Programlama kavramlarını öğrenme sürecine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır?
2. Metin tabanlı programlama öğretiminde metafor kullanımının programlama becerilerine etkisi nedir?
 - a. Deney ve kontrol gruplarındaki öğrencilerin programlama becerileri ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır?
 - b. Programlama becerilerini edinme süreci ve bu becerilerin değişimini ortaya koyan nitel kanıtlar nelerdir?
 - c. Programlama becerilerine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır?
3. Metin tabanlı programlama öğretiminde metafor kullanımının bilgi işlemsel öz yeterliklerine etkisi nedir?
 - a. Deney ve kontrol gruplarındaki öğrencilerin bilgi işlemsel öz yeterliği ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır?
 - b. Bilgi işlemsel düşünme öz yeterliğindeki değişimini ortaya koyan nitel kanıtlar nelerdir?
 - c. Bilgi işlemsel düşünme öz yeterliğine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır?

1.4. Sayıtlar

- Kontrol edilemeyen değişkenler, deney ve kontrol grubunu benzer şekilde etkilemiştir.
- Deney ve kontrol grubunda yer alan öğrenciler aynı sınıf düzeyi (9. sınıf) ve yaş gruplarında (14-15 yaş) yer aldıklarından denel işlem süreci (18 hafta) boyunca benzer olgunlaşma göstermiştir.

- Deney ve kontrol grubundaki öğrenciler veri toplama araçlarına içtenlikle yanıt vermişlerdir.
- Araştırmanın deney ve kontrol grubu öğretmenleri denel işlemin uygulama ilkelerine uygun davranmıştır.

1.5. Sınırlılıklar

Bu araştırma,

- Deney ve kontrol grubunda denel işlem süresi olarak 36 ders saati (18 hafta),
- Manisa ili Yunusemre ilçesinde yer alan iki resmi anadolu lisesi,
- Dokuzuncu sınıfta öğrenim gören deney grubunda üç, kontrol grubunda üç olmak üzere toplam altı şube,
- Bilgisayar bilimi (Kur-1) dersinin Programlama Ünitesi,
- Deney ve kontrol grubunda yer alan öğrencilerin kavram öğrenme, programlama ve bilgi işlemsel düşünme öz yeterliği puanları ile öğrencilerin ve dersi veren öğretmenlerin görüş ve değerlendirmeleri ile sınırlıdır.

1.6. Tanımlar

Metafor: “Bir tür şeyi başka bir tür şeye göre anlama ve tecrübe etme” (Lakoff ve Johnson, 2015, s. 30).

Kavram Öğrenme: “Uyaranları belli kategorilere ayırarak, zihinde bilgiler oluşturma” (Ülgen, 2004, s. 117).

Bilgi İşlemsel Düşünme: Problem çözme, sistemler tasarlama ve insan davranışlarının anlaşılmasının bilgisayar bilimi temel kavramları kullanılarak ortaya konması (Wing, 2006).

Öz Yeterlik: “Bireyin belirli performansı elde etmek için gerekli davranışları gerçekleştirme kapasitesine olan inancı” (Bandura, 1977, s. 194).

Programlama: “Problemin tanımlanmasından kod grubu olarak bilgisayar ortamında çözümlenmesine dek gerçekleşen bir düşünme ve problem çözme süreci” (Korkmaz, 2019, s. 235).

II. BÖLÜM

KURAMSAL ALANYAZIN VE İLGİLİ ARAŞTIRMALAR

Bu bölümde ilk olarak programlama öğretiminde kullanılan yaklaşımlara, programlama becerisinin nasıl ölçülebileceğine ve Türkiye’de programlama öğretiminin mevcut durumuna ilişkin açıklamalara yer verilmiş, programlama öğretimine genel bir bakış sunulmuştur. Daha sonra metafora, kavramsal metafor kuramına ve eğitimde yansımalarına değinilmiş, son olarak da metafora dayalı öğretimin araştırmanın bağımlı değişkenleriyle doğrudan ilişkili yurt içi ve yurt dışı araştırmalar ele alınmıştır.

2.1. Programlama Öğretimi

Ülkeler, gelişmişlik düzeylerini koruyabilmek ya da gelişmiş ülkeler sınıflamasında yer bulabilmek adına teknolojiyi takip etmek, üretmek ve mevcut teknolojilerinin sürdürülebilirliğini sağlamak durumundadır. Sürdürülebilir teknolojik gelişmeler yetişmiş insan gücüyle gerçekleşebilir ve gelişmiş ülkeler yetişmiş insan gücünü daha nitelikli hale getirmek için çeşitli arayışlara girişmiştir. Barut ve Kuzu’ya (2017) göre gelişmiş ülkeler sosyal, kültürel ve ekonomik alanları etkileyen teknolojik gelişmelerin daha iyi takip edilebilmesi için eğitim sistemlerine bilgisayar bilimi derslerini dâhil etmiştir. Günümüzde bilgisayar biliminin eğitim sistemine ayrı bir disiplin olarak mı, yoksa eğitim programlarının disiplinlerarası bir unsuru olarak mı yer alacağı alanyazında (ör. Fisler ve diğ., 2021; Mabie ve diğ., 2023) tartışma konusudur.

Bilgisayar bilimine eğitim programlarında nasıl yer verileceğine ilişkin farklı ülkelerde farklı uygulamalar söz konusudur. Avrupa Komisyonu (European Commission [EC], 2016) raporuna göre bilgisayar bilimine ortaöğretim düzeyindeki programlarda ayrı bir disiplin, ilköğretim düzeyindeki öğretim programlarında ise başka bir alanla bütünleşmiş olarak yer verilmektedir. Heintz ve diğerlerine (2016) göre gelişmiş ülkeler bilgisayar bilimini başka bir alanla bütünleştirmeyi programa yeni bir disiplinin eklenmesi için yeterli alanın olmaması, farklı alanlarla bilişim teknolojilerinin birlikte kullanımının getireceği faydalar ve bilgi işlemsel düşünme becerisinin bilişim teknolojileriyle birlikte çalışmaya dair daha genel bir çerçeve sunması nedenleriyle tercih etmiştir. Bilgisayar biliminin disiplinlerarası bir unsur olarak ele alınmasının gerekçelerinden biri olan bilgi işlemsel

düşünme son yıllarda başka alanlarla birlikte çalışmaya oldukça elverişli olması ve alanlar arası bağlantılar kurmada önemli bir görev alması nedeniyle bilgisayar bilimlerinin önemli bir parçası haline gelmiştir.

Bilgisayar bilimi eğitim programlarında yer verilen bir diğer beceri programlama becerisidir. Avrupa Okul Ağının (European Schoolnet) Geleceğimizi Programlıyoruz (Computing Our Future) isimli raporuna göre programlama temel bir önceliktir ve programlama öğretiminin dijital okuryazarlığı ve bilişim teknolojilerinin gelişimini desteklediği belirtilmektedir (Balanskat ve Engelhardt, 2015). Şimşek'e (2018) göre de gün geçtikçe erken yaşta programlama eğitimine başlayan ülkelerin sayısı artmakta ve öğrencilerin programlamayı kolay öğrenebilmeleri için birtakım araçlar geliştirilmektedir. Erken yaşta programlama eğitimiyle ileriki yaşlarda daha nitelikli dijital vatandaşların yetiştirilmesi hedeflenmektedir.

Türkiye'de programlama öğretiminin gelişim süreci bilgisayara yönelik öğretim programları tarihinin incelenmesiyle daha iyi anlaşılabilir. İlköğretim okullarında bilgisayar okuryazarlığının yaygınlaştırılması amacıyla ilk kez 1998 yılında İlköğretim Okulları Seçmeli Bilgisayar Dersi (1-5) Öğretim Programı yayımlanmış, dördüncü sınıftan itibaren bir ya da iki saat olarak seçmeli ders olarak okutulmasına karar verilmiştir (Millî Eğitim Bakanlığı [MEB], 1998). 2006 yılında İlköğretim Bilgisayar Dersi (1-8. Sınıflar) Öğretim Programı 1998 yılında yayımlanan programın yerini almış, Bilişim Teknolojilerinde İleri Uygulamalar öğrenme alanıyla ilköğretim düzeyinde ilk kez programlama öğretimine yer verilmiştir. Bu öğrenme alanı programlamaya giriş, nesne tabanlı programlama ve web tasarımı alt öğrenme alanlarından oluşmuştur (MEB, 2006a). 2012 yılında Ortaokul ve İmam Hatip Ortaokulu Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı ile programın güncelleme çalışmaları gerçekleşmiş, öğretim programında yer alan dört yeterlilik alanından biri olan Programlama ve Özgün Ürün Geliştirme ile öğretim programı diğer programlardan farklılık göstermiştir (MEB, 2012). 2018 yılına Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı 1-4. sınıflar, 5 ve 6. sınıflar ve 7 ve 8. sınıflar olmak üzere ayrı ayrı yayımlanmış, her üç öğretim programında da problem çözme ve programlama ünitelerine yer verilmiştir (MEB 2018a, 2018b).

Lise düzeyinde 2005 yılında yayımlanan Ortaöğretim Bilgi ve İletişim Teknolojisi Dersi Öğretim Programının meslek liselerinde zorunlu, diğer liselerde ise seçmeli olarak okutulmasına karar verilmiştir (MEB, 2006b). Ancak programlamaya ilişkin uygulamalar, veri tabanı programı ünitesinde yer alan uygulamalarla sınırlı kalmıştır (Kalelioğlu, 2018). 2016 yılında Bilgisayar bilimi dersi öğretim programının güncellenmesine ve liselerde kademeli olarak uygulanmasına karar verilmiş, daha sonra bu karar 2018 yılında yayımlanan Bilgisayar Bilimi (Kur-1, Kur-2) Öğretim Programı ile tüm sınıf düzeylerinde uygulanabilecek şekilde yeniden düzenlenmiştir (Talim Terbiye Kurulu Başkanlığı [TTKB], 2018). Bilgisayar bilimi dersi (Kur-1) Etik, Güvenlik, Toplum, Problem Çözme ve Algoritmalar ve Programlama ünitelerini içermiş, Kur-2 ise Robot Programlama, Web Tabanlı Programlama ve Mobil Programlama ünitelerinden seçim yapmaya olanak tanımıştır (MEB, 2018c). Son olarak 2023 yılında yayımlanan Ortaöğretim Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programında alan becerileri, kazanımlar, içerik ve eğitim durumlarında programlamayla doğrudan ilişki kurulmuştur (MEB, 2023).

Türkiye’de yıllar geçtikçe öğretim programlarında programlamaya yönelik hedefler, içerik, eğitim ve sınav durumları yoğun olarak yer almaya başlamış, programlarda yer alan kazanımlar veri tabanının programlanmasından yapay zekânın programlanmasına doğru evrilmiştir. Eğitim programlarında yaşanan bu değişimle öğretmenler programlama öğretiminde yeni yaklaşımlara ihtiyaç duymuştur. Bu nedenle araştırmanın bu bölümünde programlama öğretimine yönelik yaklaşımlara, programlama becerisinin nasıl ölçülebileceğine ve programlama öğretiminin Türkiye’deki durumuna yer verilmesi uygun görülmüştür.

2.1.1. Programlama Öğretimi Yaklaşımları

Programlama öğretiminde kullanılan öğretim yaklaşımları diğer disiplinlerde kullanılan yaklaşımlarla oldukça benzerlik göstermektedir. Öğretmen merkezli, içerik temelli, uygulamaya dayalı ve öğrenen odaklı yaklaşımlar programlama öğretiminde sıklıkla kullanılmaktadır. Öğretmenler, programlamanın ilk yıllarında her ne kadar teknoloji ile pedagoji arasında kalarak teknolojik araçların öğretimine odaklandıkları görülse de programlama öğretiminin disiplinlerarası bir boyutta ele alınması ve düşünme becerilerinin gelişiminin hedeflenmesi ile programlama öğretiminde

pedagojik eğilimler ön plana çıkmaya başlamıştır (Kert, 2018). Özellikle Seymour Papert'ın öğretim yaklaşımlarının somutlaştırılmasına yönelik yapılandırmacılık önerileri programlama öğretiminde programlama dili yönelimli yaklaşımlara alternatif eğlenceli bir keşif yolu sunmuştur (Michaelson, 2018).

Programlama öğretiminde yaygın olarak kullanılan öğretim yaklaşımları öğrenen odaklı ve araç odaklı olmak üzere iki grupta sınıflanabilir. Öğrenen odaklı yaklaşımlar öğrencinin programlama becerilerini geliştirmeye yoğunlaşırken, araç odaklı yaklaşımlar öğretimde kullanılan aracın ya da dilin öğretilmesini ön planda tutmaktadır.

2.1.1.1. Öğrenen Odaklı Yaklaşımlar

Öğrenen odaklı yaklaşımlara oyun ve oyunlaştırma, dijital hikâye anlatımı, proje tabanlı öğrenme, sorgulama temelli öğrenme, disiplinlerarası yaklaşım, eşli programlama, ters yüz öğrenme modeli, harmanlanmış öğrenme ve farklılaştırılmış öğretim örnek verilebilir.

Oyun ve oyunlaştırma yaklaşımı, öğrencilerin anlamakta güçlük çektikleri durumlarla karşılaştıklarında bilgi ve becerilerinin farkına varmalarını sağlayarak öğrenme sürecine katılımlarını arttırmakta (Fidan, 2016), programlamaya yönelik içsel motivasyonlarını geliştirmelerine katkı sağlamaktadır (Rodrigues ve diğ., 2021). Öğrenmeyi olumlu yönde etkileyen bir yaklaşım olmasına (Venter, 2020) ve programlama öğretimi için pek çok oyun geliştirilmesine rağmen eğitim programlarında nasıl yer alacağı konusu henüz tam olarak netlik kazanmamıştır (Lindberg ve diğ., 2019). Bu nedenle öğretim programlarında bu yaklaşıma ilişkin açıklamalara yer verilmeli, sınıflarında oyun ve oyunlaştırma yaklaşımıyla programlama öğretimine yer vermek isteyen öğretmenler bu yaklaşımı dikkatle incelemelidir.

Dijital hikâye anlatımı yaklaşımı, öğrencileri 21. yüzyıl öğrenme becerilerini kazanmaları için motive eden ve özellikle bilgi işlemsel düşünme becerisine katkı sağlayan güncel pedagojik yaklaşımlardan biridir (Vinayakumar ve diğ., 2018). Dijital hikâye anlatımı yaklaşımı hikâyenin duygusunu içererek (Burke ve Kafai, 2012), programlamanın temel kavramlarının öğretilmesine katkıda bulunmaktadır (Yıldız Durak, 2018). Ayrıca öğrencilerin bir konu seçme, araştırma yapma, senaryo yazma

ve bir hikâye geliştirme sürecinde görev alarak yaratıcı hikâye anlatıcıları olmalarına destek olmaktadır (Voinohovska, 2020). Öğrencilerin programlama kavramlarını öğrenirken hem hikâyeler geliştirmeleri hem de hikâyenin anlatıcıları olarak süreçte aktif rol almaları kavramları daha kalıcı öğrenmelerini desteklemektedir.

Proje tabanlı öğrenme (PTÖ) yaklaşımı, öğrencilerin bağlama dayalı daha kalıcı sonuçlar elde etmeleri için uygun bir yaklaşımdır (Gary, 2015). Bu yaklaşımda öğrencilerin geliştirmesi beklenen özgün projeler, gerçek yaşam durumları ile ilgilidir ve doğası gereği bütünleştiricidir (Doppelt, 2005). PTÖ, öğrencileri belirli bir yönlendirici soru etrafında işbirlikli çalışmaya teşvik ederek derin bir etkileşim sağlamakta, öğrencilerin karmaşık bir sorunu çözmek için belirli bir çalışma süreci planlamalarını gerektirmektedir (Fincher ve Knox, 2013). Bu sayede öğrencilerin sistem düzeyinde düşünmelerine ve sistem tasarım becerilerinin geliştirilmesine yardımcı olmaktadır (Karaman ve diğ., 2017). PTÖ aracılığıyla öğrenciler programlama görevlerine dahil edilerek işbirlikli öğrenme, sistem tasarlama ve gerçek yaşam durumlarındaki problemlerle karşılaşır (Wang ve diğ., 2016), programlama sürecinde elde ettikleri soyut bilgilerle gerçek programlama problemleri arasındaki ilişkileri kurmaları desteklenir (Peng ve diğ., 2017). Bu durum programlama öğretiminde öğrencilerin kuram-uygulama bağlantısı kurmalarında PTÖ'yü önemli ve destekleyici bir süreç haline getirir.

Sorgulama temelli öğrenme yaklaşımında öğrenciler eleştirel düşünmeye, soru sormaya ve sorgulamaya teşvik edilir (Kert, 2018). Öğrenilen bilgiyi kalıcı kılmak için bilginin öğretmen tarafından aktarılmasından ziyade öğrenciler tarafından oluşturulması beklenir (Coleman ve Nichols, 2011). Sorgulama temelli yaklaşım sorunları tanımlama, inceleme, olası çözümler üretme ve uygun gerekçeyle en iyi çözümü bulmayı içerir ve programcıların bu kritik becerilerde uzmanlaşması beklenir (Lukose ve Mammen, 2018). Vaníček'e (2015) göre, özellikle doğa bilimleri öğretiminde sıklıkla kullanılan bu yaklaşımın programlama öğretiminde uygun olup olmadığı dikkatle incelenmelidir. Dolayısıyla öğretmenler, öğrencilerden yazılımla çözülmesi gereken problemi net bir şekilde ifade etmelerini, çözmeleri için gerekli becerilerin farkına varmalarını, nasıl bir yol izleyeceklerine karar vermelerini ve

geliştirilen yazılımı değerlendirerek çözümün ne kadar etkili olduğunu ortaya koymalarını beklemelidir.

Bilimsel ve teknolojik gelişmelerle ortaya çıkan bilginin hızlı paylaşımı yeni araştırma alanlarını doğurmuş, gerçek yaşam problemlerinin tek bir disipline özgü becerilerle sınırlandırılmayacağı görülmüştür. Disiplinlerarası yaklaşım, modern dünyada matematik, fen bilimleri, mühendislik ve bilgisayar bilimlerinin iç içe geçmesinin bir sonucudur (Sedgewick ve diğ., 2015). Farklı hedef kitleler için farklı teknikler kullanarak hedefe ulaşmada bütünleştirici olan bu yaklaşım (Coblenz ve diğ., 2018), özellikle programlama öğretiminde yaygın olarak kullanılmaktadır (Numanoğlu ve Keser, 2017). Programlama öğretiminde bir kavramı öğretirken disiplinlerarası yaklaşımı kullanmak, kavramı bir disipline bağlı olarak öğretmekten çok daha fazlasını sunmaktadır (Hubálovský, 2012). Robot programlama buna iyi bir örnektir ve disiplinlerarası yaklaşımı benimseyen öğretim modellerinde önemli bir yere sahiptir. Bunun nedeni, robotların bir görevi gerçekleştirmek için fen bilimleri, matematik, mühendislik ve bilgisayar bilimi ilkelerini birada kullanmaya oldukça elverişli olmalarıdır.

Eşli programlama yaklaşımı, iki öğrencinin bir bilgisayarda farklı rollerde yan yana çalışması olarak tanımlanabilir (Campe ve diğ., 2020). İşbirlikli öğrenme kuramından türetilen bu yaklaşımda öğrencilerden biri bilgisayarı yönetirken diğeri rehber görevini üstlenir ve her bir rolün avantajlarından yararlanmak için roller rutin olarak değiştirilir (Dongo ve diğ., 2016). Programlamayı öğrenmede iyi bir strateji haline gelen bu yaklaşım, bireysel programlama yaklaşımlarına göre daha çok tercih edilmektedir (Celepkolu ve Boyer, 2018). Bunun nedeni eşli programlama yaklaşımının hem akrandan öğrenme hem de işbirlikli öğrenmenin avantajlarını bir arada sunmasıdır.

Ters Yüz öğrenme modelinde, öğrencilerin sınıf ortamında öğrenecekleri içeriğe, eş zamansız (asenكرون) planlar (ders videosu ve makaleler, çeşitli elektronik veri kaynakları, görseller, resimler ve sunumlar) yardımıyla okul dışında ulaşmaları sağlanmakta (Polat ve Karatabak, 2022) öğrenciler ders öncesinde, sırasında ve sonrasında öğrenmelerinin ve kararlarının sorumluluğunu almaktadırlar (Prust ve diğ., 2015). Ancak Hendrik ve Hamzah'a (2021) göre K-12 düzeyinde programlama için

ters yüz öğrenme modelini uygulamaya yönelik çabalar yetersizdir. Bu nedenle, bu yaklaşımın programlama öğretiminde nasıl bir etki yarattığına ilişkin daha çok kanıt ihtiyacı vardır.

Harmanlanmış öğrenme modeli, yüz yüze ve çevrimiçi öğrenme deneyimlerinin birleşimi olarak ifade edilebilir (Smith ve Hill, 2019; Sontay ve Karamustafaoğlu, 2022). Alanyazında model için hibrit, karma veya esnek öğrenme gibi terimler de kullanılmaktadır (Müller ve Mildemberger, 2021). Geleneksel öğretim yaklaşımlarının sınırlılıklarını aşmada etkili olan model (Cronje, 2020; Dangwal, 2017) programlamaya giriş derslerinde öğrencilerin performansını arttırmak için de kullanılabilmektedir (Alammery, 2019). Model, programlama öğretiminde kabul edilebilir sayıda çok öğrenciye ulaşılması ve sürecin kontrolünü sağlamada kolaylıklar sağlaması nedenleriyle tercih edilmektedir (Lazarinis ve diğ., 2019). Ters Yüz öğrenme modeli ile kullanımı, programlama öğretiminde karşılaşılan zorlukların üstesinden gelmede yardımcı olabilir.

Farklılaştırılmış öğretim, öğretim yaklaşımlarının öğrenci özelliklerine göre değişmesi ve uyarlanması gerektiğine dayanan kuramdır (Hall, 2002; Tomlinson, 2003). Programlama öğretiminde yaşanan zorluklardan biri sınıflardaki öğrenci çeşitliliğidir (Gil Fonseca ve diğ., 2018) ve farklılaştırılmış öğretim, yapısında, bu sorunun çözümüne yönelik birçok öğretim tekniğini (merkezler, istasyon, kademelendirilmiş etkinlik, öğrenme sözleşmesi, ajanda, öykü temelli öğrenme, grup araştırması, karmaşık öğretim, program sıkıştırma, okuma çemberi) barındırmaktadır. Bu teknikler programlama öğretiminde öğrenci çeşitliliğinden kaynaklanan problemlerin çözümünde kullanılabilir fakat alanyazında programlama öğretiminde kullanımıyla ilgili oldukça sınırlı kaynak (Gibbs, 2020; Gil Fonseca ve diğ., 2018) yer almaktadır.

2.1.1.2. Araç Odaklı Yaklaşımlar

Programlama öğretiminde kullanılan yaklaşımlardan bir diğeri de araç odaklı yaklaşımlardır. Araç odaklı yaklaşımlar blok tabanlı programlama, metin tabanlı programlama, fiziksel programlama ve bilgisayarsız programlama olarak sıralanabilir (Kert, 2018). Bu yaklaşımlara ek olarak son yıllarda programlama öğretiminde yapay zekâ araçlarının kullanımının arttığı görülmektedir.

Blok tabanlı programlama yaklaşımında kullanıcıya geliştirilecek yazılıma ait komutların nasıl ve nerede kullanılacağına dair ipuçları sağlayan yapboz metaforundan yararlanılmaktadır. Bloklar, programlama alanına sürüklenerek birbirlerine yapışmaları sağlanmakta ve geçerli bir sözdizimsel yapı oluşturulamadığında blok yapışmamaktadır (Weintrop ve Wilensky, 2017). Blok tabanlı diller, sözdizimi ile mücadele etmek yerine öğrencilerin ve eğitimcilerin algoritmalara odaklanmalarını sağlamayı, sözdizimi problemlerini ortadan kaldırmayı ve öğrencilerin programlamayı daha hızlı öğrenmelerini desteklemeyi amaçlar (Mladenovic ve diğ., 2018). Böylece programlamaya yeni başlayan öğrencilerin programlama öğrenmenin zor olduğuna yönelik algılarının ve programlama sürecinden erken kopmalarının önüne geçilmeye çalışılır.

Metin tabanlı programlama yaklaşımı, blok tabanlı programlama yaklaşımından farklı olarak bir programlama dilinin sözdizimini öğrenmeyi de destekler. Blok tabanlı programlama, programlama dilinin uygulanmasındaki belirli ara katman yazılım ihtiyacını karşılayamadığından metin tabanlı programlamaya başvurulur (Kim ve diğ., 2018). Öğrenciler blok tabanlı programlama dilinde geleneksel uygulamalar geliştirebilir, fakat üst düzey programlar geliştirmeleri gerektiğinde metin tabanlı programlamaya ihtiyaç duyarlar (Sentance ve Waite, 2017). Gelişmiş ülkeler, öğrencilerin profesyonel iş yaşamlarında çözümü üst düzey yazılım gerektiren problemlerle karşılaşabileceklerini düşündüklerinden metin tabanlı programlama öğrenmelerini desteklemektedir.

Fiziksel programlama yaklaşımı, programlamaya yeni başlayanların ilgisi, fiziksel teknolojilerin gelişimiyle blok tabanlıdan fiziksel programlamaya doğru yönelmiştir (Morrison ve diğ., 2019). Fiziksel programlama öğretiminde robotlar, akıllı nesneler, kendin yap kitleri, sanal robot programlama ortamları ve robot programlama dilleri yaygın olarak kullanılmakta (Numanoğlu ve Keser, 2017), bu araçların programlanmasıyla öğrencilere oyun formatlı eğitim içeriği sağlanarak programlama faaliyetlerinde gerekli olan kavramlar öğretilenmektedir (Shim ve diğ., 2016). Oyun formatlı fiziksel programlama içerikleri, öğrencilerin öğrenmekte güçlük çektikleri soyut programlama kavramlarını somutlaştırmalarında kolaylaştırıcı rol oynayabilir.

Bilgisayarsız programlama yaklaşımı, farklı yaş grupları için birçok yenilikçi uygulamayı kullanmaya olanak sağlayan yaklaşımlardan biridir ve doğasında oyun tabanlı öğrenme ve keşfederek öğrenmeyi barındırır (Kalelioğlu, 2020). Bu yaklaşımın, özellikle ilkokul ve ortaokul düzeyinde bilgisayar bilimi eğitimi almaya yeni başlayan öğrenciler üzerinde olumlu bir etkisi vardır (Olmo-Munoz ve diğ., 2020). Bilgisayarsız programlama yaklaşımı grup çalışması, problem çözme ve yaratıcılık becerilerinin gelişimini destekleyebilir (Cortina, 2015; Lamagna, 2015) ve bilgisayara ulaşımın olmadığı yerlerde de rahatlıkla kullanılabilmesiyle eğitimde fırsat eşitliğine katkı sunabilir.

Denny ve diğerlerine (2024) göre, programlama öğretiminde kapsamlı ve kaliteli geri bildirimler vermede yapay zekâ öğretim asistanları yardımcı olabilir. Zhu ve Van Brummelen (2021) ise konuşmaya dayalı yapay zekâ araçlarının öğrencilerin programlama öğrenmelerine katkı sağladığını ve yapay zekânın nasıl çalıştığı konusunda bilgi sahibi olduklarını belirtmiştir. Bu doğrultuda, programlama öğretiminde kullanılabilecek yapay zekâ araçlarının çokluğu ve bu araçlar arasından eğitsel açıdan etkili ve verimli olanlarını belirleme gereksinimi düşünüldüğünde, öğretmenlerin titiz ve dikkatli davranmaları gerektiği söylenebilir.

Programlama öğretimi sırasında bilgisayar bilimi öğretmenleri hangi yaklaşım/yaklaşımları kullanılırsa kullansınlar, öğrencilerden programlama performansı bekler. Öğretmenler, öğrencilerin programlama performanslarını inceleyerek öğrencileri ve kendi öğretim sürecini değerlendirebilir, öğretim tasarımlarını bu verilere göre yeniden düzenleyebilir. Öğrencilerin programlama becerileri, programlama performanslarını belirlenmede temel göstergelerden biridir.

2.1.2. Programlama Becerilerinin Ölçülmesi

Programlama becerilerinin ölçülmesi, programlamaya yönelik kavram öğrenme, programlama performansı ve yordayıcı beceriler üzerinden gerçekleştirilmektedir. Programlama performansına bağlı testler aracılığıyla yapılan ölçme, genellikle bir programlama dili ile sınırlıdır. Korkmaz'a (2019) göre yordayıcı becerilerin ölçümünde genellikle bağımsız ölçümlere gidilmektedir. Kavram öğrenme düzeylerinin ölçülmesinde ise başarı testleri ve kavram haritaları (ör. Kuo ve Chang, 2023; Menon ve Kovalchick, 2020) yaygın olarak kullanılmaktadır.

2.1.2.1. Programlama Kavramlarını Öğrenme Düzeylerinin Belirlenmesi

Kavram, felsefede insan bilişinin birimidir ve somut bir varlığı tanımlama ya da soyut nesneyi modellemede kullanılır (Wang, 2010). Farklı nesne, olay ve olguların sahip oldukları ortak özelliklere göre gruplandırarak bu gruplara ortak isimler verilmesidir (Underwood, 1966). Kavram öğrenme ise, kategorileri öğrenme ve bu kategorilerin yeni örneklerini tanıma yeteneğini ifade eden psikolojik bir kavramdır (Sammur, 2010) ve Ülgen'e (2004) göre uyarınları belli kategorilere ayırarak, zihinde bilgiler oluşturma olarak tanımlanmaktadır.

Programlama kavramlarını öğrenmede öğrenciler zorlanmakta (Eryılmaz, 2010; Santos ve diğ., 2020), bu durum öğrencilerin gerçek yaşam problemlerini etkili bir şekilde çözebilen programlar oluşturmalarını engellemektedir (Derus ve Ali, 2012). Öğrencilerin kavram öğrenmede yaşadıkları zorlukları belirlemek ve bu zorlukların önüne geçebilmek için kavram öğrenme düzeylerinin belirlenmesi önemlidir. Bunun için başarı testleri, gözlem formları, kontrol listeleri gibi çeşitli ölçme araçları kullanılabilir. Kavram haritaları kavramlara, kavramlar arası ilişkilere bağlantı sözcüklerine ve kavrama ait örneklere yer vererek daha kapsamlı ve bütüncül bir veri seti sunmasıyla (Ülgen, 2004) kavram öğretiminde ve değerlendirilmesinde diğeri ölçme araçlarından ayrılmaktadır.

Kavram haritaları, insanın görsel imge kapasitesini ortaya koyarak anahtar kavramların hem hiyerarşik hem de çapraz bağlantılar şeklinde açık ve öz bir şekilde gösterilmesini sağlayan bir üstbiliş stratejidir (Novak, 1990). Ülgen (2004), öğrencilerin kavram oluşturmalarını, kazanmasını ve değerlendirilmesini üç aşamada tanımlamaktadır. İlk aşamada öğretmenler konuyu öğrenciyle paylaşmadan önce konunun kavram haritasını çıkararak öğrencinin belleğinde bilginin anlamlı bir şekilde nasıl yapılabileceğini kestirebilir ve öğrencilere sunacakları uyarınları buna göre düzenleyebilirler. İkinci aşamada kavram haritaları öğrencilerin belirli ölçütlere dayalı genelleme yapabilmelerini sağlayarak kavram kazanmalarına destek olur. Son aşamada ise öğrencilerin geliştirdikleri kavram haritaları kavramlar arası ilişkiler ve etkileşimler açılarından incelenir ve değerlendirilir. Öğrenciler tarafından geliştirilen kavram haritaları özellikle kavramların derinliğine ve genişliğe dair ipuçları sunmaları nedeniyle kavram öğrenme düzeyini belirlemede etkili bir araç olarak kullanılabilir.

2.1.2.2. Programlama Performansının Ölçülmesi

Performans ölçümü, bir öğrencinin belirli bir görevle etkileşimi hakkında veri sağlayan ve bütün performansın özetini veren bir modeldir (Effenberger ve Pelánek, 2019). Performans ölçümünde psikometrik ya da performansa dönük testler, yordamaya yönelik ölçme çalışmalarının doğru sonuçlar üretebilmeleri öğrencilerin sahip oldukları becerilere bağlı olduğundan yordamaya yönelik ölçme araçlarına oranla daha çok tercih edilmektedir (Korkmaz, 2019). Bu nedenle programlamaya yönelik performansın belirlenmesinde açık uçlu ya da çoktan seçmeli başarı testleri (Estácio ve diğ., 2015; Restrepo-Calle ve diğ., 2018) ve tamamlayıcı ölçme-değerlendirme araçları (Lee ve diğ., 2021; Seppälä, 2012) kullanılmaktadır.

Geleneksel ölçme-değerlendirme tekniklerine göre hazırlanan ölçüte dayalı çoktan seçmeli testler sadece basit bilgi ya da becerileri ölçmekte, üst bilişsel becerileri incelemede yetersiz kalmakta (Demir ve diğ., 2011) ve öğrenme süreci hakkında yeterince veri sunmamaktadır (Jenkins ve Pany, 1978). Bir öğrencinin programlama becerisini kazanmasının üst düzey düşünme becerilerine sahip olmasına bağlı olduğu düşünüldüğünde, başarı testlerinin öğrenci öğrenmeleri hakkında ne kadar veri sunabileceği tartışılabilir.

Tamamlayıcı ölçme-değerlendirme yaklaşımları ise öğrencileri eleştirel düşünmeye ve karmaşık problemler karşısında çözüm üretmeye motive eder (Duban ve Küçükyılmaz, 2008). Tamamlayıcı ölçme-değerlendirme, performans değerlendirme ve otantik değerlendirme süreçlerini kapsamaktadır (Arslan ve Aytaç, 2010). Performans değerlendirmede öğrencilerin üst bilişsel becerileri değerlendirilirken, otantik değerlendirmede yaşam boyu kullanabilecekleri becerileri kazanmaları esas alınmaktadır (Coblentz ve diğ., 2018).

Programlama öğretiminde amacın, öğrencilerin farklı düşünme ve problem çözme becerilerini geliştirmek olduğu düşünüldüğünde, öğrencilerin programlama performanslarını ölçmede tamamlayıcı ve geleneksel ölçme araçları birlikte kullanılmaktadır (Korkmaz, 2019). Yanı sıra kullanılabilirlik testleri (usability testing) programlama sürecinin uygulamaya dayalı olması ve genellikle bir ürün geliştirme süreci barındırması nedeniyle (Derisma, 2020; Ontanon ve diğ., 2017) programlama performansının değerlendirilmesinde tercih edilebilmektedir.

2.1.2.3. Yordayıcı Beceriler

Alanyazında programlama becerisini yordayan beceriler olarak problem çözme, algoritmik düşünme ve bilgi işlemsel düşünme ön plana çıkmakta ve bu becerilere bilgisayar bilimi araştırmalarında sıklıkla yer verilmektedir (ör. Nouri ve diğ., 2020; Tikva ve Tambouris, 2021; Türker ve Pala, 2020). Problem çözme becerisi çevreye uyum sağlamayı kolaylaştıran bir beceridir (Gunawan ve diğ., 2017). Gerçek yaşam problemlerini çözmek için mantığı kullanma yeteneğidir ve farklı durumlarda farklı stratejiler kullanarak çalışabilir olanları belirlemektir (Butterworth ve Thwaites, 2013). Problem çözme ile programlama becerisi yakın ilişkilidir (Michaelson, 2015) ve küçük yaşlardan itibaren programlama öğretimiyle günlük yaşam problemlerini programlama aracılığıyla çözebilen bireylerin yetiştirilmesi amaçlanır (Kukul ve Gökçearslan, 2014). Öğrencilerin günlük yaşam problemlerini bilişim teknolojileri ve programlama aracılığıyla çözmeleri için son yıllarda yapay zekâ ve makine öğrenmesi gibi konular da programlama öğretimiyle birlikte gündeme gelmeye başlamıştır.

Programlamanın algoritmik düşünme becerileri üzerinde de etkisi vardır. Algoritmik düşünme, matematiksel varsayımları daha açık ve somut hale getirebilmektir (Futschek ve Moschitz, 2011). Bilgisayar biliminin temel kavramlarından birini oluşturur ve sürdürülebilir bilgisayar bilimi eğitimi algoritmik düşünmeyi temele alır (Hromkovic ve diğ., 2016). Algoritmik düşünme, programlamaya yeni başlayanların programlamanın sözdizimine (syntax) ve anlambilimine (semantics) yönelik öğrenme güçlüğüne gidermede (Malik ve diğ., 2019) ve öğrencilere karmaşık sistemleri açıklamada kullanılmaktadır.

Programlama becerisinin bir diğer yordayıcısı da bilgi işlemsel düşünmedir. Bilgi işlemsel düşünme becerisini ilk gündeme getirenlerden Wing'e (2006) göre bilgi işlemsel düşünme bilgisayar biliminin temel kavramlarını kullanarak problem çözme, sistem tasarlamayı ve insan davranışını anlamayı içermektedir. Bilgi işlemsel düşünme becerisini oluşturan bileşenlere dair tam olarak bir uzlaşısı bulunmamaktadır. Çetin (2018), bu bileşenleri algoritmik düşünme, değerlendirme, problem çözme, bileşenlerine ayırma, soyutlama, örüntü tanıma ve genelleme; Üzümcü ve Bay (2018) problemi anlama, parçalara ayırma, soyutlama, örüntü tanıma, algoritma ve değerlendirme; Kalelioğlu ve Gülbahar (2015) ise soyutlaştırma, algoritma tasarımı,

otomasyon, veri toplama, veri çözümeleme, veri sunma, ayrıştırma, eş zamanlı çalışma, örüntü tanıma, örüntü genelleme ve modelleme olarak tanımlanmaktadır. Dolayısıyla bileşenlere ayırma, soyutlama, örüntü tanıma ve genelleme, algoritma ve değerlendirmenin bilgi işlemsel düşünme becerisini oluşturan bileşenler olduğu söylenebilir.

Bileşenlerine ayırma, bilgisayar biliminde özellikle problem çözmede kullanılan temel becerilerdendir. Programlamada büyük kod blokları genellikle işlevsel birimlere bölünerek yazılmakta ve problemleri bileşenlerine ayırmak program yazmayı kolaylaştırmaktadır (Rich ve diğ., 2018). Kodların anlamlı işlevsel birimlere bölünmesi öğrencilerin kodları daha kolay anlamalarını sağlayacağından programlama öğretimine katkı sağlamaktadır.

Soyutlama, gerçekleştirilen işlemleri zaman ve mekânın fiziksel boyutlarının ötesinde açıklayarak hangi ayrıntıların vurgulanması gerektiğine ve hangilerinin göz ardı edilebileceğine karar verilmesinde yardımcı olmaktadır (Wing, 2008). Bilgisayar bilimi ve matematikte bir kavramın soyutlanmasının en yaygın yolu bütün ortak özelliklere sahip bir yapı veya kategori oluşturmaktır (Çetin ve Dubinsky, 2017). Bu nedenle soyutlama, öğrencilerin programlama sürecine ait tüm aşamaları dikkatlice gözden geçirmelerini, not etmelerini ve karar vermek için veriler üzerinde yoğunlaşmalarını gerektirmektedir.

Örüntü tanıma ve genelleme öğrenciliğin ilk yıllarında karşılaşılan şekiller, çarpma işlemi ve sayı serileri gibi matematik konuları nedeniyle tanıdık gelebilmektedir (Fadiana ve diğ., 2018). Bilgi işlemsel düşünmede bu durum problem çözme boyutuna doğru genişlemektedir. Grover ve Pea'ya (2018) göre bilişim teknolojileri açısından örüntü, geçmiş problemlerin çözümlerinin incelenmesini içermekte ve bu nedenle makine öğrenmesi ve yapay zekâ çalışmaları örüntü ve genellemeye dayanmaktadır. Yapay zekâ çalışmalarında amaç eldeki verileri analiz etmek, anlamlı birimler oluşturmak ve ortaya çıkan anlamlı birimleri bir problemin çözümünde kullanmak olduğundan öğrencilerin yapay zekâ programlama çalışmalarında örüntü ve genellemelerden yararlanmaları ve amacına uygun olarak kullanmaları gerekmektedir.

Algoritma, bilgisayar biliminin önemli bir parçasıdır ve bütün algoritmalar bir problemin çözümü için tasarlanmaktadır (Schneider ve Gersting, 2018). Bazı problemler bir defalık çözülürken, bazılarının çözümü diğer problemlerin çözümünde kullanılabilir. Tekrar eden problemlerde algoritma kullanımı bir sistemin geliştirilmesine ve sistematik çözümler uygulanmasına yardımcı olmaktadır.

Değerlendirme oluşturulan algoritmanın, sürecin ya da sistemin amacın gerçekleştirilmesinde ne kadar iyi bir seçenek olduğuna emin olmakla ilgilidir (Csizmadia ve diğ., 2015). Değerlendirmede zaman, mekân, güç ve depolama konularına ayrıntılı olarak yer verilmesi (Wing, 2006), ürün ya da süreçlerin yeniden gözden geçirilmesinde yararlı olacağından uygun değerlendirme tekniklerinin kullanımı önem kazanmaktadır.

2.1.3. Türkiye’de Programlama Öğretimi

Programlama öğretimi, programlama ile ilişkili kavramların öğretiminin yanında birçok düşünme becerisinin de geliştirilmesinde etkilidir (Yükseltürk ve Altıok, 2016). Üst düzey beceriler gerektiren zor, zahmetli ve sıkıcı bir süreç olarak tanımlanması nedeniyle öğrencilerin ilgisi uzun süre programlamadan uzak kalmıştır (Demirer ve Sak, 2016). Programlama öğretimini zor ve sıkıcı olmaktan kurtarmak ve üst düzey düşünme becerilerinin gelişimine katkıda bulunmak adına alanyazında birçok araştırmaya (Erümit ve diğ., 2020; Jumankuziev, 2023; Pérez-Marin ve diğ., 2020) rastlanabilir. Türkiye’de bu araştırmalar özellikle blok tabanlı (Aydoğdu, 2020; Yılmaz ve Üstün, 2021), fiziksel programlama araçları (Ayverdi ve diğ., 2023; Yurdakök, 2022) ve bilgi işlemsel düşünme becerisi (Cücü ve Dağ, 2023; Ramazanoğlu, 2021) üzerinde yoğunlaşmış durumdadır.

Blok tabanlı programlama, programlama öğretimini kolaylaştırmak ve daha zevkli hale getirmek için kullanılan programlama yaklaşımlarından biridir. Programlamayı görsel hale getirmeyi amaçlayan Scratch, programlama öğretiminde en çok kullanılan blok tabanlı görsel programlama aracı olarak dikkat çekmektedir (Eryılmaz ve Deniz, 2019). Scratch’in baskın olarak kullanılmasının nedenleri Scratch’in öğrenen motivasyonuna olumlu katkısı, kullanıcı dostu olması, basit kullanım kolaylığı ve kullanıcıların kendilerine özgü materyaller üretebilmeleridir (Çatlak ve diğ., 2015; Zhang ve Nouri, 2019). Scratch, temelini yapılandırmacı

(constructivist) öğrenme ve logo projesinden almakta ve programlama sırasında ses ve resim dosyalarını programa dahil ederek interaktif oyunlar, animasyonlar ve simülasyonlar oluşturulabilmektedir (Saez-Lopez ve diğ., 2016). Bu özellikleri nedeniyle program yazmaktan çok programın temel kavramlarının öğrenilmesini kolaylaştırmakta ve programlama mantığının daha somut ve görsel destekli olarak anlaşılmasına katkıda bulunmaktadır.

Fiziksel programlama, yazılım ve donanım kullanarak dış dünyayla veri alışverişi yapan sistemler tasarlama işidir (Kuzu ve Türk, 2018). Fiziksel olarak programlanabilen eğitsel robotlar, programlama öğretimi ve STEM projelerinin geliştirilmesinde oldukça sık kullanılmaktadır. Teknolojinin gelişimiyle robot eğitime verilen önem artmış ve birçok firma robot eğitim seti üretmeye başlamıştır (Arocena ve diğ., 2022). Üretilen bu eğitsel robotların amaç doğrultusunda programlanabilmesi için makine dili, metin tabanlı ya da blok tabanlı programlama seçeneklerinden birini kullanmaları gerekmektedir. Öğrenciler robot programlama etkinlikleri ile genellikle günlük yaşam problemlerine çözüm aramakta, programlamanın soyut kavramları aracılığıyla işlevsel somut bir çıktı üretmektedir.

Blok tabanlı ve fiziksel programlamayla birlikte Türkiye’de programlama öğretimi açısından incelenen bir diğer araştırma alanı bilgi işlemsel düşünme becerisidir. Araştırmada bilgi işlemsel düşünme becerisine, programlama becerisinin ölçülmesinde yordayıcı beceriler bölümünde ayrıntılı olarak yer verilmiştir.

Türkiye’de programlama öğretiminde karşılaşılan sorunlara çözüm için pedagojik yaklaşımlar da kullanılmaktadır. Bu yaklaşımlardan biri de yapılandırmacı yaklaşımdır. Programlama öğretiminde yapılandırmacı yaklaşım programlamanın yapısında problem çözme sürecini barındırması, öğrencilerin sorunlara çözüm olacağı düşünülen bilgiye bağlam içerisinde aktif olarak ulaşmaları, uygulama sırasında üst düzey düşünme becerilerinin kullanımını gerektirmesi ve öğrenci performanslarını değerlendirmede öğrenme süreçlerini temele alması nedenleriyle tercih edilmektedir.

Ters-yüz eğitim son yıllarda programlama öğretiminde sıklıkla tercih edilen bir diğer pedagojik öğretim yaklaşımıdır. Eğitimin özünü okul dışında kapsamlı notların, videoların, kaydedilmiş derslerin ve diğer uygun eğitim araçlarının öğrencilerle paylaşılması, okulda ise öğrencilerin işbirlikli etkileşimli etkinlikler gerçekleştirilmesi

oluşturmaktadır (Butt, 2014). Alanyazında ters-yüz eğitimin öğrenci başarısında ve derse etkin katılımı arttırdığına ilişkin birçok araştırmaya (ör. Alper ve Öztürk, 2019; Perçin, 2019) rastlanmaktadır. Programlama öğretiminde sıklıkla kullanılan proje tabanlı öğrenme yaklaşımıyla birlikte kullanıldığında öğrencilerin yoğun bilişsel çaba gerektiren programlama etkinliklerini sınıf içi zamanla sınırlı kalmadan gerçekleştirmelerini sağlayabilmektedir.

Türkiye’de örgün eğitimde programlama öğretimi ilkökul, ortaokul ve lise düzeyinde hazırlanan Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programlarında belirlenen çerçeve dahilinde gerçekleştirilmektedir. 2023 yılında yayımlanan Ortaöğretim Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı (MEB, 2023) bilgi işlemsel düşünme becerisi, bilimsel süreç, bilgi işlemsel süreç, üst bilişsel süreç, veri tanıma-işleme-analiz ve 21. yüzyıl becerileri gibi alan becerilerine odaklanmaktadır. Programda, Programlamaya Giriş ve Algoritma modülü sonrası Robotik Kodlama, Programlama Dilleri ve Mobil Uygulama modüllerinden biri seçilebilmekte Programlama Dilleri modülü seçilmişse Yapay Zekâ modülüne geçilebilmektedir. Öğretim programında modüllerden de anlaşılabacağı üzere programlama ile elektronik ürün geliştirme sürecine vurgu yapılmakta ve kazanımlara yer verilmektedir. Bunun yanında 2024 yılında yayımlanan Ortaöğretim Matematik Dersi Öğretim Programı (Hazırlık, 9, 10, 11 ve 12. sınıflar) 9. ve 10. Sınıf düzeyinde “Algoritma ve Bilişim” temasına yer vermektedir. Bu temayla algoritmik düşünme ve matematiksel düşünme ilişkisinden hareketle matematiksel kavramların algoritma diliyle ifade edilerek kazandırılması amaçlanmaktadır (MEB, 2024).

2.2. Metafor ve Eğitimde Kullanımı

2.2.1. Metafor

Kelime anlamı itibariyle metafor birçok anlamda kullanılmaktadır. Bunun temel nedeni kavramın sonsuz bir anlam evrenine yol açabilecek olan taşıma (bir anlamı ötekine gönderme) anlamına karşılık gelmesidir (Güneş ve Tezcan, 2019). Metafor, Yunanca metaferinden (“meta” değiştirmek, “pherein” taşımak) gelmektedir (Levine, 2005) ve oldukça soyut, karmaşık ve kuramsal bir olguyu açıklamak için güçlü bir zihinsel araç olarak tanımlanmaktadır (Çelenk, 2019). Ayrıca metafor, soyut ve anlaşılması güç olguların tanınan ifadelerle anlatılması, iki nesne ya da kavramı

birbirine bağlayan dilsel bir araç olarak ele alınmaktadır (Şeyihoğlu ve Gencer, 2011). Lakoff ve Johnson’a (2015) göre metafor bir şeyi başka bir şeye göre anlamak ve tecrübe etmektir. Sadece gündelik hayatta dille ilgili değil, düşünce ve eylemde de oldukça yaygındır. Bireylerin yaratıcılıklarının, gözlemlerinin, tecrübelerinin, bilgi birikimlerinin, kültürel değerlerinin oldukça değerli ürünleridir (Ekici, 2016). Anlama kattığı güç yanında anlam çeşitliliği nedeniyle de doğru kullanımı oldukça dikkat ve derinlemesine bir anlayışı gerektirmektedir.

Türk Dil Kurumu (TDK) (2023) güncel çevrim içi Türkçe sözlüğü metaforu “mecaz” olarak Türkçeleştirmiştir. Aynı sözlük mecazı, “Bir kelimeyi veya kavramı kabul edilenin dışında başka anlamlara gelecek biçimde kullanma, metafor” olarak tanımlamaktadır. Cambridge çevrim içi sözlüğü ise metafor sözcüğünü “bir şeyi aynı niteliklere sahip başka bir şeyle karşılaştırarak tanımlamanın bir yolu” olarak tanımlamaktadır (Cambridge Dictionary, 2023). Türkçe’de metafor, istiare, eğretilme/iğretilme, aktarma, ödüncleme, mecaz aktarma gibi değişik sözcüklerle karşılanmaya çalışılmaktadır (Demir, 2009). Genel olarak sosyal bilimler alanında yaygın bir kullanıma sahip olan metafor; sosyoloji ve felsefede analogi; edebiyat ve dilbilim alanında eğretilme, mecaz, ödüncleme, istiare; eğitimbilim alanında ise daha çok benzetme anlamında kullanılmaktadır (Zeren ve Yapıcı, 2014). Yine de bunların hiçbirisi tam olarak metaforik düşüncüyü açıklayamadığından metafor kavramının tercih edildiği görülmektedir. Metaforu daha iyi anlamak adına metaforun bazı özelliklerini almış ve metaforu bir üst kavram olarak kabul eden analogi, metonimi, sinekdoka ve kişileştirme kavramlarından kısaca bahsetmek yararlı olabilir.

20. yüzyılın ortalarından beri, filozoflar metafor ve analoginin tüm söylemlere nüfuz ettiğini, insan düşüncesinin temelini oluşturduğunu ve zihinsel sıçramalara neden olduğunu kabul etmişlerdir (Aubusson ve diğ., 2006). Metaforlar, daha çok kültürel ve kişisel deneyimlerle oluşturulurken, analogide kişisel tecrübelerle daha az yer verilmektedir. Örneğin, bir gruba barış kavramı ile ilgili metafor oluşturmaları istendiğinde büyük çoğunluk beyaz güvercin diyorsa aynı zamanda bir analogi de oluşmuştur (Özelmacı ve Çakır, 2019). Analogi için insan zihnindeki kalıplaşmış metaforlar da denilebilir.

Metonimi ise metaforun alt türü olarak tanımlanabilir. Metaforlar alanlar arası eşleşmeler içerirken metonimi, tek bir alan veya alan matrisi içindeki eşleşmeleri içermektedir (Özer, 2017). Sinekdoka ise metoniminin bir alt türü olarak kabul edilmektedir ve genel olarak parça ve bütün ifadelerini kullanmaktadır. Sinekdokada kategori (category) aktarımı söz konusu iken metonimi de varlık (entity) aktarımı söz konusudur (Seto, 1999). Bu nedenlerle metonimi ve sinekdoka metaforlara göre daha dar kapsamlı aktarımlar içermektedir.

Kişileştirme ise ontolojik metaforlardır ve çok değişik türde insan dışı varlıklarla olan deneyimin insani motivasyonlara, niteliklere ve etkinliklere göre kavranmasına imkân veren yapılardır (Lakoff ve Johnson, 2015). Edebi anlatılarda anlatımı daha etkili hale getirmek ve metinde geçen nesnelerle empati kurabilmek için oldukça sık kullanılan yöntemlerdendir. Kişileştirmede metaforun temelini oluşturan taşıma görevi nesne üzerinden gerçekleştirilmektedir.

Analoji, metonimi, sinekdoka ve kişileştirme gibi kavramların metaforla olan ilişkisi çok doğrusal olmadığından bu ilişkileri tanımlamak amacıyla birtakım kuramlar ortaya atılmıştır. Bu kuramlar arasında Lakoff ve Johnson'ın Kavramsal Metafor Kuramı günümüze kadar gelen ve en çok bilinen metafor kuramlarından biri olmuştur.

2.2.2. Kavramsal Metafor Kuramı

Kavramsal Metafor Kuramı ilk kez George Lakoff ve Mark Johnson'ın 1980 yılında yayınlanan "Metaphors We Live by" adlı eserinde dile getirilmiştir ve bu tarihten itibaren kuram George Lakoff, Zoltan Kövecses, Mark Johnson, Mark Turner ve Joseph Grady gibi isimlerin çalışmalarıyla zenginleşmiştir (Çiçekler ve Aydın, 2019). Ruiz de Mendoza Ibanez ve Pérez Hernandez, (2011) metaforun retorik bir figür olarak dilin becerikli bir şekilde kullanımı için bir araç olarak anlaşılmasında bir dil değil, biliş meselesi olduğunu savunmuşlardır. Dolayısıyla metaforun dil ve biliş üzerindeki etkisi tartışmaları klasik ve çağdaş metafor yaklaşımlarının oluşmasına neden olmuştur.

Klasik metafor yaklaşımlarında metafor bir dil meselesi olarak görülüyorken metaforik ifadelerin sıradan günlük dil ile sınırlı olduğu varsayılmış ve bu kuram sadece doğru olarak değil, tanımlayıcı olarak da kabul edilmiştir (Çiçekler ve Aydın,

2019). Aslında dilde yer alan metaforları yöneten genellemeler düşüncededir ve çağdaş metafor anlayışında metafor kelimesi, kavramsal sistemde etki alanları arası eşleşme anlamına gelmektedir (Lakoff, 1993). Çağdaş metafor kuramı, kavramsal metafor kuramı olmadan anlatılamamaktadır (Hampe, 2017). Lakoff ve Johnson (1980) kavramsal sistem ile metafor ilişkisini şöyle açıklamaktadır:

Düşüncemizi yöneten kavramlar sadece aklın meseleleri değildir. Bu kavramlar aynı zamanda günlük işlerimizi en sıradan ayrıntılara kadar yönetiyorlar. Kavramlarımız, algıladıklarımızı, dünyada nasıl dolaştığımızı ve diğer insanlarla olan ilişkimizi yapılandırır. Böylece kavramsal sistemimiz günlük gerçeklerimizin tanımlanmasında merkezi bir rol oynar. Kavramsal sistemimizin büyük ölçüde mecazi olduğunu düşünmekte haklıysak, o zaman düşünme şeklimiz, deneyimlerimiz ve her gün yaptığımız şey bir metafor meselesidir (s. 454).

Lakoff ve Johnson (2015) metaforları yapısal, ontolojik ve yönelim metaforları olmak üzere üç şekilde ele almaktadır. Yapısal metaforlar bir kavramın diğerine göre mecazi olarak yapılandırıldığı örneklerdir. Bunlar genellikle bir alandan bir kavramın başka bir alandan entellektüel bir kavramı kullanmasını içermektedir (Lakoff ve Johnson, 1980). Yapısal metaforlar kaynak etki alanından hedef etki alanına zengin bir bilgi yapısı sunarak kaynak a'nın yapısı aracılığıyla hedef b'nin anlaşılmasını sağlamaktadır (Kövecses, 2010). Örneğin savaş, sık sık tartışma ve sevgi gibi hedef alanlarla eşleşen soyut bir kavramdır ve bu nedenle de savaştan bahsederken tartışma ve sevgi hakkında da konuşulur (Lakoff ve Johnson, 1980). Herkes savaşın plan, saldırı, savunma, karşı saldırı, kavga, kazan, kaybet, ateşkes vb. içeren çok karmaşık bir süreç olduğunu bilir ve bilinmeyen soyut kavramlar ise savaş hakkında konuşarak tanımlanabilir (Li, 2010). Yine de tüm metaforlar birini diğerine göre anlamak için kullanılmamaktadır.

Ontolojik metaforlar daha çok çevreyle ilişkili deneyimlerin tanımlanmasına yardımcı olurlar ve fiziksel nesnelere ya da evreni oluşturan asıl maddeye ilişkin konuşabilmeye, kategorize edebilmeye ve gruplayabilmeye olanak sağlarlar (Lakoff ve Johnson, 2015). Örneğin, “zaman bir varlıktır”, “yeterli zamanım yok” ya da “bana zaman ayırdığınız için teşekkürler” gibi ifadeler kullanıldığında bu gözlemlenebilir ve zamanın anlaşılmasını ve özelliklerinin nicelleştirilebilmesini sağlar (Barr ve diğ., 2002). Böylece sadece kavramların tanımlanması değil, aynı zamanda bulunan çevrenin de anlaşılması sağlanmış olur.

Yönelimsel metafor, bir kavramı bütün bir kavram sistemine göre organize eder ve çoğu uzay/mekân istikameti ile ilişkilidir (Lakoff ve Johnson, 1980). Yapısal metaforlar kavramsal yapılar hakkında daha zengin bir yapı sunarken anlamda nasıl ve ne zaman kullanılacağı her zaman net değildir. Yönelimsel metaforlar, kavramsal metaforlara göre daha kısıtlı kapsama sahip olduğundan daha erişebilir veriler sunulabilmektedir (Langston, 2002). Lakoff ve Johnson (2015) yönelim metaforlarına yönelik birçok örneğe yer vermektedir:

Mutlu olan yukarıda; kederli olan aşağıdadır... Kendimi yukarıda hissediyorum. Bu moralimi çok yükseltti. Ruhum kanatlandı. Moralim yüksek. Ne zaman onu düşünsem ayaklarım yerden kesilir. Düşmüş hissediyorum. Dozer altında kalmış gibiyim. Bugünlerde gerçekten hayatı tepetaklak gidiyor. Derde düştüm. Ruhen dibe vurduğum (s. 41).

Özetle yapısal metaforlar kaynak ve hedef alanlar arası aktarımlar, ontolojik metaforlar bulunduğumuz çevreyle olan ilişkimiz ve yönelimsel metaforlar zaman/mekân istikameti aracılığıyla anlam dünyamıza katlı sağlamaktadır.

2.2.3. Eğitimde Metafor Kullanımı

Metaforlar her yerdedir ve insanları yaşamları boyunca kuşatırlar (Ahlgren ve diğ., 2021). İnsanoğlunun düşünme şeklinin modellenmesi, soyutlanması ve algılanması arasında ilişkiler kurarak (Lakoff ve Johnson, 1980) bireyi yaratıcı düşünmeye ve hayal etmeye teşvik ederler. Metaforlar felsefe, dilbilim, edebiyat, sosyoloji, psikoloji, iletişim, hukuk, siyaset, sanat, mimari, işletme ve eğitim gibi pek çok alanda olayları daha anlaşılır hale getirdiği için tercih edilmektedir (Akyol, 2019; Low, 2008). Özellikle pedagojinin temelini oluşturan kuramların, görüşlerin ya da felsefi varsayımların güçlü destekleyicisidir (Botha, 2009) ve öğrenmeye bakışı açıklamada oldukça etkilidir.

Eğitimde metaforlar genellikle normalde zor olan bir kavram için dikkat çekici ve akılda kalıcı bir etiket bulmak, soyut ve karmaşık olan kavramı açıklığa kavuşturmak ve kavrama dair var olan düşünceyi genişletmek amaçlarıyla kullanılmaktadır (Low, 2008; Saban, 2004). Böylece metaforlar kavramların daha iyi anlaşılmasına yardımcı olmakta, duyguları tetiklemekte ve kullanılan günlük dili zenginleştirmektedir (Ahlgren ve diğ., 2021). Soyut kavramların geliştirilmesine ve kullanımına rehberlik eden bir araç olmanın yanında (Jamrozik ve diğ., 2016) zihinsel gelişime, eleştirel düşünmeye ve problem çözmeye de katkı sağlamaktadır (Low,

2008). Metaforların eğitime olan çeşitli katkılarına rağmen özellikle öğrenilmesi zor kavramların daha açık ve anlaşılır hale getirilmesine odaklanılmaktadır.

Bir kavramın ne olduğunu ve nasıl öğrenildiğini anlamak psikoloji, dilbilim ve diğer pek çok alanda araştırma ve tartışma konusu olmuş (Danesi, 2007) ve kavram öğretimine yönelik tartışmalarda metaforlar her zaman önemli bir yer tutmuştur (Demirbilek ve Korkmaz, 2021). Metaforlar kavram öğretiminde öğrencilere erişilmesi, incelenmesi, değerlendirilmesi ve sorgulanması zor olan konularda kavramları daha açık hale getirmek için bir dil aracı olarak görev yaparlar (Thomas, 2006). Kavramsallaştırılması zor bir kavramın özelliklerini, öğrencilerin ön bilgilerindeki daha erişilebilir ve tanıdık başka bir kavram açısından inceleyerek onlara yardımcı olurlar (Ekici, 2016; Thomas ve McRobbie, 2001). Metaforun değişim yaratan bir ajan olarak kullanılması öğrencilerin bildiklerini yeni anlayışlara dönüştürmelerine yardımcı olmaktadır (Levine, 2005). Nasıl ki bir resim 1000 kelimeye bedel ise, bir metafor 1000 resme bedeldir ve bir şeyler hakkında düşünürken resim durağan bir imge sunarken, metafor kavramsal bir çerçeve sunmaktadır (Shuell, 1990). Metaforların kavramsal sistem üzerindeki etkisi göz önüne alındığında eğitim sisteminde rolü olan her bireyin metaforlar üzerinde düşünmesi gerekmektedir (Ahmady ve diğ., 2016). Metaforların eğitimde kullanımına kavram öğretimiyle sınırlı kalmadan beceri geliştirmede, öğretimi farklılaştırmada ve öğretmen eğitiminde kullanımına dikkat çekilebilir.

Metaforlar, eğitimde öğrencilerin bir konuyu anlamalarını ve kavramları içselleştirmelerini kolaylaştırmanın yanında konuşma becerilerinin gelişimini teşvik etmek için de kullanılmaktadır (Cortazzi ve Jin, 1999). Sadece dilin konusu değil aynı zamanda öğrencilerin öğrenme davranışlarını dikkate almalarına yardımcı bir kaynak ve üstbilişlerini geliştirmenin bir aracıdır (Thomas ve McRobbie, 2001). Metaforlar ayrıca öğrencilerin eleştirel düşünme, sorgulama ve problem çözme becerilerini geliştirmede etkilidir (Skorczynska, 2014). Birçok beceriye katkısının yanında öğrencilerin öğrenmeye bakış açılarını ortaya koymada ve öğrencilerin üstbiliş becerilerinin geliştirilmesinde kullanılabilir.

Öğretimi farklılaştırmak açısından metaforlar, genelliği tek bir açıdan ele aldıkları ve çeşitliliğe saygı gösterdikleri için farklı ilgi alanlarına, karakterlere ve

geçmişlere sahip öğrenciler tarafından tercih edilmektedir (Ahmady ve diğ., 2016). Öğrencilerin metaforları ile inanç sistemleri arasındaki varsayılan bağlantı göz önüne alındığında, farklı metaforlar kullanan öğrencilerin zihinsel modelleri de farklı olmaktadır. Benzer şekilde farklı öğrenme bağlamlarındaki öğrencilerin zihinsel öğrenme modelleri farklılık göstereceğinden öğrencilerin öğrenme metaforlarının da farklılık göstermesi beklenmektedir (Wegner ve diğ., 2020). Bu nedenle öğrenme ortamındaki farklılıkları tespit etmede ve eğitimi tespit edilen farklılıklara göre düzenlemede metaforlar etkili birer araca dönüşebilmektedir.

Öğretmenler tarafından uygun görülen metaforlar; öğretmenlerin öğretim yöntemi seçimlerini, etkinliklerini ve planlamalarını etkilemektedir (Csorba, 2015). Diğer bir anlatımla, öğretmeye ilişkin metaforlar, öğretmenlerin öğretmeye dair algılarını yansıtmaktadır (Ahmady ve diğ., 2016). Bu nedenle metaforun eğitim araştırmalarında, özellikle öğretmenlerin bireysel performanslarının geliştirilmesinde oynayacağı rol incelemeye değerdir (McCandless, 2012). Metaforlar aynı zamanda öğretmenliğe yeni başlayanların eğitiminde pedagojik bir araç olarak kullanılabilir ve öğretmenlerin gelişimi için geniş olanaklar sunan bir fırsat olarak değerlendirilmektedir (Lynch ve Fisher-Ari, 2017). Öğretim süreçlerinde kullanacağı uygun metaforu belirleyen, bunları öğretim planlarına yansıtan ve etkili uygulama yeterliğine sahip öğretmenlerin bilgisayar bilimi derslerinde etkili öğrenme yaşantılarını oluşturabilecekleri söylenebilir.

2.3. İlgili Araştırmalar

Bu bölümde araştırmanın bağımlı değişkenleri dikkate alınarak programlama kavramlarının öğretiminde, bilgi işlemsel düşünme ve programlama becerilerini geliştirmede metafor kullanıma yönelik araştırmalarla eğitimde metafor kullanımına yönelik araştırmalara yer verilmiştir.

2.3.1. Programlama Kavramlarının Öğretiminde Metafor Kullanımı

Larsson ve Stolpe (2023a) lise düzeyinde ders veren üç ayrı öğretmenin konuşma ve jestlerinde kavramsal metafor kullanımını incelemeyi ve karşılaştırmayı amaçlamıştır. İsveç'te üç farklı belediyeye bağlı lise öğretmenlerinden elde edilen toplam 120 dakikalık video kaydındaki sözlü verileri Pragglejaz Grubunun (2007) Metafor Tanımlama Prosedürü'nün (Metaphor Identification Procedure, MIP) bir

uyarlamasıyla analiz etmiş, öğretmenlerin jestlerini keşfetmek için Cienki'nin (2016) Jestler için Metafor Tanımlama Yönergelerini (Metaphor Identification Guidelines for Gestures, MIG-G) kullanmışlardır. Ulaşılan sonuçlar, öğretmenlerin metaforları bir programlama kavramına uzamsal özellikler eklemek ya da bir programlama kavramı için ek görseller sağlamak amacıyla kullandıklarını göstermiştir. Ayrıca öğretmenlerin genel olarak soyut kavramları somut hale getirmek için metafor kullandıkları belirlenmiştir.

Larsson ve Stolpe (2023b) öğretmenlerin öğretimle ilgili bilgi ve inançları arasındaki etkileşimi ve öğretmenlerin bilgilerini nasıl hayata geçirdiklerini ortaya koymayı amaçlamıştır. Araştırmada ayrıca öğretmenlerin öğretim uygulamalarının örtülü unsurlarıyla, ilgili kavramsal metaforları arasında nasıl bir etkileşim olduğu da ortaya çıkarılmaya çalışılmıştır. Durum çalışması deseninde gerçekleşen araştırmaya üç öğretmen dâhil olmuş, öğretmenler İsveç sınıflarında beklenen çeşitliliği yansıtabilecek şekilde kişisel geçmişlerinden ziyade öğretim biçimlerine göre seçilmiştir. Veriler, öğretmenlerin sınıftaki eylem videoları ve pedagojik yansımalarını anlamak için Epizodik Anlatı Görüşmeleri (Episodic Narrative Interviews, ENI) kayıtlarından elde edilmiştir. Araştırmada üç temel bulguya ulaşılmıştır. İlk olarak üç öğretmen de programlamayı daha küçük kod parçalarının bir amaca ulaşabilmek için iç içe geçebildiği bir etkinlik olarak görmüştür. İkinci olarak öğretmenler kendilerini öğrencilerin başarıları için geleceğin ustaları, matematikçileri ve sistem geliştiricileri olarak görmüşlerdir. Son olarak da programlamayı öğrenme söz konusu olduğunda kod yazmanın önemli bir bileşen görülmediği ortaya çıkmıştır.

Larsson (2022) metaforik dilin bilgisayar programlamaya nasıl katkıda bulunduğunu belirlemek için İsveç'te yüksekokul düzeyindeki dört bilgisayar programlama ders kitabı ve çevrimiçi kaynakları metaforik açıdan incelemiştir. Veriler, dört farklı metin kaynağından (ikisi çevrimiçi eğitim ve ikisi ders kitabı) metaforları sistematik tanımlamada kullanılan Pragglez Grubunun (2007) Metafor Tanımlama Prosedürü (Metaphor Identification Procedure, MIP) aracılığıyla elde edilmiştir. Bulgular, öğretmenlerin kaynak seçiminin öğrencilerin bilgisayar programlama kavramlarını ve bilgisayarla ilişkilerini tanımlamada etkili olduğunu göstermiştir. Ayrıca, kaynakların programlama kavramlarını bilgisayar içinde

gerçekleşen insani eylemler olarak yapılandırma eğiliminde olduğunu ortaya koymuştur. Buna örnek olarak kaynaklardaki bilgisayarın “akıllı” olması, “belleğe” ya da “hesaplama” yeteneğine sahip olması örnek gösterilmiştir.

Fanny ve diğerleri (2020) düzenlenen iki ayrı robot eğitimi etkinliğinde öğretmen ve öğrenciler tarafından kullanılan yapısal ve mekânsal metaforları belirlemeyi ve bunların öğrenme üzerindeki rolünü analiz etmeyi amaçlamıştır. İki farklı durumun analiz edildiği çalışmaya ilk duruma 3-10 yaş arası 140 öğrenci, ikinci duruma ise 8-15 yaş arası 11 öğrenci katılmıştır. Veriler gözlem kayıtları, öğrenci ürünleri, ortam ses kayıtları ve süreçte alınan notlar aracılığıyla toplanmış; öğretmenlerin ve öğrencilerin söylemlerinde kullandıkları metaforlar yönelim ve yapısal metafor türlerine dikkat edilerek kayıt altına alınmıştır. Ortaya çıkan farklı metaforlar daha sonra etkileşimlerdeki rollere göre analiz edilmiş, analiz sonucu kullanılan metaforlar anlamaya yardımcı, somutlaştırıcı ve slogan işlevi gören olmak üzere üç grupta toplanmıştır. Bulgular, bir metaforun birden fazla rol alabileceği ve rolü her ne olursa olsun yaşayan ya da yaşamayan metafor olarak sınıflandırılabilceğini göstermiştir.

Numic’in (2020) tasarım tabanlı araştırmasında, yükseköğretim öğrencilerine programlama kavramlarını metaforlar aracılığıyla öğretmeyi hedefleyen bir araç geliştirilmiş ve bu aracın programlama kavramlarının öğretime etkisi araştırılmıştır. Çalışmada ilk olarak metafor temelli somut aracı geliştirmek için programlama kavramlarına yönelik önceki metafor belirleme çalışmaları incelenmiş, sonrasında araç tasarımını bu metaforları dikkate alarak oluşturmuştur. Geliştirilen aracın kullanışlılığına ilişkin veriler yedi yükseköğretim öğrencisinden yarı yapılandırılmış görüşmeler aracılığıyla toplanmıştır. Katılımcılardan biri dışındaki tüm katılımcılar geliştirilen tasarımı programlamaya yönelik kavramları öğrenmelerinde kolaylaştırıcı bulmuştur.

Chibaya (2019) metafora dayalı bir modelin tasarımını ve uygulanmasını değerlendirmiştir. Veri toplama işlemini iki aşamada gerçekleştirmiş, birinci aşamada daha önce herhangi bir programlama dersini tamamlamış 80 son sınıf lisans öğrencisinden bir anket yardımıyla programlama kavramlarına yönelik metaforlar geliştirmelerini istemiştir. Öğrencilerden gelen yanıtları analiz edilerek her bir kavram

için toplanan metaforları uygun kategorilere ayırmış, daha sonra ortak amaçlar doğrultusunda birleştirerek metafor temelli bir model oluşturmuştur. İkinci aşamada ise ilk kez programlama öğrencisi olan lisans düzeyinde 100'er kişilik iki grubu deney ve kontrol grubu olarak atamış, deney grubuna birinci aşamada geliştirdiği metafor temelli modeli uygulamıştır. Uygulama sonunda her iki gruba da geleneksel başarı testi uygulamış, bulgular geliştirilen modelin uygulandığı deney grubu lehine başarı testi puanlarında anlamlı bir fark göstermiştir.

Hermans ve diğerleri (2018) programlamanın temel kavramlarından olan değişken kavramının öğretiminde yaygın olarak kullanılan “kutu” metaforunu başka metafor seçenekleri ile karşılaştırarak metaforun kavram öğretimi açısından uygunluğunu değerlendirmiştir. Bu amaçla Hollanda'daki bir bilim müzesini ziyaret eden 496 kişiyle 14 günlük kontrollü bir deney gerçekleştirilmiş, deney süresince katılımcılara Programlamaya Giriş dersleri verilmiştir. Katılımcıların yarısına değişken kavramı kutu metaforuyla, diğer yarısına ise etiket metaforuyla aktarılmıştır. Öğretim sonunda katılımcılara değişkene ait ne anladıkları sorulmuş ve kayıt altına alınmıştır. Bulgular, kutu metaforunun etiket metaforuna oranla daha iyi bir performans elde ettiğini fakat kutu metaforu ile karşılaşan bireylerde kavram yanlışlığının daha çok görüldüğünü ortaya koymuştur.

Pérez-Marin ve diğerleri (2018) programlama kavramlarını öğretmede etkili olabileceğini düşündükleri metaforlar içeren senaryolar geliştirmiş, etkililiğini denemek için senaryoları İspanya'da bir ilköğretim okulunun dördüncü, beşinci ve altıncı sınıf 62 öğrencisine uygulamıştır. Uygulama sonunda öğrenci ve öğretmenlerden açık uçlu sorular aracılığıyla senaryolara yönelik veriler toplanmış, bulgular senaryolarda yer alan ve öğretimde kullanılan dört tür metaforun (yemek tarifi, bilgisayar ekranı, karar verici olarak bilgisayar ve tekrarlayıcı olarak bilgisayar) katılımcıların çoğu (%65) tarafından kullanışlı bulunduğunu göstermiştir.

Ontanon ve diğerleri (2017) bilgisayar bilimleri lisans programı için paralel programlama eğitimini gerçekleştirmede ve öğrencilerin paralel programlama problemlerini nasıl çözdüklerini ortaya koymada kullanılabilecek bir oyun geliştirmiş, paralel programlama ile oyun arasında daha anlaşılır bir yol bulmak için uygun metaforları belirlemeyi amaçlamıştır. Daha önce bilgisayar bilimi dersi ve paralel

programlama derslerini almış dokuz kullanıcıyı oyunun kullanıcı testlerine dâhil ederek onların oyuna ilişkin değerlendirmelerini almışlardır. Bulgular, kullanıcıların paralel programlama kavramları ile onlara karşılık gelen oyun kavramları arasındaki bağlantıyı görebildiklerini ve görsel soyut metaforların öğrencilerin öğrenme süreçlerini desteklediğini ortaya koymuştur.

Programlama kavramları öğretiminde metafor kullanımına yönelik araştırmalar incelendiğinde durum çalışması ya da deneysel desenin sıklıkla kullanıldığı görülebilir. Bu araştırmalarda görsel, işitsel ve jest/mimiğe dayalı çeşitli metaforların kullanım durumu ya da kavram öğrenme üzerindeki etkileri incelenmiş, kimi zaman bu metafor türleri arasından seçim yapılmasının, kimi zaman da bir denel işlemde senaryolar aracılığıyla uygulanmasının kavramları öğrenmede katkıları araştırılmıştır. Araştırmalarda metaforlar, öğretimde genellikle soyut kavramları somutlaştırmada kullanılmış ve metafora dayalı gerçekleştirilen öğretimin kavramları öğrenmede kolaylaştırıcı rolüne dikkat çekilmiştir. Metaforun farklı anlamları barındırabileceği göz önüne alındığında, programlama kavramlarının öğretiminde metafor kullanılırken oluşabilecek kavram yanılgılarına dikkat çekilmiştir.

2.3.2. Bilgi işlemsel Düşünme Becerisini Geliştirmede Metafor Kullanımı

Pérez-Marin ve diğerleri (2020) geliştirdikleri metafora dayalı öğretimi temele alan MECOPROG programının öğrencilerin bilgi işlemsel düşünme becerilerine etkilerini belirlemeyi amaçlamışlardır. Ön test-son test yarı deneysel desende tasarlanan araştırmaya 9-12 yaş aralığında 132 ilköğretim öğrencisi katılmış ve öğrencilere bir dönem boyunca MECOPROG programı uygulanmıştır. Veriler programlama kavramları testi ve iki adet bilgi işlemsel düşünme beceri ölçeği aracılığıyla toplanmış, bulgular öğrencilerin kavram öğrenme ve bilgi işlemsel düşünmelerinde anlamlı bir artışı ortaya koymuştur. Araştırma, MECOPROG programının ilköğretim öğrencilerinin bilgi işlemsel düşünme becerilerini geliştirmede olumlu etkisinin olduğunu göstermiştir.

Oliveira ve diğerleri (2019) Kültürel Bakış Açısı Metaforlarının (Cultural Viewpoint Metaphors, CVM) bilgi işlemsel düşünme öğretiminin analizinde, sınıflandırılmasında ve tanımlanmasında nasıl bir görev alabileceğini incelemiştir. Bu amaçla Brezilya'da ortaokul öğrencilerine bilgi işlemsel düşünmenin öğretilmesine

yönelik arařtırmalar CVM dikkate alınarak araçlar ve yöntemler bakımından analiz edilmiştir. Bulgular bilgi işlemsel düşünmenin öğretime yönelik araç ve yöntemleri sınıflandırmada CVM'nin temel bir araç olarak kullanılabileceğini ortaya koymuştur.

Moreno-León ve diğerleri (2018) doküman analizi aracılığıyla bilgi işlemsel düşünmeyi farklı açılardan inceleyerek, bu becerinin kapsamını belirlemeyi ve becerinin eğitimde kullanımına yönelik sonuçları içeren araştırma bulgularını özetlemeyi amaçlamıştır. Araştırma bulguları, son yıllarda okulların kodlama öğretimiyle bilgi işlemsel düşünme gibi becerileri geliřtirmelerinden çok programlama öğretime odaklandığını göstermiştir. Oysaki arařtırmada bulgulardan hareketle yapay zekâdaki ilerlemelerin ya da araçlarının desteklenmesi nedeniyle programlamanın birkaç yıl içinde ortadan kaybolabileceği, bu nedenle eğitimde bilgi işlemsel düşünme becerisine evrensel bir bakış açısı ile bakılması gerektiği vurgulanmıştır.

Parmar ve diğerlerinin (2016) bir sanal gerçeklik eğitim ve eğlence uygulamasının tasarımını, uygulamasını ve değerlendirmesini gerçekleřtirdikleri araştırmanın katılımcılarını kadınların bilim ve mühendisliğe katılımını teşvik etmeye yönelik bir yaz kampına katılan 54 kiři oluřturmuřtur. Katılımcılara öncelikle sanal bir dans performansı hazırlanmış, sonrasında 16 kişilik bir alt küme daha oluřturularak uygulamanın sürükleyici, somutlařtırılmış etkileşim metaforu uygulanmıştır. Metaforun katılımcıların bilgi işlemsel düşünmeye yönelik heyecanlarına ve ilgilerine etkileri incelenmiştir. Veriler altı soruluk bir anket ve çizim yapmalarını gerektiren bir yönerge aracılığıyla toplanmış, bulgular kişisel avatarları kullanarak somutlařtırılmış deneyimler saėlayan metaforun katılımcıların bilgi işlemsel düşünmeye yönelik heyecanlarına ve ilgilerine katkı saėladığını göstermiştir.

Bilgi işlemsel düşünme becerisine yönelik arařtırmalar incelendiğinde metafor kullanımının bilgi işlemsel düşünmeye etkisine yönelik oldukça sınırlı arařtırmaya rastlanmıştır. Bu arařtırmaların daha çok diğer araştırma bulgularının analizine dayandığı ve metaforların bilgi işlemsel düşünmeyi dolaylı olarak öğretmeyi amaçlayan araçların tasarlanmasına, bu araçların denenmesine ve uygulanmasına odaklandığı görülmüřtür. Sınırlı sayıdaki araştırma bulguları metafora dayalı

öğretimin bilgi işlemsel düşünme becerisi üzerinde olumlu etkisi olduğunu göstermiştir.

2.3.3. Programlama Becerilerini Geliştirmede Metafor Kullanımı

Farida ve diğerleri (2022) sadece STEM’de (Science, Technology, Engineering and Mathematics) ve STEM-Bilgisayar Bilimlerinin (STEM-CS) birlikte kullanımında öğrencilerin metaforik düşünme becerilerindeki farklılıkları tanımlamayı amaçlamıştır. Yarı deneysel desen kullanılarak gerçekleştirilen araştırmaya Endonezya’nın Bandar Lampung kentinden 93 ortaokul öğrencisi katılmış, öğrencilerden 32’si STEM ile öğretim gerçekleştiren birinci deney grubuna, 30’u STEM-bilgisayar bilimi dersini birlikte gerçekleştiren ikinci deney grubuna ve 31 öğrenci de geleneksel öğretim gerçekleştiren kontrol grubuna atanmıştır. 18 saat süren denel işlem sonrası gruplara anket uygulanmıştır. Bulgular hem STEM-Bilgisayar Bilimi grubunda metaforik düşünmenin bu dersleri öğrenmeye yardımcı olduğunu ortaya koymuştur.

Barmpoutis ve Huynh (2019) öğrencilerin metin tabanlı kodlama arayüzleriyle ilk karşılaştıklarında duygusal olarak ne hissettiklerini değerlendirmek amacıyla bir durum çalışması gerçekleştirmiştir. Araştırmada aynı zamanda öğrencilerin metin tabanlı kodlama arayüzleriyle ilk karşılaşmalarının olumsuz tutuma yol açabileceği düşünülmüş ve çıkartma ve simgeler gibi ifadelerden oluşan metafor tabanlı bir kod düzenleyici prototip geliştirilmiştir. Bu prototipin etkililiğini incelemek adına prototip beşinci sınıfa devam eden 40 ortaokul öğrencisiyle denenmiş ve öğrencilerin uygulamayı kullanımlarından elde edilen veriler prototip kullanılarak blok tabanlı programlamadan metin tabanlı programlamaya geçişin prototip kullanılmadan yapılan geçişe oranla sözdizimini kolaylaştırdığını ortaya koymuştur.

Corno ve diğerleri (2019) programlamaya yeni başlayanların kodlamada karşılaştıkları If-Then yapısına ait komut hatalarını ayıklamada kullanabilecekleri yapboz metaforunu temele alan görsel bir araç geliştirmişlerdir. Tasarım temelli araştırma (design-based research) türlerinden geliştirme araştırması olarak kurgulanan araştırmada ilk olarak son kullanıcıların bu hataları anlaması, tanımlaması ve düzeltilmesi için hangi bilgilere ihtiyaç duyduklarını belirlemek adına alanyazın taraması gerçekleştirilmiştir. Ardından, yapboz metaforuna dayalı bir araç olan My

IoT Puzzle geliştirilmiş ve kullanılabilirlik testi için altı katılımcıyla denenmiştir. Bulgular, yapboz metaforunun gerçek zamanlı geri bildirim ve açıklamalarla birlikte kullanılmasının IF-Then kuralları arasındaki çelişkilerin anlaşılmasına ve düzeltilmesine yardımcı olduğunu göstermiştir.

Schez-Sobrino ve diğerleri (2019) programlamaya yeni başlayan öğrencilere programlama öğrenmelerinde yardımcı olmak adına yollar ve trafik işaretleri metaforuna dayanan ANGELA isimli bir öğretim aracı geliştirmişlerdir. İki boyutlu bu grafik gösterim, bilgisayar bilimleri lisans programının ilk yılındaki 89 öğrenciye sunulmuş, öğrencilerden uygulamada yer alan 50 dakikalık iki görevi yapmaları istenmiştir. Uygulama öncesinde ve sonrasında programlama bilgilerini ölçen sorulardan elde edilen verilerin analizi, ANGELA isimli grafik gösteriminin programlama sürecinin anlaşılması ve öğretiminde uygunluk açısından anlamlı ve olumlu puanlar aldığını göstermiştir.

Hidalgo-Cespedes ve diğerleri (2018) bilgisayar biliminde problem çözme görevinde geleneksel metaforlar ve alegorilerle öğretimin etkisini deneysel olarak karşılaştırmıştır. Deneysel çalışmaya Kosta Rika Üniversitesinden Programlama I (CS1) derslerini alan 34 öğrenci katılmış, deney grubu öğrencilerine geleneksel metafor grubu, kontrol grubu öğrencilerine ise alegori grubu adı verilmiştir. Gruplarda gerçekleşen denel işlem sonunda öğrencilere “Goldbach Calculator Problem” isimli bir programlama görevi verilmiş ve öğrencilerin programlama performansları değerlendirilmiştir. Ayrıca katılımcılara öğretim yönteminin motive edici yönleri hakkında bir anket uygulanmıştır. Verilerin analizi sonucu elde edilen bulgular, programlama performansı açısından deney ve kontrol grubunda anlamlı bir fark olmadığını göstermiştir.

Burke (2016) bilgisayar programlamanın okullara ve ailelere yeni bir öğrenme biçimi olarak tanıtılmasının çeşitli yollarını gözden geçirmeyi ve tartışmayı amaçlamıştır. Alanyazın incelemesi türündeki çalışmasında son 30 yılda yayımlanan 67 kitap, dergi ya da editör yazısını incelemiş, belgelerde en çok yeni okuryazarlık (new literacy), temel matematik (grounded math) ve teknik beceriler (technical skills) gibi üç tanımlama öne çıkmıştır.

Sanford ve diğerleri (2014) bilgisayar bilimi eğitiminde metaforların oynadığı rolün daha iyi anlaşılmasını sağlamayı, kullanılan metafor türlerini kataloglamayı ve bunların öğrenmeyi desteklemedeki etkililiğini değerlendirmeyi amaçlamışlardır. Durum çalışması deseninde yürütülen araştırmaya ABD’de bilgisayar bilimi dersi veren 10 öğretici katılmış ve öğrencilerden yarı yapılandırılmış görüşmeler aracılığıyla veri toplanmıştır. Bulgular, öğrencilerin çok çeşitli metaforlar kullandıklarını ancak öğrencilerin bu metaforları kullanırken karşılaştıkları ortak sorunları ve yanlış anlamaları açıklamada yetersiz kaldıklarını göstermiştir.

Esper ve diğerleri (2013) yeni başlayanların programlamaya yönelik kavramları öğrenmede ve program yazmada yaşadıkları zorluklara çözüm olması adına “sihirbazlık” metaforunu temele alan bir Java eklentisi geliştirmiş, öğrenciler, bu eklenti üzerinde sihir adı verilen kodlar yazabilmiş ve sonuçlarını sihirli bir değnek aracılığıyla görebilmişlerdir. Eklenti, yaşları 8 ile 22 arasında değişen altısı kadın 11’i erkek toplam 17 kişide denemiş, örnekleme yer alan kişilere uygulama öncesi bir anket uygulanarak uygulama sonrasında ortalama 15 dakika süren yarı yapılandırılmış görüşmeler gerçekleştirilmiş, Araştırma sonucunda katılımcıların sihirbazlık metaforunu temele alan eklentiyi programı yazmada ve düzenlemede başarılı buldukları ortaya çıkmıştır.

Umar ve Hui (2012), öğrencilerin öğrenme stili tercihlerinin C++ dilini öğrenmeye etkisini özellikle yazılan kodların kalitesi ve programlamada yapılan hata sayısı açısından incelemeyi amaçlamışlardır. Araştırmada üç öğretim yöntemi (metaforla öğretim, eşli programlama ve metaforla eşli programlama) ve iki bağımlı değişken (kod kalitesi ve hata sayısı) dikkate alınarak 3x2 faktöriyel desen kullanılmış, yedi hafta süren deneysel çalışmaya 120 üniversite birinci sınıf öğrencisi katılmıştır. Öğrenciler araştırmada üç ayrı gruba ayrılmış, birinci gruba metaforla eşli programlama öğretimi, ikinci gruba sadece eşli programlama ile öğretim üçüncü gruba da sadece metaforla öğretim gerçekleştirilmiştir. Öğretim sonrasında öğrencilerin geliştirdiği ürünler programlama kalitesi ve yapılan hata sayısı açısından incelenmiştir. Sonuçlar, metaforla eşli programlama grubundaki görsel öğrencilerin kod kalitesinde önemli ölçüde daha iyi performans gösterdiğini ve kodlama sırasında eşli programlama ve metaforla öğretim gruplarındakilere göre daha az hata yaptığını

göstermiştir. Diğer yandan metaforlar, öğrencilerin bilinenleri yeni edinilen kavramlarla ilişkilendirerek daha derin bir kavramsal anlayış geliştirmelerine yardımcı olmuştur.

Hui ve Umar (2011), bilgisayar bilimi derslerinde farklı öz düzenleme düzeylerine sahip öğrencilerin programlama performansları üzerindeki metafor ve eşli programlama kullanımının etkisini belirlemeye çalışmışlardır. Araştırma, 2x2 faktöriyel desende kurgulanmış, yedi hafta sürecek denel işleme 84 üniversite birinci sınıf öğrencisi katılmıştır. Araştırmada yer alan öğrencilerin yarısı deney grubuna ($n=42$) yarısı da kontrol grubuna ($n=42$) seçkisiz olarak atanmış, öğrenciler eşli programlama ve metaforun birlikte kullanıldığı gruba ya da sadece eşli programlamanın uygulandığı gruba dahil edilmiştir. Öğrencilere, denel uygulama öncesi ve sonrası Bilgisayar Programlama Performans Testi (Computer Programming Performance Test, [CPPT]) uygulanmış, bulgular metafor ve eşli programlamanın birlikte kullanıldığı gruplarda programlama performansının sadece eşli programlamada yer alan öğrencilere oranla daha yüksek olduğunu göstermiştir.

Özetle, programlama becerisini geliştirmede metafor kullanımına yönelik araştırmalar çoğunlukla deneysel desenle yürütülmesine rağmen durum çalışması ve doküman analiziyle gerçekleştirilen çalışmalar da vardır. Deneysel araştırmalar, daha çok bir metaforu temele alarak geliştirilmiş araçların geliştirilme süreci ve kullanıcı testleri sonucu programlama becerisine olan katkısına odaklanmıştır. Bir araştırma, metafora dayalı geliştirilen bir öğretim aracının programlama becerisini geliştirmede anlamlı bir fark yaratmadığını raporlanmıştır.

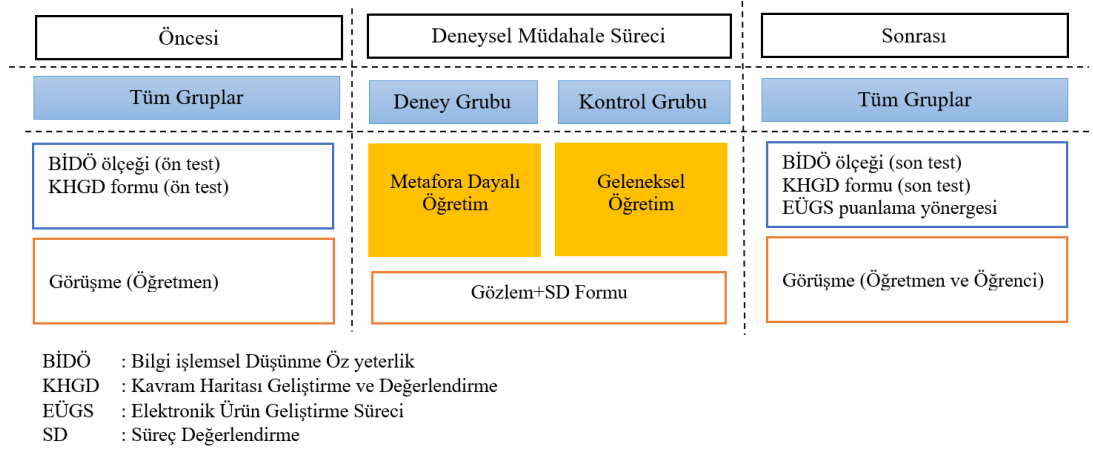
III. BÖLÜM

YÖNTEM

Bu bölümde araştırmanın amacına ulaşmak için izlenen yöntem; araştırmanın deseni, çalışma grubu, veri toplama araçları, veri toplama süreci, verilerin çözümlenmesi ve yorumlanması, geçerlik ve güvenirlik, araştırmacının rolü ve etik başlıkları altında açıklanmıştır.

3.1. Araştırmanın Deseni

Programlama öğretiminde metafor kullanımının programlama kavramlarını öğrenmeye, programlama ve bilgi işlemsel düşünme öz yeterliğini geliştirmeye etkisini inceleyen bu araştırmada karma yöntem deneysel desen kullanılmıştır. Bu desende nicel ve nitel veriler, deneysel nicel bir araştırma deseni içerisinde toplanarak analiz edilmekte, nitel verilere katılımcıların ortamından ya da kültüründen kişisel, bağlamsal ve niteliksel deneyimlere ulaşmak amacıyla yer verilmektedir (Creswell ve Plano Clark, 2018). Bu araştırmada, denel işlem materyali olarak tasarlanan metafora dayalı öğretimin etkililiğine yönelik kanıtlar elde etmenin yanında katılımcıların denel işlemi nasıl deneyimledikleri de belirlenmek istendiğinden karma yöntem deneysel desen tercih edilmiştir. Araştırmada kullanılan karma yöntem deneysel desene Şekil 1’de yer verilmiştir.



Şekil 1. Araştırmada kullanılan karma yöntem deneysel deseni

Creswell ve Plano Clark’a (2018) göre karma yöntem deneysel desende nitel veriler denel işlem öncesi, sırası ve sonrasında birçok nedenle toplanabilmektedir. Araştırmada üç aşamada (öncesi, sırası ve sonrası) nitel verilerden yararlanılmış, denel işlem öncesinde denel işlemin gerçekleşeceği sosyal ve fiziksel ortamı tanımlamak,

denel işlem sırasında öğrencilerin denel işleme ait deneyimlerini ortaya çıkarmak, denel işlemin sonunda ise gerçekleştirilen müdahale sonrası öğrencilerin kavram öğrenme durumlarını ve metafora dayalı öğretimle ilgili görüşlerini belirlemek için nitel veri toplanmıştır.

3.2. Çalışma Grubu

Araştırmanın çalışma grubunu lise 9. sınıf bilgisayar bilimi (Kur-1) öğrencileri ve öğretmenleri oluşturmuştur. Araştırmanın odağını metin tabanlı programlama oluşturmaktadır. İlk ve ortaokullarda Bilişim Teknolojileri ve Yazılım dersi öğretim programlarının metin tabanlı programlama yerine blok tabanlı programlamaya yönelik kazanımlar içermesi nedeniyle lise düzeyi tercih edilmiştir.

Araştırmanın çalışma grubunda yer alan deney ve kontrol gruplarını 2021-2022 eğitim-öğretim yılında bilgisayar bilimi (Kur-1) dersini 9. sınıf düzeyinde seçen resmi iki anadolu lisesinin altı şubesi oluşturmuş, denel işlem sürecinde grupların birbirlerinden etkilenme ihtimalleri ve aynı öğretmenin farklı gruplarda öğretimi farklılaştırmada yaşayacağı güçlükler ya da etkilenme düşünülerek deney ve kontrol grubu farklı okullardan seçilmiştir. Deney ve kontrol grupları seçkisiz grup ataması yoluyla oluşturulmuştur. Bunun için iki anadolu lisesinin birinde yer alan üç şube deney, diğer lisenin üç şubesi ise kontrol grubu olarak atanmıştır. Her iki anadolu lisesinde de birer öğretmen bilgisayar bilimi (Kur-1) dersini yürütmektedir. Tablo 1 incelendiğinde deney ($n=66$) ve kontrol grubunda ($n=69$) yer alan öğrenci sayılarının dengeli dağıldığı görülebilir.

Tablo 1. Deney ve Kontrol Gruplarında Yer alan Öğrenci Sayıları ve Şubelere Göre Dağılımı

Grup	Şube	<i>f</i>	%	<i>f</i>	%
Deney	Şube 1	19	14.1	66	48.89
	Şube 2	27	20.0		
	Şube 3	20	14.8		
Kontrol	Şube 1	25	18.5	69	51.11
	Şube 2	20	14.8		
	Şube 3	24	17.8		
Toplam	6	135	100	135	100

Araştırmanın nitel verilerini toplamak için görüşme yapılacak öğrencilerin seçiminde çalışma grubundaki öğrencilerin Kavram Haritası Geliştirme ve Değerlendirme (KHGD) formu, Elektronik Ürün Geliştirme Süreci (EÜGS) puanlama yönergesi ve Bilgi İşlemsel Düşünme Öz yeterlik Ölçeğinden (BİDÖ) aldıkları son test puanlarından yararlanılmıştır. Deney ve kontrol grubu öğrencilerinin son test puanları düşük, orta ve yüksek olarak kategorilendirilmiş, her kategoriden üçer öğrenci seçilmiştir. Böylelikle, denel işlem sonrası dokuzu deney, dokuzu kontrol grubundan olmak üzere 18 öğrenciyle görüşme gerçekleştirilmiştir.

Araştırma deseninin gereği olarak hem denel işlem öncesi süregiden öğretim süreçlerini anlamak ve bağımsız değişkene yönelik öğretim girişimini planlamak hem de denel işlem sonunda bağımsız değişkenin yarattığı öğretim sürecini daha detaylı betimlemek ve etkileri konusunda nitel veri toplamak için deney ve kontrol grubu öğretmenleri de çalışma grubuna dâhil edilmiştir.

Araştırmaya katılan öğretmenlerin ise ikisi de eğitim fakültesi, bilgisayar ve öğretim teknolojileri eğitimi bölümü mezunudur. Öğretmenler, bilgisayar bilimi dersi ve programlama öğretimi açısından benzer deneyime sahip olup, deney grubu öğretmeni 15 yıllık öğretmenlik kariyerinin sekiz yılını, kontrol grubu öğretmeni ise 16 yıllık öğretmenlik kariyerinin dört yılını şu an bulundukları okullarda geçirmişlerdir. Öğretmenler, öğretim yöntem ve tekniklerine yönelik benzer sayıda (deney grubu öğretmeni yedi, kontrol grubu öğretmeni dokuz) mesleki gelişim çalışmalarına katılmıştır. Her iki öğretmen de bilgisayar bilimi (Kur-1) dersini öğretim programının yayınlandığı 2018 yılından beri uygulamaktadır.

3.2.1. Grupların Denkliğine İlişkin Bilgiler

Araştırmada, deney ve kontrol grubunda yer alan öğrencilerin denk olup olmadıklarını belirlemek amacıyla kavram öğrenme, bilgi işlemsel düşünme öz yeterliği ve programlama becerileri açılarından incelenmiştir. Öğrencilerin denel işlem öncesi programlamaya yönelik kavram öğrenme düzeyleri KHGD formu, bilgi işlemsel düşünme öz yeterlik düzeyleri ise BİDÖ ölçeği aracılığıyla ölçülmüş, deney ve kontrol grubunun denkliğinin belirlemesinde KHGD formu ve BİDÖ ölçeğinden aldıkları ön test puanları karşılaştırılmıştır. Bunun yanında deney ve kontrol grubunda yer alan öğrenciler metin tabanlı programlama ile ilk kez karşılaştıklarından

programlama beceri düzeylerinin denel işlem öncesi ölçülmesinin anlamlı olmayacağına karar verilmiştir.

Denel işlem öncesi deney ve kontrol grubunda yer alan öğrencilerin kavram öğrenme düzeylerini belirlemeyi amaçlayan KHGD formu öğrencilere dağıtılmış, öğrencilerden elde edilen kavram haritaları KHGD formunda yer alan puanlama yönergesi aracılığıyla üç uzman tarafından puanlanmıştır. Grupların ortalama puanları arasındaki farkın anlamlılığını test etmek amacıyla bağımsız gruplar için *t* testi analizi uygulanmış ve KHGD formu ön test puanlarına göre deney ve kontrol grubu giriş özellikleri Tablo 2’de sunulmuştur.

Tablo 2. *KHGD Formu Ön test Puanlarına Göre Deney ve Kontrol Grubu Giriş Özellikleri*

Gruplar	<i>n</i>	$\bar{X}$	<i>S</i>	<i>Sd</i>	<i>t</i>	<i>p</i>
Deney	66	2.32	1.03	133	.04	.97
Kontrol	69	2.33	1.24			

Tablo 2’de görüldüğü üzere öğrencilerin denel işlem öncesi programlamaya yönelik kavramları öğrenme düzeyi ortalama puanları deney grubu için 2.32, kontrol grubu için 2.33 bulunmuştur. Analiz, deney grubu ve kontrol grubu öğrencileri arasında denel işlem öncesinde programlamaya yönelik kavramları öğrenme düzeyleri açısından anlamlı bir fark olmadığını göstermiştir [$t_{(133)} = .04$; ($p = .97 > .05$)].

Deney ve kontrol grubu öğrencilerinin bilgi işlemsel düşünme öz yeterliklerini belirlemek amacıyla her iki gruba BİDÖ ölçeği ön test olarak uygulanmış, grupların ortalama puanları arasındaki farkın anlamlılığını test etmek için bağımsız gruplar için *t* testi analizi gerçekleştirilmiştir. BİDÖ ön test puanlarına göre deney ve kontrol grubu giriş özellikleri Tablo 3’te sunulmuştur.

Tablo 3. *BİDÖ Ölçeği Ön Test Puanları Deney ve Kontrol Grubu Giriş Özellikleri*

Gruplar	<i>n</i>	$\bar{X}$	<i>S</i>	<i>Sd</i>	<i>t</i>	<i>p</i>
Deney	66	165.62	12.26	133	.93	.35
Kontrol	69	163.38	15.44			

Tablo 3’te görüldüğü üzere öğrencilerin denel işlem öncesi bilgi işlemsel düşünme öz yeterlik ortalama puanları deney grubu için 165.62, kontrol grubu için 163.38 bulunmuştur. Analiz, deney grubu ve kontrol grubu öğrencileri arasında denel

işlem öncesinde bilgi işlemsel düşünme öz yeterlikleri açısından anlamlı bir fark olmadığını göstermiştir [$t_{(133)} = .93$; ($p = .35 > .05$)].

3.3. Denel İşlem

3.3.1. Denel İşlem Materyali

Bu araştırmanın denel işlem materyalini Gagnon ve Collay'ın (2001) altı basamaklı (durum, gruplama, köprü, sorular, sergileme ve yansıtma) yapılandırıcı öğrenme tasarım modeli kullanılarak geliştirilen ve metafora dayalı öğretimi temel alan ders planları oluşturmuştur.

Denel işlem materyalini oluşturan ders planları iki bölümden oluşmuş, birinci bölümde ders planının adına, konusuna, dersin kazanımlarına, kavramlar listesine ve süresine ait genel bilgilere, ikinci bölümde ise Gagnon ve Collay (2001) tarafından geliştirilen modelin basamakları temele alınarak yapılacaklara yer verilmiştir. Genel bilgilerin yer aldığı ilk bölümde planlara isim verirken planda yer alan senaryoyla ilişkili isimler vermeye özen gösterilmiş, konu isimleri ve kazanımlarda bir değişikliğe gidilmeden dersin öğretim programında belirtildiği şekliyle yer verilmiştir. Kavramlar başlığı altında, uzman görüşleri sonucu ortaya çıkan ve konu ile ilgili olduğu düşünülen kavramlar listelenmiş, ayrıca her bir etkinliğe ayrılacak süre kazanım sayısı, öğrenci seviyesi ve uzman görüşleri dikkate alınarak araştırmacı tarafından belirlenmiştir. Ders planının ikinci bölümünü oluşturan model durum (situation), gruplama (grouping), köprü (bridge), sorular (questions), sergileme (exhibit) ve yansıtma (reflections) olmak üzere altı basamaktan oluşmuş, aşağıda bu modeli oluşturan basamaklara ilişkin Gagnon ve Collay'ın (2001) açıklamaları ve ders planlarında nasıl yer aldıkları belirtilmiştir.

Modelin *durum* basamağında öğrencilerin ilgilerinin çekilmesi, hedeften haberdar edilmeleri ve öğretim sürecinin çerçevesinin belirlenmesi beklenmektedir (Gagnon ve Collay, 2001). Bu nedenle araştırma kapsamında geliştirilen ders planlarında programlama kavramlarıyla ilişkilendirilmesi planlanan kaynak alana ait (ör. pastacılık, kalite kontrol) resim, video, karikatür vb. ürünlerin öğrencilerle paylaşılmasıyla başlanmış, daha sonra öğrencilere kaynak alan ile programlama arasında ilişki kurabilecekleri ifadeleri içeren senaryolara, öğrencilerin senaryo ile

programlama alanı arasındaki ilişkileri ifade etmelerini sağlayacak sorulara yer verilmiştir.

Gruplama basamağında, model, öğrencilerden sosyal bir ortamda etkinliklerini gerçekleştirmelerini beklemektedir (Gagnon ve Collay, 2001). Bu nedenle araştırmada denel işlemin gerçekleşeceği ortamın bilişim teknolojileri sınıfı olması ve bilgisayar bilimini öğrenmede grup halinde çalışmanın getireceği faydalar düşünülerek öğrencilerin eşli programlama aracılığıyla çalışmalarına izin verilmiş, eşli programlama sırasında programlama becerileri birbirine yakın ve uyum içinde çalışabilecek öğrencilerin birlikte çalışmaları önerilmiştir. Öğrencilerden her ders planının gruplama basamağı için ayrı gruplar oluşturmaları beklenmemiş, eğer isterlerse farklı ders planlarında farklı grup arkadaşlarını seçebilecekleri ifade edilmiştir.

Model, ders planının *köprü* basamağında öğrencilerin ön bilgileri ile yeni öğrenilecek kavramları ilişkilendirmelerini ve yeni bilgiyi doğru biçimde yapılandırmalarını beklemektedir (Gagnon ve Collay, 2001). Araştırmada köprü basamağına genellikle senaryoda yer alan kaynak alana ait öğrenci ya da öğretmen deneyimlerini paylaşımlarını sağlayacak sorularla başlanmış, daha sonra öğrencilere senaryo ile ilişkili bir programlama görevi verilerek kodlamaları istenmiştir. Bu sayede öğrencilerden programlama kavramlarını uygulama esnasında görmeleri ve kaynak-hedef-alan ilişkisini bu süreçte yapılandırmaları istenmiştir.

Sorular basamağında öğrencilerin öğrenme durumlarına dair düşünmelerinin sağlanması ve daha sonra düşüncelerini grup arkadaşlarıyla ve diğer gruplarla paylaşımları beklenmektedir (Gagnon ve Collay, 2001). Ders planlarında bu basamakta öğrencilerden genellikle belirtilen kodları yeni karşılaştıkları programlama kavramları açısından gözden geçirmelerine, kaynak alan açısından yeni öğrenilen kavramlarla ilişkiler kurmalarına ve verilen programlama görevinin gerçekleştirilmesine rehberlik edecek sorular sorulmasına dikkat edilmiştir.

Sergileme basamağında öğrenci ürünlerinin grup içinde ve diğer gruplarla paylaşılması beklenmektedir (Gagnon ve Collay, 2001). Bu nedenle sergileme basamağına denel işlem materyalini oluşturan ders planlarının tamamının programlama görevi içermesi nedeniyle öğrencilerden geliştirdikleri elektronik

ürünlerini grup içinde tartışmaları ve daha sonra da diğer gruplarla paylaşımları istenmiştir. Böylelikle her grup diğer grupların ürünlerini görme, test etme ve değerlendirme olanağı bulmuştur.

Yansıtma basamağında öğrencilerden öğrenme süreci hakkında düşünceleri, düşüncelerini paylaşımları ve öğrenme sürecini iyileştirmeye yönelik önerilerini belirtmeleri beklenmektedir (Gagnon ve Collay, 2001). Ders planlarında bu bölüm genellikle sürece ait deneyim paylaşımları ile başlanmış, daha sonra programlama kavramları ile senaryoda geçen kaynak alan ilişkilendirmelerine yönelik öğrenci görüşleri alınmıştır. Bu bölümde aynı zamanda sergilenen ürünler öğrenciler tarafından değerlendirilerek görüşleri alınmış, öğrenci ürünlerini değerlendirmede ders planının gereksinimleri dikkate alınarak ürün değerlendirme, grup içi değerlendirme, akran değerlendirme formları ve e-gelişim dosyası gibi tamamlayıcı değerlendirme araçları kullanılmıştır.

Araştırmada programlamanın temelleri için iki, program kontrolü için üç, veri yapıları ve dosya işlemleri için birer olmak üzere toplam yedi ders planı hazırlanmış, ders planlarında pastacılık, asansör, orman yolu, kış uykusu, ankesörlü telefon, depo ve nüfus kayıt sistemi gibi çeşitli kaynak alanlarından yararlanılmıştır. Ders planlarının hazırlandığı ünite ve konu bilgilerine Tablo 4'te yer verilmiştir. Ayrıca denel işlem materyaline yönelik bütüncül bir bakış getirebileceği düşünülerek Pastadan Programa başlıklı ders planı Ek-1'de sunulmuştur.

Tablo 4. *Denel İşlem Materyalini Oluşturan Ders Planlarına Ait Bilgiler*

Ünite	Konu	Ders Planı	Kazanım Sayısı	Süre (Ders Saati)
Programlama	Programlamanın Temelleri	Pastadan Programa	4	5
		Program Bir Makinedir	3	3
	Program Kontrolü	Karar Kontrol Yapıları Yol Ayrımıdır	1	4
		Tekrarlı Yapılar ve Kış Uykusu	1	4
		Fonksiyonlar ve Ankesörlü Telefonlar	3	2
	Veri Yapıları	Veri Yapıları ve Devasa Depolar	7	6
	Dosya İşlemleri	Geçmişten Bugüne Nüfus Kayıt Sistemi	2	4

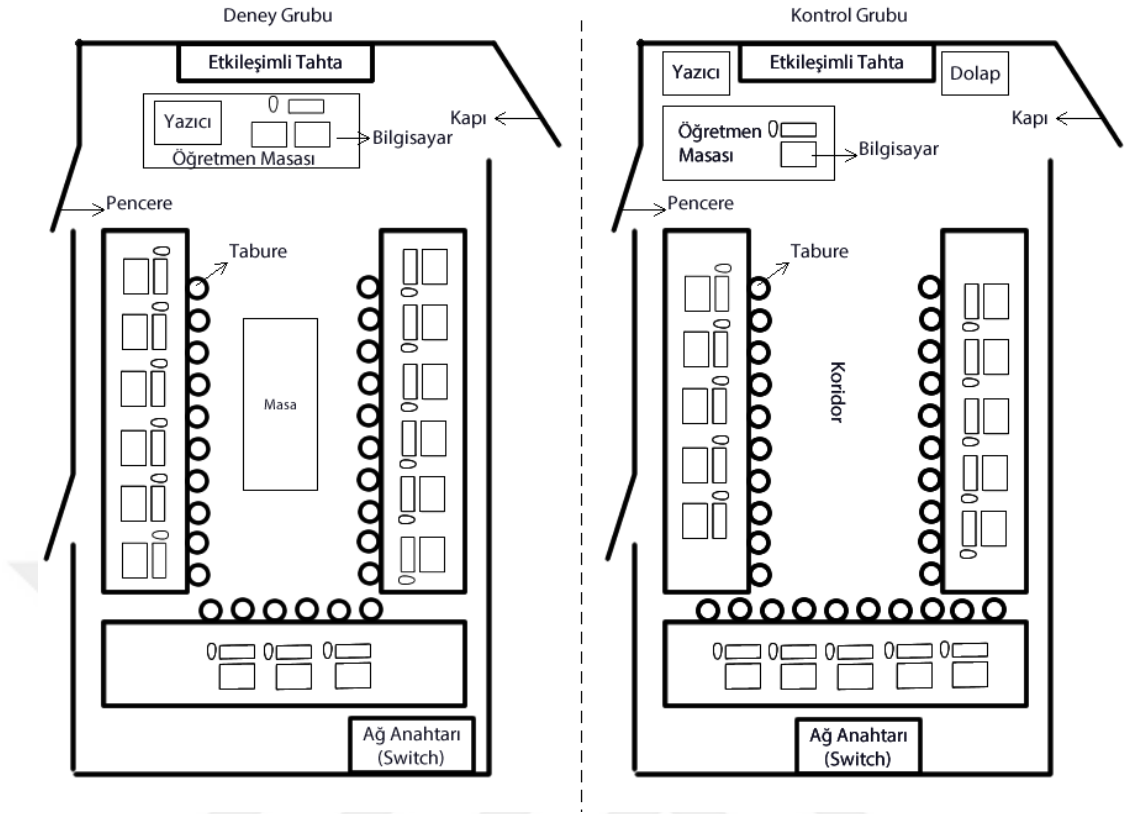
Tablo 4'te görüldüğü üzere en çok ders planı program kontrolü konusu için hazırlanmıştır. Bunun nedeni Programlama ünitesi içerisinde en fazla sürenin bu konuya ayrılması ve konunun anlaşılması zor ve soyut kavramlara diğer konulara oranla daha fazla sahip olmasıdır. Bunun yanında veri yapıları konusu yedi kazanıma sahip olmasına rağmen öğretim planında her bir veri türü için ayrı kazanım tanımlandığından bir ders planı yeterli bulunmuştur.

3.3.2. Denel İşlem Ortamı

Deney grubu toplam 685 öğrenci, 24 şube ve çeşitli branşlarda 44 öğretmenin bulunduğu bir okuldan seçilmiştir. Kontrol grubunun yer aldığı okulda ise toplam 560 öğrenci, 19 şube ve çeşitli branşlarda 40 öğretmen bulunmaktadır. Araştırmanın denel işlem ortamını bu okulların bilişim teknolojileri sınıfları oluşturmıştır.

Deney ve kontrol grubunu oluşturan iki anadolu lisesi yapılarında birer bilişim teknolojileri sınıfı bulundurmakta, bu sınıflardan deney grubunun bulunduğu sınıf birinci, kontrol grubunun bulunduğu sınıf ise okulun zemin katında yer almaktadır. Her iki bilişim teknolojileri sınıfı da Millî Eğitim Bakanlığı destekli projeler kapsamında oluşturulduğundan fiziksel olarak birçok ortak özelliği barındırmaktadır. Deney ve kontrol gruplarının öğrenim gördüğü farklı okullardaki bu sınıflarda öğrencilerin kodlarını metin biçiminde yazabilmelerini sağlayan editörlerin bulunduğu 15'i öğrenci ve biri öğretmen bilgisayarı olmak üzere toplam 16 bilgisayar yer almakta, ayrıca her iki sınıfta da MEB Fatih Projesi Faz-1 kapsamında kurulmuş birer etkileşimli tahta, birer yazıcı ve proje kapsamında kurulumu tamamlanmış ağ alt yapısı bulunmaktadır. Genel anlamda sınıflar birçok açıdan benzerlik gösterse de deney grubunda sınıfın ortasında çok amaçlı bir masa ve öğretmen masasında iki ekranın yer alması, kontrol grubunda ise etkileşimli tahtanın yanında çok amaçlı bir dolabın bulunması açısından farklılaşmaktadır. Deney ve kontrol grubu sınıflarının yerleşim düzenine Şekil 2'de yer verilmiştir.

Deney grubunun denel işlem ortamında gerçekleşen öğrenme sürecine yönelik açıklamalara denel işlem materyali bölümünde ayrıntılı biçimde yer verilmiştir. Bu nedenle kontrol grubunda nasıl bir süreç geçirildiğine yönelik açıklamalara araştırmanın geçerlik ve güvenilirliğine katkı sağlaması amacıyla "Kontrol Grubu Öğrenme Süreci" başlığı altında aşağıda ayrıca yer verilmiştir.



Şekil 2. Deney ve kontrol grubu bilişim teknolojileri sınıflarının yerleşim düzeni

3.3.3. Kontrol Grubu Öğrenme-Öğretme Süreci

Denel işlem sürecinde kontrol grubunda nasıl bir öğrenme sürecinin gerçekleştiğinin bilinmesi deney grubunda denenen metafora dayalı öğretimin ne tür bir öğrenme-öğretme süreciyle karşılaştırıldığını belirlemede ve ulaşılan bulguları anlamlandırmada önemli olabilir. Bu gerekçe ile kontrol grubu öğrenme-öğretme sürecini tanımlamada denel işlem öncesi yapılan öğretmen görüşmelerinden, denel işlem sırasında yapılan gözlem verilerinden, kimi zaman da deney grubu öğrenme süreciyle karşılaştırmalardan yararlanılmıştır. Kontrol grubu öğrenme-öğretme sürecini daha anlaşılır kılmak için açıklamalara beklentiler, kavramların öğretimi, dersin nasıl işlendiği, öğretim materyalleri ve ölçme-değerlendirme açılarından aşağıda yer verilmiştir.

Kontrol grubu öğretmeni öğrencilerden beklentilerini programda yer alan bilişsel kazanımlara ulaşmaları, problem çözebilmeleri, analitik düşünebilmeleri ve bilgi işlemsel düşünme becerileri hakkında bilgi sahibi olmaları şeklinde açıklamıştır. Derste, öğretim programında da belirtildiği üzere algoritma, akış şeması, girdi, çıktı,

karar verme, tekrarlı yapılar, veri yapıları ve fonksiyonlar gibi temel programlama kavramlarını ele aldıklarını ifade etmiştir. Programlamaya yönelik kavramların öğretiminde örneklendirmeyi ve mantığını açıklamayı kullandığını; kavramların öğrenimine destek olması için de basit bir programlama örneği vererek örneği açıkladığını; daha sonra da öğrencilere benzer görevler vererek yapmalarını beklediğini belirtmiştir. Öğretmen bir dersi nasıl işlediğini tanımlarken derse kuramsal açıklamalarla başladığını, öğrencilere bu açıklamalara yönelik sorular sorduğunu, sonrasında küçük çaplı programlama görevleri verdiğini ve görevleri kontrol ettiğini ifade etmiştir. Derste öğretim materyali olarak ise çeşitli internet sitelerinden ve forumlardan edindiği örnekleri kullandığını, dersi yıllardır verdiği için her ders öncesi elinde o derse ait materyallerinin hazır olduğunu ve bu materyallerle derse girdiğini ifade etmiştir. Ölçme-değerlendirme açısından ise her bir öğrenciye iki sınav ve iki performans notu verdiğini, öğrencilerin performans notlarını derste yaptıkları uygulamalara göre aldıklarını belirtmiştir. Sınavlar öncesi öğretmen öğrencilere örnek Python kodlarının yer aldığı çalışma kâğıtlarını dağıttığını ifade etmiştir. Bunun yanında öğrencilere ders dışı etkinlik vermediğini, başlarda öğrencilerden böyle bir talebi olduğunu ancak zamanla derslerin yoğunlaşmaya başlamasıyla öğrencilerin yarım bıraktıklarını gözlemlemiştir.

Genel olarak denel işlem öncesi öğretmen görüşmesi aracılığıyla elde edilen veriler kontrol grubu öğrenme-öğretme süreci hakkında öğrencilerden beklentilerin öğretim programındaki bilişsel kazanımlarla sınırlandırıldığı, öğretim programında yer alan temel programlama kavramlarının öğretiminde örneklendirme ve mantığını açıklamanın tercih edildiği, derse kuramsal açıklamalar ile başlanıp daha sonra programlama görevlerine geçildiği, öğretim sırasında materyal olarak çoğunlukla kod örneklerinin paylaşıldığı ve ölçme-değerlendirmede öğrenci ürünlerinin değerlendirildiği ortaya çıkmıştır.

3.4. Veri Toplama Araçları

Araştırmada, nicel verilerin toplamada Kavram Haritası Geliştirme ve Değerlendirme (KHGD) formu, Bilgi işlemsel Düşünme Öz Yeterlik (BİDÖ) ölçeği ve Elektronik Ürün Geliştirme Süreci (EÜGS) puanlama yönergesi kullanılmış; nitel verilerin toplanmasında gözlem formu, Süreç Değerlendirme (SD) formu, denel işlem

öncesi ve sonrası öğretmen görüşme formları ve denel işlem sonrası öğrenci görüşme formu kullanılmıştır.

3.4.1. Nicel Veri Toplama Araçları

Araştırmada nicel veri toplama araçları olarak kullanılan KHGD formu, BİDÖ ölçeği ve EÜGS puanlama yönergesinin nasıl geliştirildiğine yönelik açıklamalara aşağıda yer verilmiştir.

3.4.1.1. Kavram Haritası Geliştirme ve Değerlendirme (KHGD) Formu

Denel işlem öncesinde ve sonrasında, deney ve kontrol grubunda yer alan öğrencilere araştırmacı tarafından geliştirilen KHGD formu uygulanmıştır. Form, denel işlem öncesinde deney ve kontrol grubu öğrencilerinin programlama ünitesinde yer alan kavramlara yönelik ön bilgilerini, denel işlem sonrasında ise kavram öğrenme düzeylerini belirlemek amacıyla kullanılmıştır. Aynı zamanda öğrencilerin kavramları nasıl yapılandırdıkları ve metin tabanlı programlama öğretiminde metafor kullanımının programlama kavramlarını öğrenmeye etkisinin ortaya çıkarılması amaçlanmıştır.

KHGD iki bölümden oluşmaktadır. Birinci bölüm öğrencilerin kavram haritalarını oluşturmalarını sağlarken, ikinci bölüm kavram haritasının uzmanlar tarafından nasıl değerlendirileceğini içermektedir. Birinci bölümde araştırmaya ilişkin bilgiler ve öğrencilerin kavram haritalarını hazırlamaları için bir alan yer alırken, ikinci bölümde bir puanlama yönergesi ve puanlama yönergesine ait açıklamalar yer almaktadır.

Formun geliştirilmesine ilk olarak alanyazın taraması ve kavram haritalamada dikkat edilecek özelliklerin belirlenmesi ile başlanmış, daha sonra öğrenciler tarafından oluşturulacak kavram haritalarının değerlendirilmesinde kolaylık sağlaması amacıyla programlama ünitesinde yer alan konuların kapsamı belirlenmiştir. Taslak hali oluşturulan formda öğrencilerin kavram şemalarını özgürce yansıtabilmeleri amacıyla sadece “programlama” kavramına yer verilmiştir.

Formun ikinci bölümünü oluşturmak amacıyla geliştirilen kavram haritalarının nasıl değerlendirileceğine yönelik alanyazın taraması gerçekleştirilmiş ve uzman görüşleri alınmıştır. Alanyazın taraması ve uzman görüşü sonucu kavram haritalarının değerlendirilmesinde puanlama yönergesi (rubric) kullanımının uygun olacağına karar

verilmiş, analitik puanlamaya uygun olarak hazırlanan yönergede kavram haritalarının doğası gereği nicel ve nitel özellikleri bir arada barındırabileceği düşünülerek haritaların bu özellikler açısından birlikte değerlendirilmesine karar verilmiştir. Kavram haritalarının değerlendirilmesinde kullanılacak puanlama yönergesi kavram genişliği (nicelik), derinlik (nitelik-ilişkilendirme), derinlik (nitelik-bağlantı sözcüklerinin kullanımı) ve derinlik (nitelik-örnek kullanımı) başlıklarından oluşmuştur. Yönerge, puanlayıcıların kavram genişliği açısından kapsam dâhilinde kullanılan kavramların sayısını, diğer başlıklarda yine kapsam dâhilinde kavramlar arası ilişkiler, bağlantı sözcükleri ve sunulan örnekleri değerlendirebilecekleri şekilde hazırlanmıştır.

Hazırlanan form üçü Bilgisayar ve Öğretim Teknolojileri Eğitimi (BÖTE), diğer üçü de Eğitim Programları ve Öğretim (EPÖ) ana bilim dalının toplam altı akademisyeni tarafından incelenmiş ve forma ilişkin geribildirimler alınmıştır. Geribildirimler doğrultusunda düzenlenen form, 40 öğrenciye deneme amaçlı uygulanmış, öğrencilerden elde edilen kavram haritaları biri BÖTE bölümünden, biri Bilgisayar Mühendisliğinden, biri de araştırmacı olmak üzere toplamda üç puanlayıcı tarafından puanlama yönergesi kullanılarak değerlendirilmiştir. Puanlama yönergesi aracılığıyla öğrenci kavram haritalarını değerlendiren puanlayıcılar arası uyum düzeylerini gösteren Kappa istatistiği sonuçlarına Tablo 5'te ve geliştirilen KHGD formuna Ek-2'de yer verilmiştir.

Tablo 5. KHGD Formu Puanlayıcılar Arası Uyum Düzeyini Gösteren Kappa İstatistiği Sonuçları

Puanlama Yönergesi Kategorileri	Kappa İstatistiği Değeri (κ)
Programlamanın Temelleri	0.52*
Program Kontrolü	0.66*
Veri Yapıları	0.52*
Dosya İşlemleri	0.64*

* $p < .001$

Tablo 5 incelendiğinde puanlayıcılara ait Kappa değerlerinin (κ) 0.52 ile 0.66 arasında değiştiği görülmüş, bu değerler istatistiksel açıdan anlamlı bulunmuştur ($p < .001$). Üç puanlayıcı arasında en düşük uyuma Programlamanın Temelleri kategorisinde ($\kappa = 0.52$), en yüksek uyuma ise Program Kontrolü kategorisinde ($\kappa =$

0.66) ulaşılmış, Landis ve Koch (1977) tarafından önerilen uyum düzeyleri açısından tüm düzeylerde en az orta (moderate) derecede uyum gözlenmiştir. Bu uyum düzeyleri sonraki puanlamalarda tek bir puanlayıcının kullanılmasında yeterli bulunulmakla birlikte araştırmaya üç puanlayıcıyla devam edilmiştir.

3.4.1.2. Bilgi İşlemsel Düşünme Öz Yeterlik (BİDÖ) Ölçeği

Araştırmada, öğrencilerin bilgi işlemsel düşünme öz yeterlik düzeylerini belirlemek amacıyla Bilgi İşlemsel Düşünme Öz Yeterlik ölçeği kullanılmıştır. Yağcı (2019) tarafından geliştirilen ölçeğin geçerlik ve güvenirlik çalışmalarına basit seçkisiz yöntemle belirlenen 785 lise öğrencisi katılmış, öğrencilerin 275'i açımlayıcı faktör analizine (AFA), 270'i doğrulayıcı faktör analizine (DFA), 240'ı da tutarlılığı belirlemek adına test tekrar test yöntemine dâhil edilmiştir. Ölçek geliştirilirken ilk olarak alanyazın taranarak 53 maddeden oluşan bir madde havuzu oluşturulmuş, madde havuzu beşi BÖTE, ikisi bilgisayar mühendisliği ve üçü eğitim bilimleri alanından olmak üzere toplam 10 uzmana geribildirim alınmak üzere gönderilmiştir. Öneriler sonucunda her bir maddenin kapsam geçerlilik oranı belirlenmiş, oluşan taslak form 22 öğrenciyle denenmiştir. Ölçeğin faktörlerini belirlemek amacıyla temel bileşenler analizi gerçekleştirilmiş, faktör yükü .40'ın altında olan ve iki faktör üzerindeki yükleri arasında en az 0.100 fark olmayan maddeler (yani iki faktöre çapraz yüklenen maddeler) ölçekten çıkarılmıştır.

Yağcı (2019), açımlayıcı faktör analizinin sonuçlarına göre kapsam geçerlik oranını (KMO) .92, Bartlett Sphericity test sonucunun ($\chi^2 = 8497.54$; $sd = 861$, $p < .01$) ise anlamlı çıktığını rapor etmektedir. Ölçeğin dört faktörden oluşan yapısı (Problem Çözme, İşbirlikli Öğrenme ve Eleştirel Düşünme, Yaratıcı Düşünme, Algoritmik Düşünme) toplam varyansın %66.49'unu açıklamıştır. Toplam varyansın birinci faktör %47.69'unu, ikinci faktör %9.45'ini, üçüncü faktör %5.09'unu ve dördüncü faktör %4.25'ini açıklamıştır. Doğrulayıcı faktör analizine göre ki-kare değeri ($\chi^2 = 2679.07$; $sd = 815$, $p < 0.05$) değeri anlamlı bulunmuş, uyum indeks değerleri $RMSA = .0075$; $SRMR = .081$; $NNFI = .91$; $CFI = .92$; $GFI = .90$ ve $AGFI = .88$ olarak hesaplanmıştır. Ölçeğin geneli için Cronbach Alfa güvenirlik katsayısı .97 olarak belirlenmiş, iç tutarlılık ve yarı yarıya güvenirlik katsayılarına ilişkin bu bulgular genel ölçeğin ve faktörlerinin güvenilir sonuçlar ürettiğini göstermiştir.

Sonuç olarak ölçekten elde edilen açımlayıcı ve doğrulayıcı faktör analizi sonuçları kabul edilebilir düzeyde bulunmuştur. Bu araştırma kapsamında elde edilen deney ve kontrol grubu BİDÖ öntest verileri aracılığıyla ölçeğin geneli için Cronbach Alfa güvenirlik katsayısı ise .83 olarak hesaplanmıştır.

3.4.1.3. Elektronik Ürün Geliştirme Süreci (EÜGS) Puanlama Yönergesi

Programlama, bir problem çözme sürecidir ve öğrencilerin programla performansları dolaylı olarak bir problem üzerinden gözlemlenebilir. Bu çalışmada öğrencilerin programlama kullanarak problem çözme performanslarını ölçmek için şifrelenmiş mesajlar oluşturma senaryosu geliştirilmiş (Ek-3), senaryonun hazırlanmasında bilgisayar bilimi (Kur-1) dersi öğretim programında yer alan “Programlama” ünitesi kazanımları ve öğrencilerden beklenen programlama performans ölçütleri dikkate alınmıştır. Senaryonun geliştirilmesinde öğrencilerin meraklarını uyandırabileceği düşünülen “şifreleme” ve “ölümsüzlük sırrını bulma” gibi konular seçilmiş, öğrencilerden beklenenler açık bir şekilde tanımlanarak elektronik ürünlerin nasıl değerlendirileceği belirtilmiştir. Geliştirilen senaryo, deney ve kontrol grubu dışında iki bilgisayar bilimi (Kur-1) dersi öğretmeni, iki EPÖ Anabilim Dalı ve bir BÖTE bölümünden olmak üzere toplam beş uzman tarafından içerik ve öğrenci seviyesine uygunluk açılarından incelenmiştir.

Senaryonun geliştirilmesinden sonra öğrencilerin elektronik ürün geliştirme süreçlerinin nasıl değerlendirileceğine yönelik alanyazın taraması yapılmış ve performans tabanlı değerlendirme yaklaşımlarından puanlama yönergelerinin (rubric) kullanılmasına karar verilmiştir. Puanlama yönergeleri, özellikle performans değerlendirmede geleneksel testlerden farklı olarak kesin doğru ya da yanlışlara dayalı olmadan başarı ve başarısızlıkların derecelerini gösterebilmeleri (Kan, 2007) nedeniyle tercih edilmiştir. Puanlama yönergesi geliştirilirken bilgisayar bilimi (Kur-1) dersi öğretim programı incelenmiş ve programlama ünitesinde yer alan Programlamanın Temelleri, Program Kontrolü, Veri Yapıları, Dosya İşlemleri, Proje Geliştirme konularının yönergenin kapsamını oluşturmaya karar verilmiştir. Daha sonra puanlama yönergesinde yer alacak maddeler alanyazın taraması, programda yer alan kazanımlar ve performans ölçütleri dikkate alınarak oluşturulmuştur. Taslak puanlama yönergesi geliştirilirken elektronik ürün geliştirme süreci dikkate alınmış,

yapılacak işler aşama aşama planlanmış ve oluşacak ürünün bileşenleri ayrıntılı olarak tanımlanmıştır. Bu nedenlerle puanlama yönergesi analitik puanlamaya uygun olarak hazırlanmış, dördü EPÖ Anabilim Dalı, dördü de BÖTE bölümünden olmak üzere toplam sekiz alan uzmanının görüşüne sunulmuştur. Öneriler doğrultusunda yeniden düzenlenen puanlama yönergesi kullanışlılık açısından incelenmesi için araştırmacı dışında üç bilgisayar bilimi (Kur-1) dersi öğretmenine yönerge hakkındaki genel izlenimleri için gönderilmiş, öğretmenler forma yönelik herhangi bir değişiklik önermemiştir. Nihai form, Algoritma ve Akış Şeması Oluşturma, Veri Girişi ve Çıkışı, Karar Yapıları, Tekrarlı Yapılar, Fonksiyonlar, Veri Yapıları, Dosya İşlemleri ve Proje Geliştirme başlıklarından oluşmuş, 56 öğrenciden form aracılığıyla elektronik ürünler geliştirmeleri beklenmiştir. Geliştirilen öğrenci ürünleri, her bir öğrencinin kimlikleri şifrelenerek araştırmacı dâhil üç bilişim teknolojileri öğretmeni tarafından değerlendirilmiş, üç puanlayıcının uyum düzeylerini gösteren Kappa istatistiği sonuçlarına Tablo 6’da geliştirilen EÜGS puanlama yönergesine Ek-4’te yer verilmiştir.

Tablo 6. *EÜGS Puanlama Yönergesi Puanlayıcılar Arası Uyum Düzeyini Gösteren Kappa İstatistiği Sonuçları*

Puanlama Yönergesi Kategorileri	Kappa İstatistiği Değeri (κ)
Algoritma ve Akış Şeması Oluşturma	0.64*
Veri Girişi ve Çıkışı	0.84*
Karar Yapıları	0.64*
Tekrarlı Yapılar	0.78*
Fonksiyonlar	0.60*
Veri Yapıları	0.67*
Dosya İşlemleri	0.69*
Proje Geliştirme	0.52*

* $p < .001$

Tablo 6 incelendiğinde puanlayıcılara ait Kappa değerlerinin .52 ile .84 arasında değiştiği görülmüş ve değerler istatistiksel açıdan anlamlı bulunmuştur ($p < 0.001$). Üç puanlayıcı, en düşük uyuma Proje Geliştirme kategorisinde ($\kappa = 0.52$), en yüksek uyuma ise Veri Giriş Çıkışı kategorisinde ($\kappa = 0.84$) ulaşmıştır. Landis ve Koch (1977) tarafından önerilen uyum düzeylerine göre Proje Geliştirme kategorisi orta

(moderate), diğer kategoriler ise önemli (substantial) düzeyde uyumlu bulunmuştur. Bu uyum düzeyleri sonraki puanlamalarda tek bir puanlayıcının kullanılmasında yeterli bulunulmakla birlikte araştırmaya üç puanlayıcıyla devam edilmiştir.

3.4.2. Nitel Veri Toplama Araçları

Araştırmada nitel veri toplama araçları olarak kullanılan Görüşme Formları, Gözlem Formu ve Süreç Değerlendirme Formu'nun nasıl geliştirildiğine ilişkin bilgilere sırasıyla aşağıda yer verilmiştir.

3.4.2.1. Görüşme Formu

Araştırmanın nitel alt problemlerine yanıt aramak amacıyla deney ve kontrol gruplarında denel işlem öncesi ve sonrası öğretmen ve denel işlem sonrası öğrenci görüşmelerinde kullanılmak üzere yarı yapılandırılmış görüşme formları geliştirilmiştir. Denel işlem öncesi yapılan öğretmen görüşmesi ile programlama ünitesinin var olan öğretim süreçlerinin betimlenmesi ve ünitenin metafora dayalı planlanmasında işe koşulabilecek çalışmaların neler olabileceği; denel işlem sonrasında öğretmenlere uygulanan yarı yapılandırılmış görüşme formuyla ise araştırma kapsamında metafora dayalı öğretimin nasıl çalıştığı ve çıktıların oluşma nedenleri ortaya çıkarılmak istenmiştir. Denel işlem sonrası öğrenci görüşme formuyla da öğrencilerin bakış açısından denel işlem süreci ve çıktılarının anlaşılması amaçlanmıştır.

Araştırmada denel işlem öncesi yapılan öğretmen görüşmesi için gerekli taslak sorular ve form bilgisayar bilimi (Kur-1) dersi öğretim programını oluşturan öğeler (kazanım, içerik, eğitim durumları ve değerlendirme); denel işlem sonrasında öğretmenlere ve öğrencilere uygulanan yarı yapılandırılmış görüşme taslak soruları ve formlar ise araştırma alt problemleri ve denel işlem süreci dikkate alınarak hazırlanmış, görüşme formlarını oluştururken araştırma sorularıyla ilişkisini kurmak için matris oluşturulmuştur. Geliştirilen taslak formlara ilişkin dördü EPÖ Anabilim Dalı, dördü BÖTE bölümünden olmak üzere sekiz uzmandan geribildirim alınmış, öğretmen ve öğrenciler için deneme formları geliştirilmiş, denel işlem öncesi ve sonrası öğretmen görüşme formları deney ve kontrol grubu dışında bir bilgisayar bilimi (Kur-1) dersi öğretmeniyle, denel işlem sonrası öğrenci görüşme formu ise üç

9. sınıf öğrencisiyle denenerek öğrenci seviyesine uygunluk açısından gözden geçirilmiştir.

Sonuç olarak denel işlem öncesi öğretmen görüşme formu (Ek-5) üçü giriş, beşi içeriğe yönelik olmak üzere toplam sekiz sorudan, denel işlem sonrası öğretmen (Ek-6) ve öğrenci (Ek-7) görüşme formları ise ikisi görüşmeye hazırlık ve 10'u içeriğe yönelik olmak üzere 12 sorudan oluşmuştur. Görüşme formlarında bilgisayar bilimi (Kur-1) öğretim programı öğeleri ve dersin öğretim süreci temele alındığından deney ve kontrol gruplarında aynı görüşme formları kullanılmıştır.

3.4.2.2. Gözlem Formu

Denel işlem boyunca deney ve kontrol gruplarına ait nitel verileri sistematik ve kontrollü bir şekilde toplamak amacıyla araştırmacı tarafından gözlem formu geliştirilmiştir. Süreçte olup biten her şeyin gözlenmesi mümkün olmadığından gerçekleşecek gözlemin kapsamının belirlenmesi için araştırmanın alt problemleri dikkate alınarak alanyazın taraması gerçekleştirilmiş, formda yer alacak boyutlara ve gözleme odaklanmayı kolaylaştırması planlanan geçici kod listesine yer verilmiştir. Daha sonra taslak gözlem formu tez izleme komitesi üyelerinin görüşlerine sunulmuş, gelen geribildirimler doğrultusunda değişiklikler uygulanmıştır. Öneriler doğrultusunda oluşan form, tez izleme komitesi üyeleri dışında ikisi EPÖ, üçü de BÖTE alanında toplam beş uzmanın görüşlerine sunulmuş, geri bildirimler sonucu gözlem formuna son hali verilmiştir. Nihai form (Ek-8) gözleme ilişkin açıklamalar, gözlemin amacı, veri toplama süreci, gözlem boyutları ve bu boyutlara ilişkin geçici kod listesinden oluşmuştur.

3.4.2.3. Süreç Değerlendirme (SD) Formu

Form, deney ve kontrol grubu öğrencilerinin denel işlem boyunca programlama ünitesindeki programlamanın temelleri, program kontrolü, veri yapıları, dosya işlemleri ve proje geliştirme ile ilgili gerçekleştirilen etkinlikler hakkında görüşlerini almak amacıyla geliştirilmiştir. Değerlendirme formunda özellikle öğrencilerin süreçte yaşadıklarını yansıtmaları amaçlandığından açık uçlu sorulardan oluşmasına özen gösterilmiştir. Formda, her konu sonunda sorulabileceği düşünülen dört soru belirlenmiş, soruların öğrencilere elektronik bir form aracılığıyla ulaştırılması planlandığından süreç değerlendirme formu elektronik ortama

aktarılmıştır. Formda yer alan yönerge ve açık uçlu sorulara yönelik dördü EPÖ ve dördü de BÖTE alanında toplam sekiz uzmandan görüş alınarak forma son hali (Ek-9) verilmiştir.

3.5. Veri Toplama Süreci

Veriler, 2021-2022 eğitim-öğretim yılının ikinci döneminde toplanmıştır. Denel işlem ilk olarak 2020-2021 eğitim öğretim yılı için planlanarak uygulanmaya başlamış fakat denel işlem sırasında, 188 ülkede 1.5 milyar öğrencinin okuldan uzak kalmasına neden olan (OECD, 2021) COVID-19 pandemisinin ortaya çıkmasıyla kimi zaman yüz yüze kimi zaman çevrimiçi gerçekleşen dersler ve öğrencilerin derse devam durumlarındaki azalmalar nedeniyle uygulama, planlandığı gibi tamamlanamamıştır.

COVID-19 pandemisinin ortaya çıkmasıyla 2020-2021 eğitim-öğretim yılında tamamlanamayan uygulamaya İl Milli Eğitim Müdürlüğünden (Ek-10) ve Ege Üniversitesi Sosyal ve Beşerî Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulu'ndan (Ek-11) gerekli izinler alınarak başlanmıştır. Denel işlem öncesinde Programlama ünitesinde yer alan kavramları belirlemek amacıyla elektronik bir form geliştirilerek dört bilgisayar bilimi (Kur-1) dersi öğretmeni ve bir BÖTE bölümü uzmanına incelemeleri için gönderilmiş, form aracılığıyla programlama ünitesinde yer alan kavramların kapsamını belirleyen liste (Tablo 7) oluşturulmuştur. Öğrencilere, kavram haritası hazırlama eğitimi verilerek KHGD formu ve BİDÖ ölçeği ön test olarak uygulanmış, denel işlem öncesi deney ve kontrol grubu öğretmenleriyle görüşmeler gerçekleştirilmiş, 18 hafta sürmesi planlanan denel uygulamaya geçilmiş, fakat denel işlemin 13 haftası uzaktan ve beş haftası yüz yüze gerçekleşebilmiştir. Bu süreçte programlama ünitesinde 36 saatlik planlanan uygulamanın 18 saatine yönelik süreç değerlendirmeleri toplanabilmiş, deney ve kontrol grubundan ise herhangi bir elektronik ürün toplanamamıştır. Son olarak KHGD formu ve BİDÖ ölçeği son test olarak uygulanmış, deney ve kontrol grubu öğretmenleriyle denel işlem sonrası öğretmen görüşmeleri ve üç öğrenciyle çevrimiçi denel işlem sonrası görüşme gerçekleştirilebilmiştir.

Tablo 7. *Programlama Ünitesinde Yer Alan Kavramları Belirleme Formu Aracılığıyla Alan Uzmanları Tarafından Belirlenen Kavramların Listesi*

Konu	Kavramlar
Programlamanın Temelleri	Programlama Metin tabanlı programlama Algoritma Değişken Sabit Girdi Çıktı Program hatası Program testi Derleyici Yorumlayıcı
Program Kontrolü	Kontrol Yapıları Tekrarlı Yapılar Parametre Fonksiyon Parametre Argüman
Veri Yapıları	Sayısal veri Dizgi türü veri Dizi tipinde veri Liste tipinde veri Sözlük tipinde veri Demet tipinde veri Küme tipinde veri
Dosya İşlemleri	Dosya oluşturma Dosyadan okuma Dosyaya yazma Dosya silme Dosya düzenleme

2021-2022 eğitim-öğretim yılında gerçekleşen ve bu araştırmanın bulgularına dayanak oluşturan nihai uygulamaya ise ilk olarak İl Milli Eğitim Müdürlüğü (Ek-12) ve Ege Üniversitesi Sosyal ve Beşerî Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulu'na bildirilerek gerekli onaylar (Ek-13) alınarak başlanmıştır. Aşağıda, onayların alınmasından denel işlem sonuna kadar gerçekleştirilen veri toplama süreci özetlenmiş, bu araştırmanın bulgularına dayanak oluşturan veri toplama sürecine Şekil 3'te yer verilmiştir.

- Denel işlem öncesinde deney ve kontrol grubu öğretmenleriyle uygulama süreci hakkında 45 dakikalık bilgilendirme toplantısı yapılmış, deney grubu öğretmeni ile gerçekleşecek denel işlem materyaline ait 20 dakikalık uygulama süreci toplantısı gerçekleştirilmiştir. Denel işlem materyalini

deney grubu öğretmeni 2020-2021 eğitim-öğretim yılında da uyguladığından toplantının kısa sürdüğü düşünülmüştür.

- Deney ve kontrol grubu öğretmenleri elektronik üründe yer alan senaryoyu COVID-19 pandemisinin ortaya çıkmasıyla ertelenen denel işleminden edindikleri deneyimlere dayanarak tekrar incelemiş, senaryo değiştirilerek görev farklılaştırılmış, öğrenci düzeyine uygun hale getirilmiştir.
- Denel işlem öncesi deney ve kontrol grubunda yer alan öğrencilere bir ders (40 dakika) süresince kavram haritası hazırlama eğitimi verilmiş, sonrasında KHGD formu ve BİDÖ ölçeği ön test olarak uygulanmıştır.
- Denel işlem, 36 ders saati (18 hafta) uygulanmıştır. Deney ve kontrol grupları, araştırmacı tarafından geliştirilen gözlem formu aracılığıyla 18 hafta gözlenmiş ve öğrenciler programlama ünitesinde yer alan konuların sonunda toplam dört SD formunu doldurmuşlardır. Denel işlem boyunca dersler aynı zamanda kamerayla kayıt altına alınmış, fakat video kayıtları gerçekleştirilen gözlem verilerindeki doğrudan alıntılar yazmada, bir diğer anlatımla veri kaybını önlemede kullanılmış, ayrıca deşifre edilmemiştir.
- Denel işlem sonunda öğrencilerden denel işlemin başında verilen ve EÜGS puanlama yönergesine uygun olarak hazırlanması beklenen elektronik ürünler toplanmıştır. Ayrıca çalışma grubunda yer alan öğrencilerin KHGD formunu ve BİDÖ ölçeğini son test olarak tekrar doldurmaları istenmiş; deney ve kontrol grubunda yer alan öğrenci ve öğretmenlerle yarı yapılandırılmış görüşmeler gerçekleştirilerek denel işlem materyaline ve sürece yönelik görüşleri alınmıştır. Görüşmeler öğrenci ve öğretmenlerin bulundukları okulda, okul yönetimi tarafından görüşme yapabilmek için uygun görülen boş sınıflarda gerçekleştirilmiştir. Denel işlem öncesi deney grubu öğretmeniyle gerçekleştirilen görüşme 14 dakika, kontrol grubu öğretmeniyle gerçekleştirilen görüşme 20 dakika sürmüştür. Denel işlem sonrası ise deney grubu öğretmeniyle yapılan görüşme 26 dakika, kontrol grubu öğretmeniyle yapılan görüşme 20 dakika sürmüştür. Deney ve kontrol grubu öğrencileri ile denel işlem sonrası yapılan görüşmeler ortalama 13, toplam 234 dakikadır.

Veri Toplama Süreci



Şekil 3. Veri toplama süreci

3.6. Verilerin Çözümlemesi ve Yorumlanması

Bu başlık altında nicel ve nitel verileri çözümleme ve yorumlama süreçlerine yer verilmiştir. Nicel verilerin analizinde karışık ölçümler için ANOVA (Mixed ANOVA), ilişkisiz ölçümlerde ortalama puanların karşılaştırılması için *t* testi ve Mann-Whitney *U* testi; nitel verilerin çözümlemesinde ise Creswell ve Plano Clark'ın (2018) veri analiz spirali kullanılmıştır. Aşağıda her iki veri türü için gerçekleştirilen işlemler ayrıntıları ile sunulmuştur.

3.6.1. Nicel Verilerin Çözümlemesi ve Yorumlanması

Araştırmada deney ve kontrol grubu öğrencilerinden KHGD formu, BİDÖ ölçeği ve EÜGS puanlama yönergesi aracılığıyla toplanan nicel veriler SPSS (Statistical Package for the Social Sciences) programına aktarılmış, BİDÖ ölçeğinde yer alan olumsuz maddelerin dönüşümü gerçekleştirilmiş ve çözümleme için bir veritabanı oluşturulmuştur. Bu veritabanında yer alan nicel veriler deney ve kontrol gruplarının denklik durumlarını belirlemede, puanlayıcılar arası uyumu hesaplamada ve grupların denel işlem öncesi ve sonrası ölçme sonuçlarını karşılaştırmada kullanılmıştır. Grupların denklik durumlarını belirlemede ön test olarak uygulanan KHGD formu ve BİDÖ ölçeğinden elde edilen veriler üzerinde ilişkisiz örneklem için *t* testi; KHGD formu ve EÜGS puanlama yönergesini puanlayan puanlayıcılar arası uyumu belirlemek için ise Kappa testi kullanılmıştır. Araştırmada elde edilen puanlayıcılar arası uyum düzeyleri Landis ve Koch (1977) tarafından önerilen Kappa değeri düzeylerine göre yorumlanmıştır. Ölçme sonuçlarını karşılaştırmada; KHGD formu ve BİDÖ ölçeğine yönelik gruplardan elde edilen ön test ve son test verileri üzerinde karışık ölçümler için ANOVA, denel işlem sonunda grupların EÜGS puanlama yönergesinden aldıkları puanlar üzerinde ise ilişkisiz örneklem için *t* testi gerçekleştirilmiştir. Karışık ölçümler için gerçekleştirilen ANOVA sonucunda elde edilen kısmi eta kare (η_p^2) değerleri Cohen'in (1988) sınıflamasına uygun olarak .01'den düşük değerler için küçük (small), .01 ve .06 arası değerler için orta (medium) ve .14'ten yüksek değerler için büyük (large) etki olarak yorumlanmıştır. EÜGS puanlama yönergesinde sadece "algoritma ve akış şeması" alt performans görevine ait puanlar normallik varsayımını karşılayamadığından bu alt performans göstergesine ait veriler üzerinde Mann-Whitney *U* testi analizi gerçekleştirilmiş, analiz sonucunda elde

edilen “*r*” değerinin yorumlanmasında da Cohen’in (1988) .1’den düşük değerler için küçük (small), .1 ve .3 arası değerler için orta (medium) ve .5’ten yüksek değerler için büyük (large) etki tanımlaması dikkate alınmıştır.

Nicel verilerin analizi sonucunda elde edilen bulgular araştırmada betimsel ve karşılaştırma tabloları kullanılarak sunulmuştur. Bulgulara açıklayıcı bir giriş ile başlanmış, daha sonra bu girişe yönelik betimsel tablolar sunulmuş, son olarak karşılaştırma tabloları istatistiksel ve mantıksal açıklanmıştır.

3.6.2. Nitel Verilerin Çözümlemesi ve Yorumlanması

Araştırmada nitel verilerin analiz sürecinde doğrusal bir yaklaşım izlemek yerine Creswell ve Plano Clark’ın (2018) analitik döngülere yer veren veri analiz spiralinden yararlanılmıştır. Veri analiz spirali verileri yönetme ve düzenleme (managing and organizing data), ortaya çıkan fikirleri okuma ve not etme (reading and memoing emerging ideas), kodları tanımlama ve temalar halinde sınıflandırma (describing and classifying codes into themes), yorumları geliştirme ve değerlendirme (developing and assessing interpretations), verileri temsil etme ve görselleştirme (representing and visualising data) olmak üzere beş aşamadan oluşmaktadır. Creswell ve Plano Clark’ın (2018) veri analiz spirali dikkate alınarak gerçekleştirilen nitel veri analiz sürecinde yapılan çalışmalara aşağıda yer verilmektedir.

3.6.2.1. Verileri Yönetme ve Düzenleme

Araştırmanın nitel veri setini öğrenci ve öğretmenlerle yapılan denel işlem öncesi ve sonrası görüşmeler, SD formu ve gözlem formlarından elde edilen veriler oluşturmuştur. Öğretmenlerle denel işlem öncesinde ve sonrasında ikişer olmak üzere toplam dört, öğrencilerle denel işlem sonrasında deney ve kontrol grubunda dokuzar olmak üzere toplam 18 görüşme gerçekleştirilmiş, daha sonra 18 öğrenci ve dört öğretmen görüşmesi harfiyen deşifre edilerek toplam 22 görüşme elektronik olarak saklanmıştır. Görüşmelerde öğretmenler bulundukları gruba göre D-ÖRT (Deney Grubu Öğretmeni) ve K-ÖRT (Kontrol Grubu Öğretmeni) olarak, öğrenciler de benzer şekilde bulundukları gruba göre ardışık olarak etiketlenmişlerdir. Örneğin, 1 Numaralı Kontrol Grubu Öğrencisi, K-Ö1 ve 3 Numaralı Deney Grubu Öğrencisi, D-Ö3 olarak etiketlenmiştir.

Deney ($n=66$) ve kontrol grubunda ($n=69$) yer alan tüm öğrencilerden Programlama ünitesinde yer alan programlamanın temelleri, program kontrolü, veri yapıları ve dosya işlemleri konularının her birinin sonunda çevrimiçi olarak SD formlarını doldurmaları istenmiş, verilere elektronik ortamda ulaşıldığından deşifre işlemine ihtiyaç duyulmamıştır. Ayrıca Programlama ünitesi boyunca toplam 18 hafta süren denel işlemde 18'i deney 18'i kontrol grubunda olmak üzere toplam 36 (72 ders saati) gözlem gerçekleştirilmiş, gözlem verileri yapıldıkları gruba ve haftalara göre etiketlenmiş (ör. Deney grubu için D-GZ, 3. hafta, kontrol grubu için K-GZ, 17. hafta) gözlemlere ek olarak doğrudan alıntılardaki veri kaybını önlemek için kamera kayıtlarından yararlanılmıştır. Elle tutulan gözlem notları kamera kayıtlarındaki doğrudan alıntılarla güçlendirilerek elektronik ortama aktarılmış ve toplam 84 sayfa elektronik gözlem veri seti oluşmuştur. Öğretmenler ve öğrenciler için yazılan gözlem notlarının kime ait olduklarının anlaşılması adına paydaşlar sınıf oturma düzenine göre etiketlenerek gözlem formuna aktarılmıştır. Daha sonra elektronik olarak düzenlenen tüm nitel veriler, veri düzenleme ve etiketlemede yardımcı olması amacıyla Rampin ve Rampin (2021) tarafından geliştirilen “Taguette” isimli ücretsiz ve açık kaynak kodlu yazılıma aktarılmıştır (Şekil 4).

Project info	Documents	Highlights
	Add a document	
	01 DG 05.11.2021 Gözlem Notları.docx Edit	
	Backlight	
Durum	1-10 dak Öğretmen derse öğrencilerle sohbet ederek başlıyor ve daha sonra yoklama alıyor. Sohbet ve yoklama için yaklaşık 3 dakika ayrıldı. Öğretmen derse etkileşimli tahtada pastaçılık ile ilgili yaklaşık 4 dakikalık video ile başladı. Videoda pasta ustalarının bir günü anlatılıyor. Öğrencilerin videoyu dikkatle izlediği görülüyor. Tüm öğrencilerin yüzleri ET'ye dönük. Öğretmen daha sonra "şimdi kafalardaki soru şu, biz bunu neden izledik" diyor ve öğrencilere kimler pasta seviyor diye soruyor. Öğrencilerin tamamı el kaldırıyor. (Güğümler oluyor). Öğrencilere çikolata pasta sevenler mi yoksa diğer pastaları tercih edenlerden mi olduklarını soruyor. Öğrencilerden biri "Beyaz çikolatanın yanına meyve daha iyi gidiyor. Aromasına göre" diye yanıt veriyor. Öğretmen daha sonra "Hiç ailesiyle ya da arkadaşlarıyla pasta yapımına katılan var mı?" diye soruyor öğrencilere. Öğrencilerden 3'ü (tamamı kız öğrenci) el kaldırıyor. Öğretmen "Siz mi yaptınız, yardımcı mı oldunuz?" diye soruyor. Öğrencilerden biri (3) "Kardeşimle birlikte yaptık" diye yanıt veriyor. Öğretmen tekrar öğrencilere "Sizce bu videoyu neden izlemiş olabilirsiniz?" diye tekrar soruyor. Öğrencilerden biri (2) "pasta yapacağız" diyor gülerken. Öğrenci (11) "printerda basacağız", öğrenci (4) "acaba aşamalarını mı öğreneceğiz yani planlamayı mı öğreneceğiz" öğrenci (14) "uç boyutlu kalıbını mı basacağız hocam" öğretmen daha sonra soruyu biraz daha açarak "Bizim bu dersimizle ne alakası olabilir pastaçılığını?" diye soruyor. Öğrenci (10) "bence alakası var hocam, önce internette gerekli malzemeleri alıyoruz öncelikle ve kodlamada da benzer kaynaklara bilgilere ulaşıyoruz" öğrenci (4) "bizim işimiz bir anlamda problem çözmek ve pastaçılıkta bir problemi çözüyoruz" Öğretmen yanıt vermeden önce bir düşünelim bakalım diyor.	Öğretmenin sınıfı rahat tavırları dikkat çekiyor. Geçen yıl pilot uygulamayı gerçekleştirmiş olması bunda etkili olmuş olabilir. Öğretmen etkinliği planlarındaki sıralamaya uygun ilerliyor. Etkinlik planını ayrıntılı olarak gözden geçirmiş olabilir. Dikkat Çekme Tutum -Duyuşsal -Eğlenceli zaman Tutum -Duyuşsal -Eğlenceli zaman Metafor -Bileşenler -İlişki
Durum	10-23 dak Öğretmen gür sesli bir öğrenciyi ekrandaki durumu okuması için çağırıyor. (O sırada sınıf defteri geliyor. Ders bir kaç sn. bölünüyor). Öğrencilerin dikkati dağılmış görünmüyor. Öğrenci durumu okumaya başlıyor. Durumu duraksamadan hızlı olarak okuduğu görülüyor. Öğretmen müdahale ederek durumun önemli bulunduğu bölümlerin altını çiziyor. Öğretmen durumda geçen pasta ödevini duyunca öğrencilere "bende böyle bir ödev mi versem acaba". Öğrenci (2) "Hayır hocam, Belki ödevi ben daha sonra yiyeceksem olabilir" diye yanıt veriyor. (Sınıf gülüyor) Öğretmen "Ödevleri toplar, hepsini afiyetle yerim" diye ekliyor. Öğrenci durumu okumaya devam ediyor. Öğretmen durumda geçen ellerindeki kağıt ifadesinin ne anlatacağını soruyor. Öğrenciler hep bir ağızdan "tarif" cevabını veriyorlar. Öğretmen "Tarifin kaçta ayrıldığını ve tarife geçen malzemeler ve hazırlanışa ilişkin dikkatlerini çeken bir şeyler olup olmadığını soruyor". Öğrenci (3) "ölçüler dikkatimi çekiyor ve isteğe göre değişebilir yazıyor bazılarında. Miktarı değiştirebilirsin" Öğretmen "peki diğerleri isteğe göre değişebilir yazmıyorlar?" öğrenci (5) "onlar sabit, değişmeyen" olarak yanıt veriyor. Öğrenci 1 "örneğin 1 bardak su yerine 1.5 bardak su kullanırsak kıvamı değişir. Örneğin 1 bardak su yerine 1.5 bardak su kullanırsak kıvamı değişir."	Tutum -Duyuşsal -Eğlenceli zaman Metafor -Bileşenler -Kaynak Pastaçılık alanı öğretmenin de ilgisini çekmiş benziyor. Bu alana Metafor

Şekil 4. Rampin ve Rampin (2021) tarafından geliştirilen yazılım aracılığıyla verilerin analizine ve düzenlenmesine bir kesit

3.6.2.2. Ortaya Çıkan Fikirleri Okuma ve Not Etme

Nitel veriler deşifre edildikten ve düzenlendikten sonra tüm veri setine yönelik bütüncül bir bakış açısı elde etmek amacıyla bir hafta içerisinde okunmuş geçici kodların ve temaların neler olabileceğine dair notlar alınmıştır. Bu sırada gözlem formunda yer alan yansıtıcı notlar bölümündeki alan notları da dikkate alınmıştır. Dokümanın üzerine, dokümanın geneli ve bütün dokümanlar hakkında notlar alma stratejisi uygulanmış, oluşan olası kodlar ve temalar alanyazın desteği ile incelenerek teyit edilmeye çalışılmıştır.

3.6.2.3. Kodları Tanımlama ve Temalar Halinde Sınıflandırma

Nitel verilerin analiz sürecine verilerin kodlanmasıyla başlanmış, veri seti, öncelikle alanyazın desteği ile oluşan listeden bağımsız olarak Taguette yazılımıyla kodlanmıştır. Süreçte, birbiriyle ilişkili alt kodlar bir araya gelerek kodları, birbiriyle ilişkili kodlar bir araya gelerek alt tema ya da temaları oluşturmuş, oluşan tema ya da alt temaları anlamlı biçimde kapsayacak kategori alanyazın desteğiyle belirlenmiştir. Ortaya çıkan taslak tema, alt tema, kod ve alt kodlar daha önceden alanyazın desteği ile oluşturulan liste ile karşılaştırılarak gözden geçirilmiş ve düzeltmeler yapılmıştır. Sonrasında taslak tema, alt tema, kod ve alt kodlar tez izleme kuruluna gönderilerek görüşleri alınmış, geribildirimler doğrultusunda temaların araştırmanın hangi alt problemleri ile ilişkilendirilebileceği kararlaştırılmıştır.

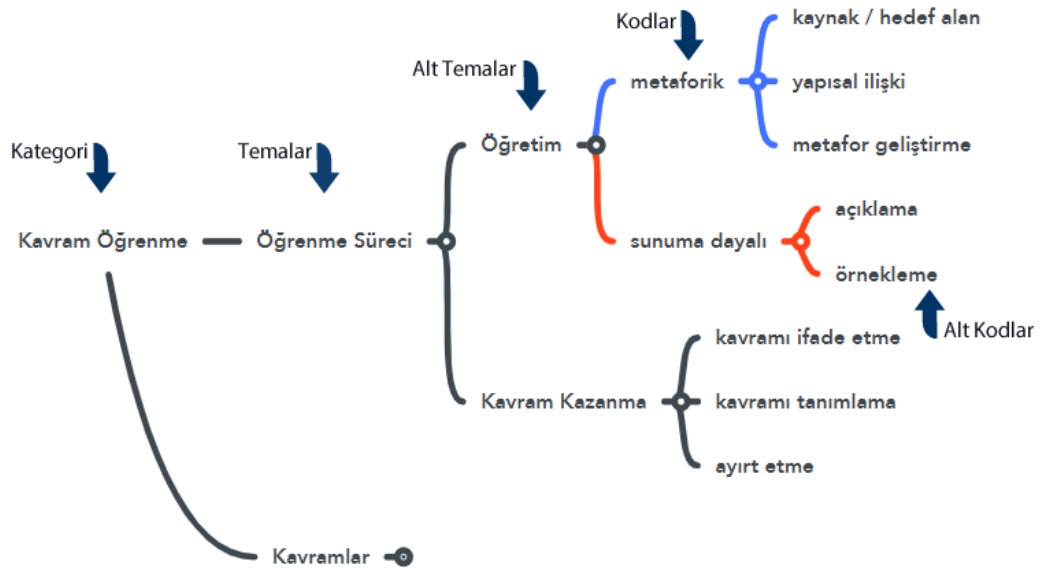
3.6.2.4. Yorumları Geliştirme ve Değerlendirme

Araştırmada nitel verileri yönetme ve düzenlemede, kodları tanımlama ve temalar halinde sınıflamada bir araç olarak kullanılan Taguette yazılımı, bileşenleri aracılığıyla yorumların geliştirilmesi ve değerlendirilmesine de katkıda bulunmuştur. Özellikle yazılımda yer alan belgeler (documents) sekmesi aracılığıyla farklı veri toplama araçlarına ait deşifreleri bir arada görmek, vurgulama (highlights) sekmesi aracılığıyla da herhangi bir kodlamaya ait farklı veri toplama araçlarından elde edilen alıntıları karşılaştırmak mümkün olabilmektedir. Bu sayede hem çeşitleme hem de geçerlik teyidi sağlanmış, aynı zamanda bulguları alıntılarla destekleyen raporlama kolaylaşmıştır. Nitel araştırmanın doğası gereği araştırmacının veri toplama sürecinin bir parçası olması (Busetto ve diğ., 2020; Fink, 2000) verilerin yorumlanmasında ve

çözümlemesinde etkili olmuş, bulguları açıklarken neden-sonuç ilişkilerini kurmaya katkı sağlamıştır.

3.6.2.5. Verileri Temsil Etme ve Görselleştirme

Veri analiziyle ulaşılan kodları, alt temaları, temaları ve bunlar arasındaki ilişkileri görsel olarak ortaya koymak için bir web aracından yararlanılmıştır. Web aracında yer alan zihin haritası (mind map) oluşturma bağlantısı sayesinde kategori, temalar, alt temalar, kodlar ve alt kodlar arasındaki ilişkiler gösterilebilmiş, bütüncül bir bakış açısı sağlanmıştır. Nitel verilerin analizinde ortaya çıkan birimlerin daha iyi anlaşılabilmesi için Şekil 5'te *Kavram Öğrenme* kategorisini oluşturan tema, alt tema, kod ve alt kodlara ait şemanın bir bölümü örnek verilmiştir.



Şekil 5. Verilerin görselleştirilmesine yönelik bir kesit

Şekil 5'te de görüldüğü gibi, tümevarımsal bir yaklaşımla nitel verilerin kodlanması sonucu ortaya çıkan “kaynak/hedef alan”, “yapısal ilişki” ve “metafor geliştirme” alt kodları metaforik kodunu; metaforik ve sunuma dayalı kodları öğretim alt temasını; öğretim alt teması ve kavram kazanma alt temaları öğrenme süreci temasını; kavramlar ve öğrenme süreci temaları da kavram öğrenme kategorisini oluşturmuştur. Bulgular bölümünde kodlar ve alt kodlar tırnak işareti (ör. “yapısal ilişki”, “metaforik”) ile, alt temalar koyu (ör. **öğretim**, **kavram kazanma**), temalar ise herhangi bir biçimlendirme yapılmadan, 4. düzey başlığı şeklinde sunulmuştur.

3.7. Geçerlik ve Güvenirlik

Araştırmalarda kullanılan yöntemin kalitesi, çalışmaya olan önyargıların geçerlik ve güvenirlilik önlemleri ile nasıl en aza indirildiğiyle doğrudan ilişkilidir (Hong ve diğ., 2019). Guba ve Lincoln (1982), araştırmanın nitel bölümleri için geçerlik ve güvenirlilik yerine inandırıcılık (trustworthiness) ifadesinin kullanılmasını önermekte; inandırıcılığı inanılrlık, güvenilebilirlik, onaylanabilirlik ve aktarılabirlik olmak üzere dört ana başlıkta sınıflamaktadır. Araştırmada karma yöntem deneysel desen işe koşulduğundan geçerlik ve güvenirlilik için alınan önlemlerin ayrı başlıklar altında sunulması uygun görülmüştür.

3.7.1. Geçerlik

Geçerlik ve kalite ile ilgili konular, karma yöntem araştırmalarında uzun süredir tartışılan ana sorunlardan biridir ve araştırmanın en önemli yönü olarak tanımlanmaktadır (Tashakkori ve Teddlie, 2003). Karma yöntemde, nitel ve nicel verilerin veri çeşitlemesi amacıyla birlikte kullanılması geçerliği belirlemede diğer yöntemlerden farklı yollar izlenmesine neden olmaktadır (Zohrabi, 2013). Creswell ve Plano Clark (2018) hazırladıkları bir rehber aracılığıyla her desene uygun geçerlik tehditleri, uygulanabilecek stratejiler ve bu tehditleri en aza indirmek için yapılabilecekleri tanımlamaktadır. Bu araştırmanın deneysel deseni olarak tanımlanan karma yöntem deneysel desenine yönelik önerilere de yer veren rehberde geçerlik tehditleri ve bu tehditleri en aza indirmeye yönelik stratejiler Tablo 8’de sunulmaktadır.

Tablo 8. *Karma Yöntem Deneysel Desen Geçerlik Tehditleri ve Tehditleri En aza İndirme Stratejileri (Creswell ve Plano Clark, 2018)*

Desenin Tipi	Geçerlik tehditleri	Tehditleri en aza indirme stratejileri
Karma Yöntem Deneysel Desen Tasarımı	Deneysel bir tasarımda dış ya da iç geçerliliğe yönelik tehditleri ele almama	Deneysel tasarımla ilgili alanyazında belirtilen iç ve dış tehditleri ele alma
	Nitel bileşenin deneyde neden ve nerede gömülü olduğunu belirtmeme	Deneysel desende nitel verilerin toplanması ve kullanılması için açık bir gerekçe sunma
	Deney sırasında toplanan nitel verilerin deneysel tasarımda önyargı oluşturmaması	Denel işlem sırasında deney sonuçları hakkında önyargılara yol açabilecek veri toplama stratejileri kullanmama

3.7.1.1. Deneyisel tasarımıyla ilgili alanyazında belirtilen iç ve dış tehditleri ele alma

Deneyisel tasarıma yönelik alanyazın incelediğinde Campbell ve Stanley'nin (1963) deneyisel ve yarı deneyisel çalışmalarda iç ve dış geçerliği tehdit eden faktörlere ilişkin açıklamaları genel kabul görmüştür. Campbell ve Stanley (1963) iç geçerliğe yönelik sekiz, dış geçerliğe yönelik dört olmak üzere toplam 12 faktör tanımlamış, iç geçerliği etkileyen faktörler zaman (history), olgunlaşma (maturation), deney öncesi testler (testing), ölçme araçları (instrumentation), istatistiksel regresyon (statistical regression), yanlış gruplama (selection), deney kaybı (experimental mortality), gruplama ve olgunlaşma etkileşimi (selection-maturation interaction) olarak sıralanmıştır. Aşağıda bu araştırmada iç geçerliğe yönelik belirtilen bu tehditler için alınan önlemler açıklanmıştır.

Zaman: Çalışmanın denel uygulaması 36 ders saati ve 18 haftadan oluşmuş, denel işlem süresinin belirlenmesinde bilgisayar bilimi (Kur-1) dersi öğretim programında belirtilen süre dikkate alınmış, denel işlem için ayrılan süre programlama ünitesine programda ayrılan süre ile aynı tutulmuştur. Bu durumun denel işlemin öğrenme sürecinin doğal bir parçası olarak algılanmasını sağlayabileceği düşünülmüştür.

Olgunlaşma: Öğrencilerin yaş grubu ve sosyo-ekonomik çevre açısından benzer olmaları durumunda, sınırları belli bir zamanda fizyolojik ve psikolojik açıdan benzer biçimde olgunlaşabilecekleri kabul edilerek deney ve kontrol grubunda yer alan öğrencilerin aynı okul türü ve sınıf düzeyinden seçilmesine dikkat edilmiştir.

Deney öncesi testler: Araştırmada karma yöntem deneyisel desen kullanıldığından deney ve kontrol grubu öğrencilerine denel işlem öncesi birtakım testler uygulanmıştır. Araştırmanın yöntem bölümünde yer alan deney ve kontrol grubunun denkliğine ilişkin bulgular deney ve kontrol grubundaki öğrencilerin denk olduğunu göstermiştir.

Ölçme araçları: Araştırmada kullanılan ölçme araçlarından kaynaklanabilecek geçerlik tehditlerinin önüne geçebilmek için ölçme araçlarının oluşturulmasında alanyazından yararlanılmış, uzman görüşü alınmış, deneme uygulamaları gerçekleştirilmiş ve denel işlem boyunca deney ve kontrol grubunda yer alan öğrencilere aynı nicel ve nitel ölçme araçları uygulanmıştır. Araştırma süreci boyunca

deney grubunda gerçekleştirilen tüm süreçlerin (denel işlem materyali hariç) kontrol grubu ile aynı olması ve ölçme araçlarının puanlanmasında puanlayıcılar arası uyumun istatistiksel olarak hesaplanması geçerliğe katkı sağlamıştır.

İstatistiksel regresyon: Deney ve kontrol grubunda yer alan öğrencilerin denel işlem öncesinde grup içlerinde normal dağılım gösterip göstermedikleri ve grupların denk olup olmadıklarına yönelik istatistiksel analizler yapılmış, böylece gruplarda yer alan bireylerin aşırı uç ölçüm sonuçlarından elde edilen bireyler olmadığı gösterilmiştir.

Yanlı gruplama: Araştırmada deney ve kontrol grubunda yer alan öğrencilerin denk olup olmadıkları grupların denkliğine yönelik analizlerle belirlenmiştir. Bu durumun birbirinden farklı deney ve kontrol gruplarının oluşmasının ve uygulama sonunda denel işlemin sonuçlarının yanlı olarak karşılaştırılmasının önüne geçtiği düşünülmektedir. Araştırmacı, deney grubu öğrencilerinin denel işlem materyalini kontrol grubu öğrencileriyle paylaşma ihtimali, aynı okuldaki grupların birbirlerinden etkilenme olasılığı, aynı öğretmenin farklı sınıflarda farklı öğretim stratejileri uygulamasının zorluğu ve anadolu liselerinde genelde sadece bir bilgisayar bilimi dersi öğretmenin görev alması nedeniyle farklı okullardan seçerek iç geçerliğe yönelik tehditlerin önüne geçebilmeyi amaçlamıştır.

Denek kaybı: Araştırmada denel işlem öncesi, sırası ve sonrasında oluşabilecek denek kayıplarına ilişkin dikkatli olunmuş, olası denek kayıplarını önlemek adına gruplara birden fazla sınıf dâhil edilmiştir. Denel işlem öncesinde deney ve kontrol grubunda yer alan öğrencilerden okul değiştiren ya da sürekli devamsızlığı bulunan öğrenciler araştırmanın dışında bırakılmıştır.

Gruplama ve olgunlaşma etkileşimi: Hem grupların belirlenmesinde yanlılığın önüne geçilmesine yönelik çalışmalar hem de birbirine yakın yaş ve benzer sosyo-ekonomik düzeye sahip öğrencilerin yer aldığı deney ve kontrol grubu ataması yoluyla yanlı gruplama ve olgunlaşmanın ortak etkisi en aza indirilmeye çalışılmıştır.

Araştırmada, nitel veri toplama araçlarının ve araştırma deseninin iç geçerliğini tehdit edebilecek durumlara karşı ise Miles ve Huberman'ın (1994) ve Guba ve Lincoln'un (1982) önerileri doğrultusunda alınan önlemlere aşağıda yer verilmiştir:

- Araştırmada elde edilen bulguların denel işlemin gerçekleştiği ortam açısından tanımlanabilirliğini ortaya koymak adına deney ve kontrol grubunun bulunduğu ortama ve araştırma bulgularına ayrıntılı olarak yer verilmiş, denel işlem öncesi öğretmen görüşmeleriyle denel işlem ortamının ve araştırma bulgularına olası etkilerinin tanımlanması amaçlanmıştır.

- Nitel veri toplama araçları, bulguların tutarlı ve kendi içinde anlamlı olması adına araştırma soruları kapsamında ve alanyazın desteğiyle oluşturulmuş, görüşme formlarını oluştururken araştırma sorularıyla ilişkisini kurmak için matris hazırlanmıştır.

- Araştırmada nitel veriler, veri çeşitlemesi amacıyla görüşme, gözlem ve doküman olarak değerlendirilebilecek SD formları gibi farklı yöntemlerle toplanmış, elde edilen verilerin analizi sonucunda ulaşılan bulgularda doğrudan alıntılara sıklıkla yer verilmiştir.

- Araştırmanın nitel veri toplama araçlarının oluşturulmasında ve bu araçlar aracılığıyla elde edilen nitel verilerin analizinde alanyazın desteğinden sürekli olarak yararlanılmış, bu durum bulguların araştırma sorularında belirlenen kuramsal çerçeve kapsamında bulgular elde edilmesine yardımcı olmuştur.

- Denel işlem öncesi ve sonrasında yapılan öğrenci ve öğretmen görüşmeleri sonrası katılımcılara görüşmeye eklemek ya da çıkarmak istedikleri ifadeler olup olmadığı sorulmuş, yanıtları dikkate alınarak daha geçerli bulgular hedeflenmiştir.

- Araştırmacı denel işlem boyunca deney ve kontrol grubunda 18 haftalık gözlem gerçekleştirmiştir. Araştırmacının bu süre boyunca araştırmanın gerçekleştiği ortamda bulunması uzun süreli etkileşimi mümkün kılmış, böylece araştırmacının sınıf ortamını ve kültürünü tanıması, sınıfta karşılıklı güvene dayalı bir ilişki kurulması ve daha samimi veriler toplanmasını olanaklı hale getirmiştir.

- Gözlem notlarına ek olarak doğrudan kanıtlar sağlaması amacıyla denel işlem boyunca dersler kayıt altına alınmış, bulgularda doğrudan alıntılara yer vererek inandırıcılık artırılmaya çalışılmıştır.

Nicel ve nitel araştırmalarda farklılık gösteren geçerlik her iki yaklaşımda verilerin kalitesinin, sonuçların ve yazarın sonuçlara yönelik yorumunun kontrol edilmesi ile ilgilidir (Creswell ve Plano Clark, 2018). Campbell ve Stanley (1963),

çalıřmalarda dıř geerlięe ynelik oluřabilecek tehditleri lmenin tepki veya etkileřimi (the reactive or interaction effect of testing), yanlı seim ve baęımsız deęiřken etkileřimi (the interaction effects of selection biases and the experimental variable), deneme etkisi (reactive effects of experimental arrangements) ve oklu mdahale giriřimi (multiple-treatment interference) olarak tanımlamıřtır. Ařaęıda bu arařtırmada dıř geerlięe ynelik tehditlere karřı alınan nlemler Campbell ve Stanley'in (1963) sınıflaması dikkate alınarak aıklanmıřtır.

lmenin tepki veya etkileřimi: Arařtırmada deney ve kontrol grubunda yer alan ęrencilere uygulanan n testler ve son testler ile denel iřlemde gerekleřtirilen gzlemlerin neden yapıldıęı aık bir řekilde belirtilmiř, ęrencilerin zerinde bir baskı unsuru yaratmaması iin dersi veren ęretmenler tarafından notla deęerlendirilmeyeceęi belirtilmiřtir.

Yanlı seim ve baęımsız deęiřken etkileřimi: Campbell ve Stanley'e (1963) gre yanlı seim, i geerlik tehdidi oluřturduęu gibi baęımsız deęiřken aracılıęıyla elde edilen verilerin genellenebilirlięini de tehlikeye atmaktadır. Bu arařtırmada, i geerlik nlemi olarak sunulan grupların yansız seimi ve grupların denklięinin istatistiksel olarak tanımlanması, dıř geerlięe ynelik nlem olabilir.

Deneme etkisi: Deneme etkisi genellikle alıřmanın bulgularının genellenebilirlięi sorununu gndeme getirmektedir. Genellenebilirlik sorununu en aza indirmek iin arařtırmada, ilk olarak denel iřlemin gerekleřtięi ortam aık bir řekilde tanımlanmıř, deney ve kontrol grubunda yer alacak ęrencilerin sayısını arttırmaya ynelik aba harcanmıřtır. Ayrıca denel iřlemin gerekleřtirilmesi iin planlanan 18 hafta hem denel iřlemin hem de srete yer alan arařtırmacının ęretim srecinin doęal bir parası haline gelmesi iin yeterli bir sre olarak kabul edilmiřtir.

oklu mdahale giriřimi: Deney grubuna programlama ęretimi arařtırmacı tarafından geliřtirilen metafora dayalı ders planları aracılıęıyla gerekleřtirilmiř, kontrol grubunda ise herhangi bir mdahalede bulunulmamıřtır. Deney grubunun sadece uygulanan ders planları aısından farklılařmasına zen gsterilirken gerekleřtirilen denel iřlem srecine arařtırmada ayrıntılı biimde yer verilmiřtir.

Araştırmada nitel veri toplama araçlarının ve araştırma deseninin dış geçerliğine yönelik olarak ise Miles ve Huberman'ın (1994) ve Guba ve Lincoln'un (1982) dış geçerlik önerileri dikkate alınmıştır:

- Araştırmada, nitel veri toplama araçlarının desendeki yerine hem grafiksel hem de açıklamalarla yer verilerek deseni oluşturan bileşenler arasındaki ilişkiler açıkça ortaya konulmuştur.

- Araştırma sonuçlarının başka kişi ya da gruplara aktarılabilirliğini sağlamak amacıyla çalışma grubunda yer alan öğrencilerin araştırmaya nasıl dâhil oldukları ya da neden dâhil edilmediklerini belirleyen ölçütlere ayrıntılı yer verilmiştir.

- Nitel veri toplama araçlarının geliştirilmesinde izlenen taslak araçların oluşturulmasına, uzman görüşlerinin alınmasına ve deneme uygulamalarına yönelik süreçler açıklanmıştır.

- Geliştirilen nitel veri toplama araçlarının kapsamını belirlerken araştırma sorularından ve araştırmanın kuramsal temellerinin dayandığı kavramsal metafor kuramından yararlanılmış, veri toplama araçlarının birbiriyle tutarlılık göstermesine dikkat edilmiştir.

- Araştırmada elde edilen bulgular ile denel işlemin gerçekleştiği ortamın ve bağlamın uygunluğu tekrarlı okumalarla test edilmiştir.

3.7.1.2. Deneyisel desende nitel verilerin toplanması ve kullanılması için açık bir gerekçe sağlama

Bu araştırmada nitel veriler denel işlem öncesinde, sırasında ve sonrasında farklı amaçlarla kullanılmıştır. Denel işlem öncesinde öğretmenlerle yapılan görüşmelerle deney ve kontrol grubunun içinde bulunduğu ortam ve bağlamın tanımlanması, denel işlem sırasında yapılan gözlem ve SD formu aracılığıyla denel işleme yönelik katılımcıların deneyimlerinin ortaya çıkarılması, denel işlemin uygunluğunun belirlenmesi ve çalışma grubunda yer alan öğrencilerin karşılaştıkları kolaylaştırıcı ve zorlayıcı durumların tanımlanması, denel işlem sonunda yapılan öğrenci ve öğretmen görüşmeleri ile de denel işlem çıktılarının anlaşılması ve denel işlem materyalinin nasıl çalıştığının ortaya konulması amaçlanmıştır.

3.7.1.3. Deney sırasında rahatsız edici veri toplama stratejilerini göz önünde bulundurma

Araştırmada geliştirilen tüm nitel veri toplama araçlarının geçerli ve güvenilir olmaları için izlenmesi gereken yollar alanyazın desteği ile gerçekleştirilmiş, veri toplama süreci ayrıntılı olarak tanımlanmıştır. Araştırmada elde edilen verilerin nasıl analiz edildiği ve analiz sürecine ilişkin açıklamalara yer verilmiş, bulgulara analiz sonuçları çalışmanın alt problemleri ile ilişkilendirilerek sunulmuştur. Toplanan nitel veriler bir hafta içinde deşifre edilmiş, okunmuş ve analiz süreci başlatılarak deney tasarımı tehlikeye düşürecek durumlar oluşmadan müdahale edilmeye çalışılmıştır.

3.7.2. Güvenirlilik

“Güvenirlilik, genel anlamda araştırmalarda aynı süreçlerin izlenmesi ile aynı sonuçların alınması” (Karasar, 2009, s. 148) olarak tanımlanabilir. Nitel araştırmalarda güvenirlilik önemli olmakla birlikte, nicel araştırmada geçerli olan bazı etkenler nitel araştırmalarda geçerli değildir (Yıldırım ve Şimşek, 2011). Bu nedenle, karma yöntem deneysel desenle yürütülen bu araştırmada nicel veri toplama araçlarının güvenirliliğine ilişkin bilgilere istatistiksel olarak veri toplama araçlarının açıklandığı bölümde, nitel veri toplama araçları ve araştırma desenine yönelik güvenirliliği tehdit edebilecek risklere karşı alınan tedbirlere ise Miles ve Huberman (1994) ve Lecompte ve Goetz’in (1982) önerileri dikkate alınarak aşağıda yer verilmiştir. Araştırmada nitel veri toplama araçları ve desendeki yerine yönelik iç güvenirliliği sağlamak adına şu stratejiler uygulanmıştır:

- Araştırma soruları açık bir şekilde ifade edilmiş ve raporlamada araştırma soruları dikkate alınmıştır.
- Görüşme formunda yer alan taslak kodlarla, görüşme ve süreç değerlendirme formlarında yer alan soruların oluşturulmasında kavramsal metafor ve programlama öğretimine yönelik kuramsal çerçeveden yararlanılmış; veri toplama araçlarından elde edilen verilerin nasıl analiz edildiği adım adım açıklanmıştır.
- Verilerin toplanmasında ve analizinde araştırma kapsamı dışında kalan veriler temizlenmiş, geçerli olmayan veriler ayıklanmıştır.
- Araştırmada toplanan nitel verilerin analizleri ve elde edilen sonuçlar tez izleme kuruluyla paylaşılmış, geribildirim alınmıştır.

- Araştırmada gerçekleştirilen gözleme ilişkin veriler, denel işlem öncesi ve sonrası görüşme verileriyle karşılaştırılarak sunulmuştur.
- Görüşme ve süreç değerlendirme formları ile elde edilen veriler bulgular bölümünde doğrudan alıntılarla sunulmuş, bulguların yorumlanması araştırmanın sonuç ve tartışma bölümüne bırakılmıştır.
- Araştırmacı, araştırma sürecindeki rolünü ve yerini yöntem bölümünde açıklayarak araştırmaya bakış açısını ve yaklaşımını tanımlamıştır.
- Karma yöntem deneysel desenini temele alan araştırmada, nitel verileri toplamada kullanılan araçlara ve bu araçların desen içerisindeki yerine ait dış güvenilirliğe ilişkin alınan önlemlere yer verilmiştir:
 - Araştırmacı, deney ve kontrol grubunda gerçekleştirilen gözlemler boyunca ortamda katılımcı gözlemci olarak bulunmuş, çalışılan duruma ilişkin deneyimlerini denel işlem materyallerini hazırlamada kullanmıştır.
 - Araştırmanın gerçekleştiği iki anadolu lisesine ve denel işlemin uygulandığı sınıflara ait fiziksel ve sosyal ortam tanımlanmış, bu tanımlamada öğretmenlerle denel işlem öncesi yapılan görüşmelerden yararlanılmıştır.
 - Görüşme, gözlem ve SD formu aracılığıyla çalışma grubunda yer alan öğrenciler ve öğretmenlerden verilerin nasıl toplandığı, analiz edildiği ve bulgulara nasıl ulaşıldığı açıkça belirtilmiştir.

3.8. Araştırmacının Rolü ve Etik

Araştırmacı, örgün eğitimin farklı kademelerinde 19 yıllık programlama öğretimi deneyimine sahip olması ve önceden nicel ve nitel araştırmalar yürütmesi nedeniyle programlama konusunda karma yöntem araştırmalarını yürütmek için yeterli deneyime sahip olduğunu düşünmektedir.

Araştırmacının daha önce bir özel okulun görsel sanatlar ve beden eğitimi derslerine yönelik programlarının geliştirilmesinde rol alması, bu sırada öğretmenlerin yapılandırmacı öğrenme yaklaşımına uygun etkinlikler geliştirmelerine rehberlik etmesi ve Türkiye’de bilişim teknolojileri alanında yayımlanan öğretim programlarını ayrıntılı biçimde analiz ederek incelemesi denel işlem materyalini oluşturan ders planlarının bilgisayar bilimi (Kur-1) dersi öğretim programına ve yapılandırmacı öğrenme yaklaşımına uygun geliştirmesine yardımcı olmuştur.

Çoğu lise düzeyinde olmak üzere 16 yıllık bilişim teknolojileri öğretmenliği deneyimi araştırmacının denel işlemi planlamasında programlama öğretimi sorunlarıyla nasıl baş edilebileceği ve öğrenci düzeyine uygun etkinlikler tasarlama gibi konularda katkıda bulunmuştur.

Araştırmacı, veri toplama sürecinde bütün gözlemlere ve katılımcılarla görüşmelere doğrudan katılmış, aynı ortamı paylaşmış ve edindiği deneyim ve izlenimleri verileri anlamlandırmada kullanmıştır.

Araştırma, araştırmacının da öğretmen olarak bulunduğu ilde ve görev aldığı okul dışında iki anadolu lisesinde gerçekleşmiştir. Bu nedenle araştırmacı araştırma sürecini yöneten, öğretmen eğitimlerinde görev alan, denel işlemin tasarlandığı şekliyle uygulanmasını kontrol eden ve araştırmanın geçerli ve güvenilir sonuçlar üretebilmesi için önlemler alan roledir. Bu durum araştırmacının tarafsız ve nesnel kalmasını, kapsamlı veriler toplamasını kolaylaştırmıştır. Aynı zamanda araştırmacı araştırma sürecinin yakın takipçisi ve katılımcı gözlemcisidir.

IV. BÖLÜM

BULGULAR

Bulgular, “Programlama Kavramlarını Öğrenme”, Programlama Becerileri” ve “Bilgi İşlemsel Düşünme Öz Yeterliliği” olmak üzere araştırmanın alt problemleriyle ilişkili üç ana başlık altında ele alınmıştır. Her bir alt başlıkta sırasıyla deney ve kontrol grubu karşılaştırmaları, nitel kanıtlar ve nitel ile nicel kanıtların ilişkisi incelenmiştir.

4.1. Programlama Kavramlarını Öğrenme

Araştırmanın ilk alt problemi “Metin tabanlı programlama öğretiminde metafor kullanımı, programlama kavramlarını öğrenmeyi nasıl etkilemektedir?” şeklinde tanımlanmış; ilgili alt problemi yanıtlamak için a) Deney ve kontrol gruplarındaki öğrencilerin programlama kavramlarını öğrenme ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır? b) Programlama kavramlarını öğrenme sürecini ve kavramsal değişimi ortaya koyan nitel kanıtlar nelerdir? ve c) Programlama kavramlarını öğrenme sürecine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır? biçiminde üç ayrı soru oluşturulmuştur. Aşağıda, ilgili alt sorulara yönelik ulaşılan bulgular sırasıyla sunulmuştur.

4.1.1. Deney ve Kontrol Grubu Karşılaştırmaları

Deney ve kontrol grubu öğrencilerinin programlama kavramlarını öğrenme düzeylerinin belirlenmesi amacıyla uygulanan KHGD formu programlamanın temelleri, program kontrolü, veri yapıları ve dosya işlemleri bileşenlerinden oluşmaktadır. Bu bileşenlere yönelik ön test-son test puanlarının karşılaştırılmasıyla ulaşılan bulgulara aşağıda yer verilmiştir.

Programlamanın temelleri, programlamaya giriş kavramlarını ve ilkelerini içermektedir (MEB, 2018c). Deney ve kontrol grubu öğrencilerinin programlamanın temelleri ön test-son test puanlarının ortalaması ve standart sapmasına Tablo 9’da yer verilmiştir.

Tablo 9. Deney ve Kontrol Grubunun Programlamanın Temelleri Ortalama ve Standart Sapma Değerleri

Grup	Ön test			Son test		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	2.58	1.00	66	2.62	1.54
Kontrol	69	2.51	1.21	69	2.76	1.04

Tablo 9’da deney ve kontrol grubu son test ortalama puanlarının ön test ortalama puanlarına göre yüksek olduğu görülebilir. Deney ve kontrol grubunda gözlemlenen bu değişimin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemek için Karışık Ölçümler için ANOVA (Mixed ANOVA) gerçekleştirilmiş, deney ve kontrol grubu öğrencilerinin programlamanın temellerine yönelik kavramları öğrenme düzeyleri denel işlem öncesinden sonrasına anlamlı bir fark göstermemiştir, $[F_{(1, 133)} = .67, p = .41]$. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının programlamanın temellerine yönelik kavramları öğrenme düzeylerini etkilemediğini ortaya koymaktadır.

Program kontrolü; koşullu durumlar, döngüler ve fonksiyonlar gibi programın istenilen sonuçları üretmesini sağlayan mekanizmaları içermektedir (MEB, 2018c). Öğrencilerin program kontrolü kavramlarını öğrenme düzeylerine ilişkin ön test-son test puanlarının ortalaması ve standart sapmasına Tablo 10’da yer verilmiştir.

Tablo 10. *Deney ve Kontrol Grubunun Program Kontrolüne İlişkin Ortalama ve Standart Sapma Değerleri*

Grup	Ön test			Son test		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	.00	.00	66	3.37	1.31
Kontrol	69	.04	.13	69	.49	.77

Tablo 10’da görüldüğü üzere deney grubu program kontrolü ön test puanı sıfırdır. Bunun nedeni çalışma grubunda yer alan öğrencilerin denel işlem sürecinin başında olmaları ve daha önce bu kavramlarla hiç karşılaşmamış olmaları olabilir. Deney ve kontrol grubu son test ortalama puanlarının ise ön test ortalama puanlarına göre yükseldiği görülebilir. Deney ve kontrol grubu öğrencilerinde gözlemlenen bu değişimin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemeye yönelik gerçekleştirilen Karışık Ölçümler için ANOVA sonuçlarına Tablo 11’de yer verilmiştir.

Tablo 11. Program Kontrolü Ön test-Son test Puanları ANOVA Sonuçları

Varyansın Kaynağı	KT	Sd	KO	F	η_p^2
Gruplar arası	213.82	134			
Grup (Deney/Kontrol)	134.87	1	134.87	227.23*	.63
Hata	78.94	133	.59		
Grup içi	465.69	135			
Ölçüm (Ön test/Son test)	246.51	1	246.51	437.57*	.76
Grup*Ölçüm	144.25	1	144.25	256.05*	.65
Hata	74.92	133	.56		
Toplam	679.51	269			

* $p < .001$

Buna göre deney ve kontrol grubunda yer alan öğrencilerin program kontrolü kavramlarını öğrenme düzeylerinin denel işlem öncesinden sonrasına anlamlı bir fark gösterdiği, yani deney ya da kontrol grubunda olmak ile tekrarlı ölçüm faktörlerinin program kontrolü kavramlarını öğrenme üzerindeki ortak etkilerinin anlamlı olduğu bulunmuştur [$F_{(1,133)} = 256.05$, $p < .001$ $\eta_p^2 = .65$]. Ortak etki değeri olarak hesaplanan kısmi eta kare (η_p^2) değeri .14'ten yüksek olduğu için büyük (large) etki olarak yorumlanmıştır. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının program kontrolüne yönelik kavramları öğrenmeyi arttırmada farklı etkilere sahip olduğunu göstermiştir. Program kontrolü puanlarında denel işlem öncesine göre daha fazla kazanç elde eden deney grubunda uygulanan metafora dayalı öğretimin program kontrolüne yönelik kavramları öğrenmede kontrol grubuna göre daha etkili olduğu söylenebilir.

Programlamada veri yapıları; verileri organize etmek, depolamak ve işlemek için gerekli kavramları içermektedir (MEB, 2018c). Öğrencilerin veri yapıları kavramlarını öğrenme düzeylerine ilişkin ön test-son test puanlarının ortalaması ve standart sapması Tablo 12'de sunulmuştur.

Tablo 12. Deney ve Kontrol Grubunun Veri Yapılarına İlişkin Ortalama ve Standart Sapma Değerleri

Grup	Öntest			Sontest		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	.01	.08	66	1.27	1.39
Kontrol	69	.04	.15	69	.11	.31

Tablo 12’ye göre deney ve kontrol grubu son test ortalama puanlarının, ön test ortalama puanlarına göre daha yüksek olduğu görülebilir. Deney ve kontrol grubunun veri yapıları puanlarında gözlemlenen bu değişimlerin gruplar arasında anlamlı bir fark gösterip göstermediğini belirlemeye yönelik gerçekleştirilen Karışık Ölçümler için ANOVA sonuçlarına Tablo 13’te yer verilmiştir.

Tablo 13. Veri Yapıları Ön test-Son test Puanları ANOVA Sonuçları

Varyansın Kaynağı	KT	<i>Sd</i>	KO	<i>F</i>	η_p^2
Gruplar arası	92.08	134			
Grup (Deney/Kontrol)	21.47	1	21.47	40.45*	.23
Hata	70.61	133	.53		
Gruplar içi	118.73	135			
Ölçüm (Ön test/Son test)	29.85	1	29.85	61.67*	.31
Grup*Ölçüm	24.48	1	24.48	50.57*	.27
Hata	64.38	133	.48		
Toplam	210.82	269			

* $p < .001$

Tablo 13’e göre deney ve kontrol grubunda yer alan öğrencilerin veri yapılarına yönelik kavramları öğrenme düzeylerinin denel işlem öncesinden sonrasına anlamlı bir farklılık gösterdiği, yani deney ya da kontrol grubunda olmak ile tekrarlı ölçümler faktörlerinin veri yapıları kavramlarını öğrenme üzerindeki ortak etkilerinin anlamlı olduğu bulunmuştur [$F_{(1,133)} = 50.57$, $p < .001$ $\eta_p^2 = .27$]. Ortak etki değeri olarak hesaplanan kısmi eta kare (η_p^2) değeri .14’ten büyük olduğu için büyük (large) etki olarak yorumlanmıştır. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının veri yapılarına yönelik kavramları öğrenmeyi arttırmada farklı etkilere sahip olduğunu göstermiştir. Veri yapıları puanlarında denel işlem öncesine göre daha fazla kazanç elde eden deney grubunda uygulanan metafora dayalı öğretimin veri

yapılarına yönelik kavramları öğrenmede kontrol grubuna göre daha etkili olduğu söylenebilir.

Dosya işlemleri genel olarak diske veri yazma ve diskten veri okunması işlemlerini içermektedir (MEB, 2018c). Öğrencilerin dosya işlemlerine yönelik ön test-son test puanlarının ortalamalarına ve standart sapmalarına Tablo 14'te yer verilmiştir.

Tablo 14. *Deney ve Kontrol Grubu Dosya İşlemlerine İlişkin Ortalama ve Standart Sapma Değerleri*

Grup	Öntest			Sontest		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	.00	.00	66	1.02	1.37
Kontrol	69	.04	.22	69	.06	.48

Tablo 14'e göre deney ve kontrol grubu son test ortalama puanlarının, ön test ortalama puanlarına göre daha yüksek olduğu görülebilir. Deney grubunda dosya işlemleri ön test puanı sıfırdır. Bunun nedeni çalışma grubunda yer alan öğrencilerin denel işlem sürecinin başında olmaları ve daha önce bu kavramlarla hiç karşılaşmamış olmaları olabilir. Deney ve kontrol grubunun dosya işlemleri puanlarında gözlemlenen bu değişimlerin gruplar arasında anlamlı bir fark gösterip göstermediğini belirlemeye yönelik gerçekleştirilen Karışık Ölçümler için ANOVA sonuçlarına Tablo 15'te yer verilmiştir.

Tablo 15. *KHGD Formu Dosya İşlemleri Ön test ve Son test Puanları ANOVA Sonuçları*

Varyansın Kaynağı	KT	<i>Sd</i>	KO	<i>F</i>	η_p^2
Gruplar arası	90.53	134			
Grup (Deney/Kontrol)	14.09	1	14.09	24.52*	.15
Hata	76.44	133	.57		
Gruplar içi	101.66	135			
Ölçüm (Ön test/Son test)	18.57	1	18.57	37.32*	.21
Grup*Ölçüm	16.90	1	16.90	33.97*	.20
Hata	66.18	133	.49		
Toplam	192.19	269			

* $p < .001$

Tablo 15'e göre deney ve kontrol grubunda yer alan öğrencilerin dosya işlemlerine yönelik kavramları öğrenme düzeylerinin denel işlem öncesinden sonrasına anlamlı bir farklılık gösterdiği, yani deney ya da kontrol grubunda olmak ile tekrarlı ölçümler faktörlerinin dosya işlemleri kavramlarını öğrenme üzerindeki ortak etkilerinin anlamlı olduğu bulunmuştur [$F_{(1,133)} = 33.9, p < .001 \eta_p^2 = .20$]. Ortak etki değeri olarak hesaplanan kısmi eta kare (η_p^2) değeri .14'ten büyük olduğu için büyük (large) etki olarak yorumlanmıştır. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının dosya işlemlerine yönelik kavramları öğrenmeyi arttırmada farklı etkilere sahip olduğunu göstermiştir. Dosya işlemleri faktör puanlarında denel işlem öncesine göre daha fazla kazanç elde eden deney grubunda uygulanan metafora dayalı öğretimin dosya işlemlerine yönelik kavramları öğrenmede kontrol grubuna göre daha etkili olduğu söylenebilir.

4.1.2. Programlama Kavramlarını Öğrenmeye Yönelik Nitel Bulgular

Araştırmada denel işlem süreci öncesi öğretmen görüşmeleri, denel işlem sırasında yapılan gözlem ve öğrencilerden toplanan SD formları ve denel işlem sonunda öğrenci ve öğretmenlerle gerçekleştirilen görüşmeler aracılığıyla elde edilen verilerin analizi sonucu ortaya çıkan kavram öğrenme, kavramlar ve öğrenme süreci temalarından oluşmuştur (Şekil 6). Kavramlar teması, denel işlem boyunca öğrenme sürecinin parçası olan kavramları, öğrenme süreci teması ise öğrencilerin bu kavramları süreçte nasıl edindiği ve kullandığını kapsamıştır.

4.1.2.1. Kavramlar

Araştırmanın kavramlar teması **hazırlık ve planlama, karar kontrol yapıları, veri yapıları ve dosya işlemleri** alt temalarından oluşmuştur. **Hazırlık ve planlama** alt temasına “algoritma” ve “akış şeması” kodları aracılığıyla ulaşılmış, “algoritma” deney grubu tarafından hazırlık ve planlama sürecinin temel kavramlarından biri olarak tanımlanmıştır. Bu duruma D-Ö4'ün “... algoritmada neyi yapacağın kodlamanın temeli gibi...” ve D-ÖRT'nin “Algoritma, programlamanın temelidir aslında. İşe başlamadan önce yapılması gereken ABC'lerden bir tanesidir.” ifadeleri örnek gösterilebilir. Kontrol grubunda da algoritmanın programlamaya hazırlık sürecinin önemli bir parçası olduğu K-ÖRT'nin “Temelde, problem çözme ve algoritma konusunda iyi olmanız bekleniyor.” ifadesinden çıkarılabilir.



Şekil 6. Kavram öğrenme ile ilişkili tema, alt tema ve kodlar

“Akış şeması” da gruplarda algoritmaya benzer şekilde programlama sürecinin planlamasında kullanılmıştır. D-Ö1’in “Problemi belirledikten sonra çözmemiz gereken aşamaları düşündüm, algoritmasını ve akış şemasını çıkardım, kodları yazdım.” ifadesi deney grubunda akış şemasının kodlama öncesi planlamada kullanıldığını ortaya koymuştur. Kontrol grubunda da akış şemaları, üzerinde düşünülerek planlanması ve sonrasında takip edilmesi gereken bir rehber olarak kabul edilmiştir. K-Ö3’ün “Akış şemasını tamamen düşündükten sonra kod olarak bilgisayara geçirdik.” ve K-Ö1’in “Akış şemasında doğru olup da kodlamaya geldiğimizde hatalı çıkan kısımlar vardı.” ifadeleri akış şemalarının programlamada yol gösterici özelliğini ortaya koymuştur.

Kavramlar temasını oluşturan bir diğer alt tema **karar kontrol yapılarıdır** ve bu alt tema “karar”, “döngü” ve “fonksiyon” kodlarından oluşmuştur. Deney grubunda “karar” kavramına, programlama sürecinde yer alan seçeneklerden birine karar verme eylemi üzerinden değinilmiş, öğrenciler kavramı doğrudan kullanmak yerine kavrama karşılık gelen kodları kullanmayı tercih etmiştir. D-Ö5’in “Mesela yolu ikiye ayırdığında int-input kullanacak değilim, while-true da kullanacak değilim, if kullanmam lazım.” ifadesi buna örnek gösterilebilir. D-ÖRT, karar mekanizmalarını “programlamanın en kilit noktalarından biri” olarak tanımlamış, K-ÖRT de “... (programlama sürecinde) düzgün karar veremezsek bundan sonrası da gelmiyor.” ifadesiyle karar mekanizmalarının belirleyici özelliğine vurgu yapmıştır.

“Döngü”, her iki grupta da programlama öğretim sürecinde sıklıkla kullanılan bir kavramdır. Bu duruma deney grubuna ait “Öğretmen etkileşimli tahta üzerinde döngüye ilişkin olduğunu söylediği kodları ipucu olarak yazdı.” (D-GZ, 9. hafta) ve kontrol grubuna ait “Öğretmen, bu haftanın konusunun while döngüsü olduğunu belirtti.” (K-GZ, 12. hafta) gözlem notları örnek gösterilebilir. Kontrol grubunda K-ÖRT’nin “Özellikle döngülerde zorlanan çocukların sayısı çok oluyor.” ifadesi kavramın kontrol grubunda öğrenilmesi zor bir kavram olarak yer aldığını ortaya koymuştur.

“Fonksiyon” kavramı denel işlem sürecinin bir parçası olmuş, deney grubunda D-ÖRT’nin “... fonksiyonlar ile ilgili olarak bunu anlatabilmek zordur.” ve D-Ö7’nin “Mesela ayının kış uykusuna yatmasında (kış uykusu senaryosunda yer alan metafor örneği) fonksiyonlara biraz biraz giriş yaptık.” ifadeleri doğrudan öğretim sürecinde kavramın kullanıldığına işaret etmiştir. Kontrol grubuna ait “Öğretmen fonksiyon kavramının tanımını yaptı ve kod örnekleri üzerinden açıklamalarda bulundu.” (K-GZ, 13. hafta) gözlem notu da kavramın kontrol grubunda doğrudan kullanıldığına örnek olabilir.

Kavramlar teması altında **veri yapıları** alt temasına “veri” ve “değişken” olmak üzere iki kod aracılığıyla ulaşılmıştır. Deney grubunda gerçekleştirilen gözlemler sırasında öğretmen tarafından öğrencilere yöneltilen “Veri kavramını daha önce duyduk mu?” ve “Programlamada hangi tür veriler kullanılıyor?” (D-GZ, 12. hafta) soruları “veri” kavramının bu grupta öğrenme sürecinin bir parçası olduğunu göstermiştir. Kontrol grubunda da benzer şekilde öğretmenin veri türleri başlıklı

dosyayı etkileşimli tahtada açarak örnekler üzerinden “Sizce bu hangi tür veri?” (K-GZ, 2. hafta) diye sorması veri kavramının bu grupta da kullanıldığına işaret edebilir.

“Değişken” kavramının programlama sürecinde nasıl yer aldığına ise deney grubunda öğretmenin “Malzemelerde geçen ‘isteğe göre değişebilir’ ifadesi neye benzetilebilir?” sorusuna D-Ö1’in “değişken” yanıtını vermesi örnek gösterilebilir (D-GZ, 1. hafta). Kontrol grubu gözlem notlarında yer alan “Öğretmen değişkenin tanımını yaptı.” (K-GZ, 2. hafta) ve “Öğretmen öğrencilerin kullandıkları değişken isimleri konusunda düzeltme ve önerilerde bulundu.” (K-GZ, 5. hafta) biçimindeki tespitler, değişken kavramının bu grupta da ele alındığını gösterebilir.

Dosya işlemleri alt temasına ise “dosya” ve “kayıt” kodları aracılığıyla ulaşılmıştır. Dosya kavramının deney grubunda programlama sürecinde nasıl yer aldığına D-Ö5’in “Dosya işlemleri biraz ağır geldi. Evde biraz daha üzerine çalışarak yaptım.” ifadesi ve “Öğretmen, belgeyi Python ya da programlamada neye benzetebiliriz sorusuna D-Ö15, dosya diyerek yanıt verdi” (D-GZ, 15. hafta) gözlem notu örnek gösterilebilir. Kontrol grubunda dosya kavramının öğretim sürecinde kullanımına yönelik K-Ö1’in “Dosya işlemlerinde bazen hata vermese de boşluk kullanımından dolayı biz kendimiz bu hatayı bulmayı seçiyoruz.” ifadesi örnek verilebilir.

Dosya işlemlerinde dosya içerisinde düzenli bir şekilde yer alan veri setlerine “kayıt” denmektedir. Denel işlem sürecinde “kayıt” kavramına hem deney hem kontrol grubu yer vermiş, kavramı D-ÖRT “Dosya oluşturma işlemi kayıtlar için yeni bir sayfa açmak demektir.” ifadesiyle metaforik olarak tanımlarken, K-ÖRT “Kod örneklerini yansıtarak dosyaya kayıt işlemleri hakkında açıklama yaptı.” (K-GZ, 15. hafta) gözlem notunda olduğu gibi örneklerle açıklamıştır.

4.1.2.2. Öğrenme Süreci

Programlama kavramlarını öğrenme süreci **öğretim ve kavram kazanma** alt temaları ile açıklanmıştır. Öğretim alt teması kavramların öğretiminde kullanılan iki farklı yöntemi, kavram kazanma teması da öğrencilerin bu kavramları tanımlama, kullanma ve ayırt etme biçimlerini kapsamıştır.

Öğretim alt teması “metaforik” ve “sunuma dayalı” olmak üzere iki koddan oluşmuştur. Araştırmada programlamaya yönelik kavramların öğretiminde deney

grubunda metafora dayalı yöntem, kontrol grubunda ise sunuş yoluyla öğretim stratejisi kullanılmıştır.

Deney grubunda ortaya çıkan “metaforik” koduna “kaynak/hedef alan”, “yapısal ilişki” ve “metafor geliştirme” alt kodları aracılığıyla ulaşılmış, deney grubunda kavram öğretiminde pastacılık, doğa, depo ve gündelik yaşam araçları gibi birçok farklı “kaynak” alanı kullanılmıştır. Örneğin, “hedef” alanda tanımlanan değişken ve algoritma kavramları pastacılık “kaynak” alanı aracılığıyla anlamlandırılmaya çalışılmıştır. D-ÖRT’nin “Sizce ben neden pastacılıkla ilgili videoyu izletmiş olabilirim?” ve D-Ö9’un “Pastacılık vardı, algoritma kavramını öğrenmeme yardımcı oldu.” ifadeleri pastacılığın “kaynak” alanı olarak kavram öğretiminde kullanıldığına örnek gösterilebilir.

Metaforik yöntemde kaynak ve hedef alanlarını tanımlamanın yanında bu alanlar arasındaki “yapısal ilişkinin” de anlaşılmasının önemi ortaya çıkmıştır. D-ÖRT’nin kavram öğrenme sürecinde sorduğu “Pastacılık neden bu derste anlatıldı, programcılıkla ilgisi nedir?” ve “Altı çizili olan ya da italik yazılan (kaynak alana ait) kavramlar acaba neyi ifade ediyor bilgisayarda?” (D-GZ, 1. hafta) soruları öğrencileri ilişkiler kurmaya yönelttiğini göstermiştir. Deney grubunda öğretmenin hedef ve kaynak alan arasındaki yapısal ilişkiyi ortaya çıkarmaya çalıştığına “Öğretmen asansör hatası ve test edilmesi ile kod hatası ve test edilmesi arasındaki ilişkiyi sordu.” (D-GZ, 3. hafta) ve “... Öğretmenin ‘ürünlerin saklandığı depolar deyince ne anlıyorsunuz?’ sorusuna D-Ö14, ‘çok değer alan veri yapıları olabilir’ yanıtını verdi.” (D-GZ, 13. hafta) gözlem notları örnek gösterilebilir.

Araştırmada deney grubu öğrencilerinin ve öğretmenin ders planlarında yer alan metaforlarla sınırlı kalmadıkları, aynı zamanda kendi “metaforlarını geliştirdikleri” de görülmüştür. D-Ö5, “Önce bir veri var, veriyi öğrenmem lazım. Yapboz vardır, o yapbozu yapabilmek için dört tarafındaki köşeleri takman lazım.” ifadesinde veriyi yapboz ile, D-ÖRT de fonksiyon kavramını “Fonksiyonlar fabrikalardır.” ifadesinde fonksiyonları fabrika ile ilişkilendirmiştir.

Kontrol grubunda ortaya çıkan “sunuma dayalı” koduna ise “açıklama” ve “örnekleme” alt kodları aracılığıyla ulaşılmıştır. Kontrol grubu öğretmenin programlamaya yönelik kavramların öğretiminde öncelikle kavramın ne anlama geldiğini ve çalışma mantığını “açıklamayı” tercih ettiği görülmüştür. K-Ö3’ün

“(Öğretmen) kavram öğretirken önce ne anlama geldiğini söylüyor.” ve K-ÖRT’nin “Koşullu yürütmeye yeni başladık, öncelikle koşullu yürütmenin mantığını anlatıyorum.” ifadeleri buna örnek gösterilebilir.

Kavramları açıklamanın yanında kontrol grubu öğretmeni kavram öğrenmeyi sağlamak için “örnekleme” yoluna gitmiştir. K-ÖRT’nin “Kavram konu ve süreçlerin öğretimi açısından düz anlatım ve örneklendirme kullanıyorum.” ve K-Ö1’in “Genellikle hocamız önce anlattı sonrasında örneklendirdi.” ifadeleri bu bulguyu desteklemektedir. “Öğretmen, etkileşimli tahtada bir parola örneği göstererek akış şeması kavramını anlatıyor.” (K-GZ, 6. hafta) gözlem notu da kavram öğretiminde örneklerden yararlandığını ortaya koymuştur.

Kavram kazanma, kavram öğrenmede daha çok planlı bir öğretime işaret eder (Ülgen, 2004). Araştırmada kavram kazanma alt teması “kavramı ifade etme”, “kavramı tanımlama” ve “ayırt etme” kodlarından oluşmuştur.

“Kavramı ifade etmeye” yönelik kanıtlara veri toplama sürecinde kavramın doğrudan kullanılması durumunda ulaşılmış, öğrenciler denel işlem boyunca programlamaya yönelik algoritma, akış şeması, döngü, fonksiyon, veri, değişken, dosya ve kayıt kavramlarını doğrudan kullanmışlardır. Örneğin, deney grubundan D-Ö8 süreçte öğrendiği kavramlara ilişkin açıklama yaparken “...fonksiyonları, dosya açıp kapatma işlemlerini, hangi dosyaya yazılır, hangi dosya kapanır öğrendik.” ifadesinde fonksiyon ve dosya kavramlarını kullandığını ifade etmiştir. Kontrol grubundan K-Ö1 de “Akış şemasını yazarken kodları bilmemize gerek yok, döngü kullanırken örneğin sadece oklarla da gösterebiliriz. Döngü komutlarını kullanarak yapabiliriz anladığımız fonksiyonlar vasıtasıyla.” ifadesinde akış şeması, döngü ve fonksiyon kavramlarını birlikte kullandığına değinmiştir. Gözlemler sırasında da araştırmacı zaman zaman öğrencilerin bu kavramları ifade ettiklerini tespit etmiştir. Buna “D-Ö14, geçen haftaki pastaya (Pastadan programa isimli senaryoda yer alan metafor örneği) göre değişken daha az dedi.” (D-GZ, 2. hafta) gözlem notundaki değişken kavramı örnek gösterilebilir.

“Kavramı tanımlama”da öğrenci ve öğretmenlerin kavrama yönelik anlamını söyleme, görevini açıklama, mantığını aktarma ve terim karşılığını bulma gibi yollara başvurdukları görülmüştür. Kavramların tanımlanmasında kontrol grubu öğretmeni kavramın anlamını ve görevini açıklamaya giderken, deney grubu öğretmeni kaynak

ve hedef alan arasındaki ilişkileri ön plana çıkarmıştır. D-ÖRT'nin "Etkinliklerimizi işledikten sonra bilgisayarla olan bağlantılarını kendileri keşfetsin diye zorladım." ifadesi deney grubu öğretmeninin öğrencilerin ilişkiler kurmaları için ısrar ettiğini ortaya koymuştur.

Kavram kazanma alt temasını ortaya çıkaran kodlardan bir diğeri "ayırt etme"dir. Denel işlem sürecinde deney ve kontrol grubunda bulunan öğrenciler birbirinden farklı kavramları ayırt edebilmiştir. D-Ö7'nin, "Bilgisayar kodlama kavramlarının çoğunu biliyorum artık, sonuçta kullanıyoruz. Integer, string, fonksiyonlar mesela." ifadesi deney grubunda kavramların ve bu kavramlara karşılık gelen kodların ayırt edildiğini göstermiştir. Benzer şekilde araştırmacının "Öğretmen öğrencilere metinde geçen malzemeler ve araçlar arasındaki farkı sordu. D-Ö17, 'malzemeler içine konuyor, araçlar ise yapım aşamasında kullanılıyor. Malzemeler bir defa, araçlar sürekli kullanılıyor' diye yanıtladı." (D-GZ, 1. hafta) gözlem notu deney grubunda kavramları ayırt etmede kaynak alandan örneklerin kullanıldığını gösterebilir. Kavramları ayırt etme kontrol grubunda, deney grubuyla benzer şekilde genellikle kod karşılıklarını ayırt etme şeklinde gerçekleşmiştir. K-Ö8'in "Mesela ilk ünite de str, int vardı. Int sayılar için, str karakterler için." ifadesi ulaşılan bulgunun kanıtı olabilir.

4.1.3. Programlama Kavramlarını Öğrenme Sürecine Ait Nicel ve Nitel

Kanıtların İlişkisi

Programlama kavramlarına yönelik nicel bulgular ile nitel bulgulardan **kavramlar** temasını oluşturan alt temaların (hazırlık ve planlama, karar kontrol yapıları, veri yapıları ve dosya işlemleri) örtüştüğü görülmüştür. Nicel bulgular, programlamanın temelleri dışındaki programlamaya yönelik kavramları öğrenmede deney grubu lehine anlamlı fark olduğunu ortaya koymuştur. Programlamanın temellerinde gruplar arasında anlamlı bir farkın çıkmaması, nitel açıdan deney ve kontrol grubu öğretmenlerinin her ikisinin de bu kavramları temel kavramlar olarak tanımlamasından kaynaklanabilir. Program kontrolüne yönelik deney grubu lehine çıkan anlamlı fark, deney grubunda karar kontrol yapılarının kilit nokta olarak tanımlanması, kontrol grubunda ise bu yapıların oldukça zor kavramlar olarak kabul edilmesiyle açıklanabilir. Veri yapıları ve dosya işlemlerine yönelik nitel bulgular, kavramların her iki grupta da daha çok kod örnekleri üzerinden incelenmesi ile ortaya

çıkması, ancak ulaşılan nitel bulgular deney grubu lehine oluşan farkı açıklamada yetersiz kalmıştır. Bu nedenle öğrenme sürecine yönelik nitel kanıtlarla birlikte incelenmesi uygun görülmüştür.

Bu bağlamda, programlamanın temelleri dışındaki diğer programlama kavramlarını öğrenmede deney grubu lehine anlamlı çıkan fark **öğrenme süreci** teması açısından incelendiğinde, deney grubunda uygulanan metafora dayalı kavram öğretiminin program kontrolü, veri yapıları ve dosya işlemlerine yönelik kavramları öğrenmede daha etkili olmasıyla açıklanabilir. Nitel bulgular, metafora dayalı öğretimle kaynak/hedef alan ve bunlar arasındaki ilişkileri kullanarak kavramları öğretmenin kontrol grubunda belirlenen bir kavramı doğrudan açıklama ve örneklemeye oranla kavram öğrenmeye daha fazla katkı sağladığını göstermiş, ayrıca metafora dayalı öğretim, deney grubunun kendi metaforlarını üretmesini teşvik etmiştir. Kavram kazanma sürecini oluşturan kavramı ifade etme, kavramı tanımlama ve kavramları ayırt etme açısından her iki grupta da benzer örneklerle rastlanması; program kontrolü, veri yapıları ve dosya işlemlerinde görülen anlamlı farkın deney grubunda kullanılan metafora dayalı öğretimden kaynaklandığını desteklemiştir.

4.2. Programlama Becerileri

Araştırmanın ikinci alt problemi “metin tabanlı programlama öğretiminde metafor kullanımının, programlama becerisine etkisi nedir?” şeklinde tanımlanmış; ilgili alt problemi yanıtlamak için a) Deney ve kontrol gruplarındaki öğrencilerin programlama becerileri ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır? b) Programlama becerilerini edinme süreci ve bu becerilerin değişimini ortaya koyan nitel kanıtlar nelerdir? c) Programlama becerilerine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır? biçiminde üç ayrı soru oluşturulmuştur. Deney ve kontrol grubu öğrencilerinden denel işlem sonunda elektronik ürün geliştirmeleri beklenmiş, öğrencilerden EÜGS puanlama yönergesi aracılığıyla algoritma ve akış şeması oluşturma, veri girişi ve çıkışı, karar yapıları, tekrarlı yapılar, fonksiyonlar, veri yapıları, dosya işlemleri ve proje geliştirme alt performans görevlerine yönelik veriler toplanmış, elde edilen puanların karşılaştırılmasına aşağıda yer verilmiştir.

4.2.1. Deney ve Kontrol Grubu Karşılaştırmaları

Öğrenciler elektronik ürünlerini bir yönerge rehberliğinde hazırlamış, yönergenin ilk adımını algoritma ve akış şeması oluşturmuştur. Grupların algoritma ve akış şeması alt performans görevine ait puanları normallik varsayımını karşılayamadığından veriler üzerinde Mann-Whitney U testi analizi gerçekleştirilmiştir (Tablo 16).

Tablo 16. Deney ve Kontrol Grubu Algoritma ve Akış Şeması Alt Performans Görevi Puanlarının Mann-Whitney U testi Sonuçları

Grup	n	Sıra Ortalaması	Sıra Toplamı	U
Deney	66	100.70	6646.00	119.00*
Kontrol	69	36.72	2534.00	

* $p < .001$

Tablo 16'ya göre metafora dayalı öğretimin uygulandığı deney grubunda algoritma ve akış şeması alt performans görevi açısından kontrol grubuna göre anlamlı bir fark olduğu bulunmuştur [$U = 119.00$, $p < .001$, $r = .83$]. Etki değeri olarak hesaplanan r , 0.5'ten büyük olduğu için büyük (large) etki olarak yorumlanmıştır. Sıra ortalamaları dikkate alındığında metafora dayalı öğretimin gerçekleştiği deney grubunda kontrol grubuna göre algoritma ve akış şeması oluşturma alt performans görevi puanlarının daha yüksek olduğu görülebilir. Bu bulgu metafora dayalı öğretimin algoritma ve akış şeması geliştirmede etkili olduğunu göstermektedir.

EÜGS puanlama yönergesinde yer alan diğer yedi alt performans görevine (veri girişi ve çıkışı, karar yapıları, tekrarlı yapılar, fonksiyonlar, veri yapıları, dosya işlemleri ve proje geliştirme) ait veriler deney ve kontrol grupları açısından normallik varsayımını karşıladığından grupların bu performans alt görevleri ortalama puanları arasında anlamlı bir farkın olup olmadığını belirlemek amacıyla her alt performans görevine yönelik bağımsız gruplar için t testi analizi gerçekleştirilmiştir. Grupların programlama becerilerine yönelik t testi sonuçlarına Tablo 17'de yer verilmiştir.

Tablo 17. *Programlama Becerilerine Yönelik Deney ve Kontrol Grubu Puanlarının t Testi Sonuçları*

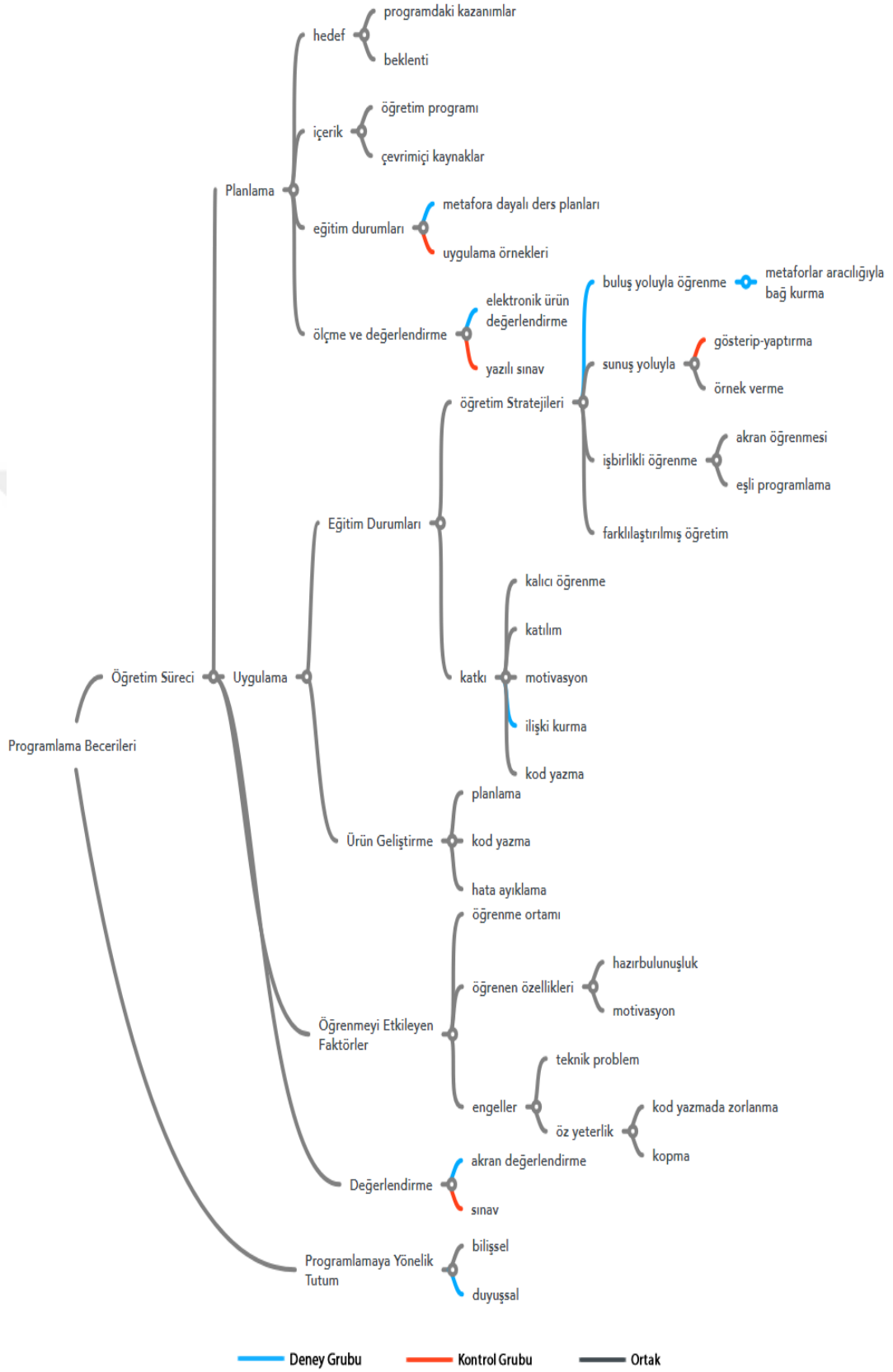
Alt Performans Görevleri	Grup	n	$\bar{X}$	S	sd	t	Cohen's d
Veri Girişi ve Çıkışı	Deney	66	3.10	.43	83.61	4.62*	.78
	Kontrol	69	2.34	1.28			
Karar yapıları	Deney	66	3.76	.36	78.04	7.31*	1.23
	Kontrol	69	2.50	1.38			
Tekrarlı yapılar	Deney	66	3.76	0.38	79.64	13.56*	2.29
	Kontrol	69	1.47	1.34			
Fonksiyonlar	Deney	66	3.83	.26	72.90	15.10*	2.54
	Kontrol	69	1.14	1.45			
Veri yapıları	Deney	66	3.88	.20	71.29	9.81*	1.65
	Kontrol	69	2.26	1.35			
Dosya işlemleri	Deney	66	2.45	.72	108.87	7.90*	1.34
	Kontrol	69	1.05	1.27			
Proje geliştirme	Deney	66	3.71	.47	89.20	10.38*	1.75
	Kontrol	69	2.08	1.21			

* $p < .001$

Tablo 17'ye göre EÜGS puanlama yönergesinde yer alan tüm alt performans görevleri ortalama puanları kontrol grubuna göre daha yüksektir ve grup ortalamaları arasında istatistiksel olarak anlamlı fark vardır. Gruplar arasındaki bu farklar etki büyüklüğü için yapılan analizde veri giriş çıkışı alt performans görevi için orta (medium), diğer alt performans görevleri için büyük (large) etki olarak yorumlanmıştır.

4.2.2. Programlama Becerilerine Yönelik Nitel Bulgular

Şekil 7'de görüldüğü gibi programlama becerisi, Öğretim Süreci ve Programlamaya Yönelik Tutum olmak üzere iki tema altında incelenmiştir. Öğretim Süreci teması **planlama, uygulama, değerlendirme ve öğrenmeyi etkileyen faktörler** olmak üzere dört alt temadan; Programlamaya Yönelik Tutum teması ise “bilişsel” ve “duyuşsal” kodlarından oluşmuştur.



Şekil 7. Programlama becerileri ile ilişkilendirilen tema, alt tema ve kodlar

4.2.2.1. Öğretim Süreci

Planlamada, deney ve kontrol grubu öğretmenlerinin “hedef”, “içerik”, “eğitim durumları” ve “ölçme-değerlendirme” bileşenlerine yer verdikleri belirlenmiştir. Her iki grup da bilgisayar bilimi (Kur-1) dersi öğretim programı programlama ünitesi için belirlenen “kazanımları” temel almış, yıllık planlarında bu kazanımlara yer vermiştir. Ayrıca, denel işlem öncesinde her iki grubun öğretmeni öğrencilerden programlamaya yönelik “beklentilerini” ders saatinin yetersiz olması ve öğretim programı kazanımlarının bilişsel olarak alt düzeylerde tanımlanması nedeniyle düşük düzeyde tutmuştur. D-ÖRT’nin “Hepsinden ileri düzeyde bir kodlama becerisi beklemiyorum.” ve K-ÖRT’nin “...onlardan çok iyi programcı olmalarını beklemiyoruz, bu kadar ders saatinde” açıklamaları her iki grupta da öğretmen beklentilerinin düşük olduğunu gösterebilir. Deney grubu öğretmeni öğrencilerden “kendi ihtiyaçlarını görebilecek programları yapmalarını” bekleyerek daha çok programlama becerilerine vurgu yaparken kontrol grubu öğretmeni “algoritmik düşünme, bilgi işlemsel düşünme, problem çözme becerileri gelişsin” ifadesiyle daha geniş kapsamlı becerilere işaret etmiştir. Öğretmenler içeriği planlarken beklentileriyle paralel olarak temel düzeyde programlama bilgisini içeren konulara yer vermiş, her iki grubun öğretmeni dersin resmi öğretim programını temele almış, dersin basılı bir kitabı olmadığı için “internet siteleri, forumlar” gibi farklı “çevrimiçi kaynaklardan” yararlanmışlardır.

Hedef ve içeriğin planlanmasında dersin öğretim programının temele alınması açısından ortaklaşan deney ve kontrol grubu, eğitim durumlarında deney grubunda metafora dayalı öğretimin uygulanması ve kontrol grubunda öğretmen tarafından daha önceki yıllarda da kullanılan uygulama örneklerinin ele alınması açılarından farklılaşmıştır. K-ÖRT’nin “...yıllardır aynı dersi verdiğim için uygulama örneklerimi her yıl tekrar kullanıyorum” ifadesi uygulama örneklerinde sık değişiklik yapmadığını göstermiştir.

“Ölçme-değerlendirmenin” planlanmasında ise deney grubu öğretmeni önceden hazırlamış olduğu değerlendirme ölçeğini öğrencilerle paylaştığını ve neye kaç puan verdiğini belirterek “elektronik ürünleri” ölçüte dayalı değerlendirdiğini, kontrol grubu öğretmeni ise “Öğrencilerin ne yapıp yapamadıklarını zaten sınavda görüyoruz.” ifadesiyle “yazılı sınav” aracılığıyla öğrenme düzeylerini belirlediğini açıklamıştır.

Programlama öğretim sürecinin uygulama alt teması **eğitim durumları** ve **ürün geliştirmeden** oluşmuştur. Eğitim durumlarında uygulamaya yönelik öğretim stratejilerine ve eğitim durumlarının katkılarına, ürün geliştirmede ise ürün geliştirme işlem basamaklarının öğretimine ilişkin bulgulara yer verilmiştir.

Denel işlem sırasında deney ve kontrol grubu öğretmenlerinin programlama öğretiminde “buluş yoluyla”, “sunuş yoluyla”, “işbirlikli” ve “farklılaştırılmış öğretim” olmak üzere birden fazla öğretim stratejisini kullandıkları görülmüştür.

Deney grubunda ortaya çıkan **buluş yoluyla öğretime** “metafor aracılığıyla bağ kurma” alt koduyla ulaşılmıştır. “Metafor aracılığıyla bağ kurmada” öğrenciler öğrenmeleri gereken bilgi, beceri ya da kavramı iki farklı alan arasındaki ilişkiler aracılığıyla kazanmışlardır. D-ÖRT’nin “Gerçek hayattaki olayları, hikâyeyi (senaryoyu) takip ederken gerçek hayatta bu iken, programcılıkta da bu anlamında kullanılabilirler.” ifadesine göre deney grubu öğrencileri hem günlük hayattan hem de senaryolar aracılığıyla programlama alanına ilişkin bağlantılar kurmuşlardır.

Sunuş yoluyla öğretime ise “gösterip-yaptırma” ve “örnek verme” alt kodlarıyla ulaşılmıştır. Kontrol grubunda denel işlem sürecinde öğrenciler birçok kez öğretmen tarafından yazılan kodları kendi bilgisayarlarına aktarmışlardır. K-ÖRT’nin, “Uygulamada daha önceden kendi yapmış olduğum etkinlikleri kullandım, ‘gösterip-yaptırmayı’ kullandım.” ve K-Ö8’in “(Öğretmen) bize yapmamız gereken şeyi gösteriyordu, sonrasında o örneği bizden kodlar halinde yazmamızı istiyordu.” ifadesi, öğretmenin önceden elinde bulunan kod örneklerini etkileşimli tahtada tekrar yazarak öğrencilere kendi bilgisayarlarına aktarmalarını beklediğini ortaya koymuştur.

Kontrol grubu öğretmeni gösterip-yaptırma tekniğinde öğrencilerden kendi yazdığı kodu bilgisayarlarına aktarmalarını beklerken “örnek verme” ile gösterdiği koda yakın kodları yazmalarını beklemiştir. K-Ö3, bu durumu “(öğretmen) tahtada bir örnek gösteriyordu. Bize bir başlık verip o konuda hakkında bir uygulama yapmamızı istiyordu.” ifadesiyle özetlemiştir. Benzer şekilde D-ÖRT “örnek benzer bir uygulama vererek ders planlarındaki etkinliklerle birleştirmeyi” tercih ettiğini ifade etmiştir. Bu durum denel işlem sırasında her iki grupta da kod örneklerinin öğrencilerle paylaşıldığını ve öğrencilerden benzer başka uygulamalar beklediğini ortaya koymuştur.

Denel işlemin gerçekleştiği bilişim teknolojileri sınıfları, genellikle bir bilgisayarı birden fazla öğrenci kullandığından **işbirlikli öğrenmeyi** temel alan “akran öğrenmesi” ve “eşli programlamaya” oldukça uygundur. Deney ve kontrol grubunda bilişim teknolojileri sınıflarında öğrenciler kodlama yaparken bir sorunla karşılaştıklarında sorunu akranlarıyla tartışarak çözmüş, çözüm bulamamaları halinde öğretmenden yardım istemişlerdir. D-Ö3’ün “Önce arkadaşlarımızdan yardım almaya çalıştık. Onlar da yapamıyorsa öğretmenimize başvurduk.” ve K-Ö5’in “...bizden daha iyi yapanlar vardı. Başta onlardan baktık...” ifadeleri örnek olarak gösterilebilir.

Deney ve kontrol gruplarında öğrenciler “eşli programlama” gerçekleştirmişlerdir. Deney grubunda öğretmen “(Kod) yazma işini yapan öğrencilerin değişmesi” konusunda zaman zaman uyarılarda bulunmuştur. Kontrol grubunda ise K-Ö7, kontrol grubunun genel görüşünün aksine “bilgisayarları tek kişi kullanmanın” başarıyı arttıracığını düşündüğünü ifade etmiştir.

Deney ve kontrol grubunda öğretmenler tam olarak *farklılaştırılmış öğretim* stratejine dayalı öğretim teknikleri kullanmamakla birlikte farklı ilgi alanlarına ve hazır bulunuşluk düzeyine sahip öğrencilere yönelik bazı önlemler almışlardır. K-ÖRT’nin “Uygulamalarda genelde çok uzman programlar vermiyorum. Alt seviyede, alt seviyeleri düşünerek uygulamaları veriyorum.” ve D-ÖRT’nin “Bütün sınıf tamam olduğunda bir sonraki probleme geçiyoruz.” açıklamaları sınıfta farklı seviyedeki öğrencileri dikkate aldıklarını göstermiştir.

Eğitim durumlarıyla ilişkili nitel veriler, çoğunlukla eğitim durumlarının programlama öğretimine **katkısı** üzerinde yoğunlaşmış, eğitim durumlarının “kalıcı öğrenmeye”, “katılıma”, “motivasyona”, “ilişkiler kurmaya” ve “kod yazmaya” olan katkıları incelenmiştir.

Deney grubunda, “kalıcı öğrenme” genellikle derste uygulanan etkinliklerle ilişkilendirilmiştir. D-Ö1’in “Bunlar (etkinlikler) aslında daha çok aklımda kalmasını sağladı. Projeyi hazırlarken aklıma sürekli hikâyeler (senaryolar) geldi.” ve D-Ö2’nin “Hikâyeleştiren daha eğlenceli, daha kolay oluyor aklımda tutmak...” ifadeleri derste kullanılan metafora dayalı senaryoların kalıcılığa etkisini ifade edebilir. Kontrol grubunda ise kalıcılığa yönelik tek açıklama K-Ö1’den gelmiş ve kalıcılığa olan katkının nedeni “(öğretmenin) kodu açıklaması” olarak tanımlanmıştır.

Öğrencilerin derse yönelik “katılımları” ise deney grubunda derste gerçekleştirilen etkinliklerin öğrencilerin role bürünmelerini kolaylaştırmasına, dersi daha eğlenceli hale getirmesine ve meraklarına katkıda bulunmasına, kontrol grubunda ise başarıma duygusuna ve öğretmen rehberliğine bağlanmıştır. Deney grubunda öğrenciler metafora dayalı senaryolar aracılığıyla kendilerini “Hikâyenin içindeymiş gibi” [D-Ö1] hissettiklerini ve etkinliklerin role bürünmelerini kolaylaştırdığını ifade etmişlerdir. K-Ö3’ün “öğretmenimin güzel ilgisinden” ve K-Ö1’in “Çünkü insan bir şeyi başardığı zaman daha fazla şey yapmak istiyor.” ifadelerinden hareketle kontrol grubunda katılımı öğretmenin ilgisi ve başarı güdüsünün etkilediği söylenebilir.

“Motivasyon” açısından deney grubunda uygulanan denel işlemin “dersi ilgi çekici” [D-Ö1] yapması ve “daha eğlenceli” [D-Ö4] hale getirmesi öğrencilerin başarabildiklerini görmelerine ve programlamaya yönelik motivasyonlarının artmasına neden olmuştur. Bunda D-Ö8’in belirttiği gibi “Öykülerin (senaryoların) bu konudaki payının büyük” olduğu ortaya çıkmıştır. Kontrol grubunda ise öğrencilerin dersi sıkıcı bulmaları programlamaya yönelik güdülenmelerini zorlaştırmıştır. Bu duruma K-Ö5’in “(Ders) bana biraz sıkıcı geliyordu, anlamıyordum. Severe başlama şansım olmadı.” ifadesi örnek gösterilebilir.

Araştırmanın deney grubunda, öğrencilerin programlama kavramlarını kaynak alanlarla ilişkilendirmeleri amaçlanmıştır. Metafora dayalı öğretim programlama sürecinde öğrencilere yardımcı olmuş, metafora dayalı ders planlarında uygulamaya dayalı “ilişkilendirmelere” yer verilmiştir. D-Ö8’in “Siz (araştırmacı) soru sorarken aklıma etkinlikler hemen geliyor. Mesela depo etkinliği” ifadesi öğrencinin programlamada yer alan veri yapıları hedef alanı ile lojistik kaynak alanını etkinlik aracılığıyla ilişkilendirdiğini göstermiştir.

Deney ve kontrol grubunda gerçekleştirilen eğitim durumlarının “kod yazmaya” katkısı açısından incelendiğinde D-ÖRT’nin açıkladığı gibi deney grubunda “gerçekçi bir uygulama üzerinde bu işin nasıl yapıldığını” öğrencilerin görmeleri sağlanarak kod yazmaları kolaylaşmıştır:

“Kış uykusu ile ilgili olan etkinliğimizde yine döngüleri anlatırken çok güzel bir benzeşim yapıldığı için, burada biz zorlanıyoruz döngüleri anlatırken, bu etkinlikteki olay neredeyse birebir programcılıkta olduğu için çok daha iyi anlaşıldığını ve döngülerin çok daha net bir şekilde yapıldığını gördüm.” (D-ÖRT).

Kontrol grubunda ise öğretmen geçmiş deneyimlerine dayanarak öğrencilere kod görevleri vermiş ancak onların “özellikle koşullu durumlar ve döngüler için yarıdan fazlasının anlamadığı” [K-ÖRT] ve kodlamada zorlandıkları ortaya çıkmıştır. K-Ö5’in “Ben de herkes gibi bilgisayarda bir şey yapmaya çalışıyordum, yapamıyordum, çok anlamıyordum.” ifadesi, öğrencilerin anlayarak kod yazmak için çaba harcadıkları halde zorlandıklarını gösterebilir.

Ürün geliştirme, “planlama”, “kod yazma” ve “hata ayıklama” kodlarından oluşmuştur. Öğrenciler, programlama aracılığıyla bir ürün geliştirirken “planlamada” algoritma ve akış şemalarından yararlanmışlardır. Deney grubunda problemin çözümünde öğrenciler “Problemi belirledikten sonra çözmemiz gereken aşamaları düşündüm, algoritmasını ve akış şemasını çıkardım, kodları yazdım.” [D-Ö1] ifadesinde olduğu gibi algoritma ve akış şemalarını kodlama aşamasından önce planlamada kullanmışlardır. Kontrol grubunda ise algoritma ve akış şemasını elektronik ürün geliştirme sürecinin bir parçası olarak doğru yerde kullanan öğrenciler olmakla birlikte, K-Ö3’ün “İlk başta bir kod yazıldı. Sonradan akış şemasını yaptık.” ve K-Ö1’in “Yazdık kodları, hep beraber çözdük. Sonra akış şemasını yaptık.” ifadeleri algoritma ve akış şemasının yapılması gereken zorunlu bir basamak olarak görüldüğünü ve amacı dışında kullanan öğrenciler olduğunu ortaya koymuştur.

Öğrenciler, elektronik ürünlerini geliştirme süreçlerinde planlama aşaması sonrası “kod yazmaya” başlamışlardır. Deney ve kontrol grubu öğrencileri kod yazmayla ilk kez metin tabanlı programlama ile başladıkları için zorlanmış, bu durum kontrol grubunda denel işlem süresi boyunca sürmüş, deney grubunda “kavramların öğretildikten sonra bilgisayara aktarılmasında yani yazım diline geçiş (syntax) kısmı” [D-ÖRT] ile sınırlı kalmıştır.

Öğrenciler etkinliklerde yer alan problemleri çözerken ya da kendilerinden beklenen elektronik ürünü geliştirirken de sıklıkla test etmiş ve “hata ayıklamışlardır”. Öğrenciler bu test etme sürecinde karşılaştıkları hataları çözmek için grup arkadaşlarından, akranlarından ve öğretmenlerinden yardım almışlardır. D-Ö7’nin “Bazı zor etkinliklerde yapamadım. Öğretmenimiz yardım etti.” ve K-Ö2’nin “Yapamazsam arkadaşlarımdan yardım istiyorum. Onlar da yapamaz ise hoca'dan” ifadeleri grupların hata ayıklamada arkadaşlarından ve öğretmenlerinden yardım istediklerini ortaya koymuştur.

Araştırmada **değerlendirme** alt temasına “akran değerlendirme” ve “yazılı sınav” kodlarıyla ulaşılmıştır. Deney grubunda yer alan öğrenciler hemen hemen her ekinlikte “sınıfta akranlarının elektronik ürünlerini iki adet elektronik form” [D-ÖRT] aracılığıyla değerlendirmiş, kontrol grubunda ise öğrenciler bir planlamaya dahil olmaksızın nadiren “diğer arkadaşlarının görevlerini sınıf içerisinde dolaşarak” [K-ÖRT] incelemişlerdir.

Kontrol grubunda sadece yazılı “sınav” yapılırken deney grubunda daha çok performansa dayalı değerlendirmeye gidilmiştir. K-ÖRT, “Bütün yaptığım sınavlarda genelde çoktan seçmeli boşluk doldurmalı.” sorular sorduğunu ifade etmiştir.

Araştırmada **öğrenmeyi etkileyen faktörler** alt temasına “öğrenme ortamı”, “öğrenen özellikleri” ve “engeller” kodları aracılığıyla ulaşılmıştır. Denel işlem sırasında deney grubunda olumlu bir “öğrenme ortamı” oluşmuş, öğrenciler bu durumu dersin öğretmenine ve metafora dayalı etkinliklere bağlamışlardır. D-Ö5’in “Derslerde baya güldük. Hocamız da baya güldüren bir insan. Yani güle güle güzel bir şekilde yaptık. Yoksa bu etkinlikler olmasa çok gülebileceğimiz bir şey yok zaten.” ifadesi, öğretmen özelliklerinin ve etkinliklerin öğrenme ortamını olumlu etkilediğini gösterebilir. Kontrol grubunda ise öğrenme ortamının daha çok fiziksel özelliklerine vurgu yapılmış, öğrenciler sayılarının bilişim teknolojileri sınıfı için fazla olmasından, derse yönelik yeterince materyalin sınıfta bulunmamasından ve bilgisayarların eski olmasından yakınmışlardır. K-Ö3’ün “Sınıfta daha az kişi olsa daha iyi olabilirdi.” ve K-Ö7’nin “... Zaten farklı bir şey kullanmıyoruz bir tek akıllı tahta var.” ifadeleri kontrol grubu öğrenme ortamının fiziksel özelliklerine ilişkin görüşleri örnekleyebilir.

“Öğrenen özellikleri” açısından ise öğrencilerin “hazır bulunuşluk” düzeyleri ve “öğrenme motivasyonları” denel işlem sırasında programlama öğrenme sürecini etkilemiştir. Bilgisayar bilimi (Kur-1) dersi öncesinde deney ve kontrol grubu öğrencilerinden hiçbiri metin tabanlı bir programlama eğitimi almamış, 5. ve 6. sınıfta bilişim teknolojileri ve yazılım dersinde aracılığıyla blok tabanlı kodlama gerçekleştirmişlerdir. D-Ö4’ün “Ortaokulun sonunda almıştım ama böyle kod yazarak değil. Orada daha çok Scratch gibi uygulamalarla daha çok hazır olan kodları yerlerine yerleştirme yaptık” ifadesi bu açıklamayı destekleyebilir.

Deney ve kontrol grubunda yüksek ve düşük motivasyonlu öğrenciler bir arada yer almıştır. Deney grubundan D-Ö4, “(Mobil uygulama) yapması çok zor bir şey

değilse ben kendim de böyle bir şey yapmak isterim.” diyerek ilgisini ortaya koyarken D-Ö2, “...yapmışlar işte, hiç kafa yormadım daha önce” ifadesiyle uygulama geliştirme sürecini merak etmediğini aktarmıştır. Kontrol grubunda da benzer şekilde K-Ö2 “Programlamaya yönelik merakım çok.” derken K-Ö1 “(program geliştirme süreçlerini) Genellikle çok merak etmem.” diyerek farklı motivasyonlara sahip öğrencilerin kontrol grubunda da yer aldığını göstermiştir.

Denel işlem süresince öğrenmeyi etkileyen engeller “teknik problemler” ve “öz yeterlik” olarak ortaya çıkmıştır. Deney ve kontrol grubunda benzer teknik problemler oluşmuş, problem kimi zaman “bir öğrenci bilgisayarında açılmama sorunu” [D-GZ, 2. hafta] kimi zaman da “öğretmenin başka bir sınıfın etkileşimli tahtasında yaşanan teknik bir arızadan kaynaklı sınıftan ayrılması ve dersin bölünmesi.” [K-GZ, 7. hafta] olarak gerçekleşmiştir.

“Öz yeterlik” ise bazı öğrencilerin “kod yazmada zorlanmaları” ve dersten “kopma” yaşamaları ile ortaya çıkmıştır. Her iki grup öğrencileri de sürecin başında kod yazmada zorlanmışlardır. D-Ö7’nin “İlk başta korktum yapamayacağız diye. İlk başta biraz zor geldi” ve K-Ö7’nin “Dediğim gibi başlarda pek yapabildiğimi sanmıyorum.” ifadeleri öğrencilerin programlama öz yeterlikleri ile ilişkili bulunmuştur. Sonrasında deney grubundaki öğrenciler metafora dayalı etkinlikler sayesinde daha kolay ve sıkılmadan kod yazmaya başlamış, kontrol grubunda yer alan öğrenciler ise zamana bağlı olarak oluşan başarıma hissiyle bu zorluğu yenmişlerdir. D-Ö2’nin “Problemleri etkinlikler aracılığıyla günlük hayatla bağdaştırdığımız için etkinlikler problemi kolay çözmeme yardımcı oluyor.” ve K-Ö5’in “Başta nasıl yapacağını araştırmak zor oluyor. Sonrasında bitince gurur duyuyorsun kendinle, başardım diyorsun.” ifadeleri bu bulguyu desteklemektedir.

Deney ve kontrol grubunda öğrencilerden bazılarında konular ilerledikçe dersten kopmalar yaşanmış, deney grubunda derste gerçekleştirilen etkinlikler bu öğrencilerin tekrar ders sürecine dâhil olmalarına yardımcı olmuştur. D-Ö8’in “(Etkinlikler) dersten kopmamı önledi ilk önce.” ifadesi denel işlemin dersten kopmayı önlemeye katkısına örnek verilebilir. K-Ö2’nin “Her şey birbirine bağlı olduğu için hepsini yakalayamadın mı hepsi bozuluyor.” ifadesi de kontrol grubunda konular ilerledikçe yapamayacaklarına olan inancın arttığını ve dersten kopmaların yaşandığını gösterebilir.

4.2.2.2. Programlamaya Yönelik Tutum

Programlamaya yönelik tutum “bilişsel” ve “duyuşsal” kodlarından oluşmuştur. Öğrencilerin programlamaya yönelik öğrenmeleri arttıkça programlamaya dönük tutumları bilişsel bileşeni değişim göstermiştir. D-Ö2’nin “Önceden değil ama şimdi öğrendiğimiz için merak ediyorum acaba hangi kodu yazmışlar falan diye” ifadesi ve K-Ö1’nin “Genellikle kimin geliştirdiğini merak etmiyorum ama nasıl geliştirildiğini ve teknikleri merak ediyorum.” ifadeleri denel işlem sonrası deney ve kontrol grubundaki öğrencilerin başkaları tarafından geliştirilen uygulamalardaki kodlamaları merak ettiklerini göstermiştir.

Denel işlem gruplarında tutumu oluşturan “duyuşsal” bileşene yönelik değişim deney grubunda ortaya çıkmış, deney grubundaki duyuşsal değişime metafora dayalı ders planlarında yer alan etkinlikler neden olmuştur. D-Ö9’un “...ilk ünitelerde çok hevesimiz yoktu ama zaman geçtikçe bir şeyler yapmaya başladıkça bence daha güzeldi.” ifadesi öğrencideki zamanla oluşan duyuşsal değişime örnek verilebilir. D-ÖRT’nin “Programcılığa isteği artırma açısından, olayın nasıl döndüğünü anlatma açısından, günlük hayattaki karşılığı açısından ve de çocuklardan aldığım karşılığı gördüğümde, mutluluğu gördüğümde bunun aslında çok iyi bir başlangıç olduğunu düşündüm.” ifadesi deney grubunda gerçekleştirilen etkinliklerin duyuşsal bilene olumlu katkısını örnekleyebilir.

4.2.3. Programlama Becerisi Açısından Nitel ve Nicel Kanıtların İlişkisi

Araştırmanın nicel bulguları elektronik ürün geliştirmenin algoritma ve akış şeması oluşturma, veri girişi ve çıkışı, karar yapıları, tekrarlı yapılar, fonksiyonlar, veri yapıları, dosya işlemleri ve proje geliştirme boyutlarında deney grubu lehine istatistiksel anlamlı bir fark olduğunu ortaya koymuştur. Bu bulgular, nitel bulgularda ortaya çıkan **öğretim süreci** ve **programlamaya yönelik tutum** temaları aracılığıyla açıklanmıştır.

Araştırmada, programlama becerisine yönelik nicel veriler deney ve kontrol grubu öğrencilerinin elektronik ürünlerinin değerlendirilmesinde kullanılan EÜGS puanlama yönergesi, nitel veriler ise denel işlem öncesi, sırası ve sonrası sürecini ele alan görüşme, gözlem ve değerlendirme formları aracılığıyla toplanmıştır. Nitel veri toplama sürecinin kapsamının oldukça geniş olması ve elektronik ürün geliştirme sürecini de içine alması öğretim süreci temasında ortaya çıkan **ürün geliştirme** alt

temasının EÜGS puanlama yönergesinde yer alan tüm alt performans göstergeleri ile örtüşmesini sağlamıştır.

Elektronik ürün geliştirme sürecini “planlamada” deney grubu öğrencileri kontrol grubu öğrencilerine göre daha kapsamlı algoritmalar ve akış şemaları oluşturmuş, nitel bulgulara göre bu durum kontrol grubu öğrencilerinin planlamayı bir gereklilik değil zorunluluk olarak görmelerinden kaynaklanmıştır. “Uygulama” aşamasında ise hem deney hem kontrol grubunda yer alan öğrenciler ilk zamanlarda kodlamada zorlanmış, deney grubu öğrencileri sınıfta gerçekleştirilen etkinlikler aracılığıyla, kontrol grubu öğrencileri de başarıma duygusunun getirdiği motivasyonla bu zorluğu aşmışlardır. “Hata ayıklamaya” yönelik her iki grup da akranları ve öğretmenleri aracılığıyla sorunları tespit etmiş ve çözmüştür. Bu nitel bulgular, grupların ürün geliştirme sürecini planlamada ve uygulamada kullanılan ders planları açılarından farklılaştığını ortaya koymuş ve bu farklılıklar deney grubu lehine oluşan istatistiksel anlamlı farkı açıklamıştır.

Öğretim süreci ana temasını oluşturan bir diğer alt tema **eğitim durumlarıdır**. Nicel bulgular eğitim durumları açısından incelendiğinde, deney grubunda “motivasyon”, “ilişki kurma” ve “kod yazma” açılarından kontrol grubuna oranla daha olumlu özellikler tespit edilmiş, özellikle deney grubunda ortaya çıkan alanlar arası bağ kurmaya ilişkin farklılık uygulanan ders planlarıyla ilişkilendirilmiştir.

Nitel bulgularda öğretim sürecine yönelik ortaya çıkan tema, alt tema ve kodlar ürün geliştirme sürecini doğrudan ya da dolaylı olarak etkilemiştir. **Öğretim stratejileri** bunlardan biridir. Öğretmenler denel işlem sırasında buluş yoluyla, sunuş yoluyla, işbirlikli ve farklılaştırılmış öğretim gibi farklı öğretim stratejilerini bir arada kullanmış, deney grubu öğretimde buluş yoluyla öğretim stratejisini, kontrol grubunda ise sunuş yoluyla öğretim stratejisi kullanmakla birbirlerinden farklılaşmıştır. Denel işlem sırasında deney grubu öğretmeni buluş yoluyla öğrenme stratejisini temele alarak öğrencilerin metafor aracılığıyla bağ kurmalarına, kontrol grubu öğretmeni sunuş yoluyla öğretim stratejisini temele alarak gösterip-yaptırma tekniğiyle uygulama örneklerine odaklanmıştır. Her iki grupta işbirlikli ve farklılaştırılmış öğretim stratejilerinin uygulanması açılarından ortaklaşmıştır. Deney grubunu kontrol grubundan ayıran ve metafora dayalı öğretimin temelini oluşturan alanlar arası metafor

aracılığıyla bağ kurma nicel bulgulardaki deney grubu lehine anlamlı farkı açıklamıştır.

Programlama **öğrenmeyi etkileyen faktörler** açısından deney grubunda öğrenme ortamının daha çok olumlu duyuşsal özellikleri, kontrol grubunda ise fiziksel yetersizlikler öne çıkmıştır. Deney ve kontrol grubu öğrencileri denel işlemin başında “motivasyon” ve “hazır bulunuşluk” açılarından benzer öğrenen özellikleri göstermiş, sonrasında karşılaşılan engeller, deney grubunda metafora dayalı etkinlikler aracılığıyla kolaylıkla aşılabilmektedir. Araştırmada her iki grup da benzer “teknik problemlerle” karşılaşmış, “kod yazmada zorlanma” ve “kopmaları” deney grubu sınıfta gerçekleştirilen etkinlikler aracılığıyla daha kolay atlatmıştır. Öğrenmeyi etkileyen faktörlere ilişkin deney grubunda oluşan olumlu duyuşsal ortam ve etkinlikler kontrol grubundan farklılaşarak deney grubu lehine ortaya çıkan EÜGS puanlama yönergesi alt performans görevleri puanları arasındaki anlamlı farkı açıklamıştır.

Planlanma ve değerlendirme alt temalarına yönelik bulgular, gruplarda yer alan öğrencilerin programlama becerilerindeki değişimle ilişkili yeterli veriler sunmadığından EÜGS puanlama yönergesi alt performans görevleri puanlarının deney grubu lehine anlamlı çıkmasını açıklamada yeterli bulunmamıştır.

Programlama kategorisini oluşturan bir diğer tema olan **programlamaya yönelik tutum** teması ise bilişsel ve duyuşsal kodları aracılığıyla oluşmuştur. Deney ve kontrol grubunun her ikisinde benzer tutum bilişsel bileşen değişiklikleri gerçekleşmiş fakat tutumu oluşturan duyuşsal bileşenler açısından deney grubu, kontrol grubundan olumlu yönde farklılaşmıştır. Deney grubunda bu değişimin temel nedeni denel işleme bağlanmıştır. Deney grubu lehine ortaya çıkan bu olumlu tutum nicel bulgulara ortaya çıkan programlama becerisine yönelik anlamlı farkı açıklamaya yardımcı olmuştur.

4.3. Bilgi İşlemsel Düşünme Öz Yeterliği

Araştırmanın üçüncü alt problemi “Metin tabanlı programlama öğretiminde metafor kullanımının, bilgi işlemsel düşünme becerilerine etkisi nedir?” şeklinde tanımlanmış; ilgili alt problemi yanıtlamak için a) Deney ve kontrol gruplarındaki öğrencilerin bilgi işlemsel düşünme becerisi ön test-son test puan ortalamaları arasında anlamlı bir fark var mıdır? b) Bilgi işlemsel düşünme becerisinin gelişimini ve değişimini ortaya koyan nitel kanıtlar nelerdir? c) Bilgi işlemsel düşünme becerisine ait nitel ile nicel kanıtlar arasında nasıl bir ilişki vardır? biçiminde üç ayrı soru

oluşturulmuştur. Aşağıda, ilgili sorulara yönelik ulaşılan bulgular sırasıyla sunulmuştur.

4.3.1. Deney ve Kontrol Grubu Karşılaştırmaları

BİDÖ ölçeği; “problem çözme”, “işbirlikli öğrenme ve eleştirel düşünme”, “farklı düşünme” ve “algoritmik düşünme” faktörlerinden oluşmaktadır. Deney ve kontrol grubunda yer alan öğrencilerin BİDÖ ölçeği aracılığıyla elde edilen ön test ve son test puanlarının karşılaştırılmasıyla ulaşılan bulgular aşağıda sunulmuştur.

Araştırmada öğrencilerin problem çözme beceri düzeyleri problem çözme faktörü aracılığıyla ölçülmüş, grupların problem çözme faktörü ön test-son test ortalama puanlarına ve standart sapmalarına Tablo 18’de yer verilmiştir.

Tablo 18. Deney ve Kontrol Grubu BİDÖ Ölçeği Problem Çözme Faktörü Ortalama ve Standart Sapma Değerleri

Grup	Ön test			Son test		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	78.74	6.88	66	79.33	8.24
Kontrol	69	78.74	9.52	69	74.32	12.20

Tablo 18 incelendiğinde deney grubu son test ortalama puanının ön test ortalama puanına göre daha yüksek, kontrol grubu son test ortalama puanının ise ön test ortalama puanına göre daha düşük olduğu görülebilir. Deney ve kontrol grubu öğrencilerinde gözlemlenen değişimin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemeye yönelik gerçekleştirilen Karışık Ölçümler için ANOVA sonuçlarına Tablo 19’da yer verilmiştir.

Aşağıda verilen Tablo 19’a göre deney ve kontrol grubunda yer alan öğrencilerin problem çözmeye yönelik faktör puanlarının denel işlem öncesinden sonrasına anlamlı bir farklılık gösterdiği, yani deney ya da kontrol grubunda olmak ile tekrarlı ölçümler faktörlerinin problem çözmeye yönelik faktör puanları üzerindeki ortak etkilerinin anlamlı olduğu bulunmuştur [$F_{(1, 133)} = 6.97, p < .05, \eta_p^2 = .05$]. Kısmi eta kare (η_p^2) değeri .01 ve .06 değerleri arasında yer aldığından orta (medium) etki olarak yorumlanmıştır. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının problem çözmeye yönelik faktör puanlarını arttırmada farklı etkilere sahip olduğunu göstermiştir. Problem çözmeye yönelik faktör puanları denel işlem öncesine göre artış gösteren deney grubunda uygulanan metafora dayalı öğretimin

problem çözmeye yönelik faktör puanları açısından kontrol grubuna göre daha etkili olduğu söylenebilir.

Tablo 19. *BİDÖ Ölçeği Problem Çözme Faktörü Ön test ve Son test Puanları ANOVA Sonuçları*

Varyansın Kaynağı	KT	sd	KO	F	η_p^2
Gruplar arası	16156.86	134			
Grup (Deney/Kontrol)	424.67	1	424.67	3.59	.02
Hata	15732.19	133	118.28		
Gruplar içi	8748.27	135			
Ölçüm (Ön test/Son test)	247.33	1	247.33	4.07*	.03
Grup*Ölçüm	423.55	1	423.55	6.97*	.05
Hata	8077.38	133	60.73		
Toplam	24905.14	269			

* $p < .05$

Araştırmada öğrencilerin eleştirel düşünme ve işbirlikli öğrenme beceri düzeyleri eleştirel düşünme ve işbirlikli öğrenme faktörü aracılığıyla ölçülmüş, grupların ön test-son test ortalama puanlarına ve standart sapmalarına Tablo 20’de yer verilmiştir.

Tablo 20. *Deney ve Kontrol Grubu BİDÖ Ölçeği Eleştirel Düşünme ve İşbirlikli Öğrenme Faktörü Ortalama ve Standart Sapma Değerleri*

Grup	Ön test			Son test		
	n	$\bar{X}$	S	n	$\bar{X}$	S
Deney	66	31.98	5.07	66	29.95	5.56
Kontrol	69	30.99	4.31	69	28.17	4.67

Tablo 20’de grupların son test ortalama puanlarının ön test ortalama puanlarına göre düşük olduğu görülebilir. Deney ve kontrol grubunda gözlemlenen bu değişimin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemek için Karışık Ölçümler için ANOVA gerçekleştirilmiş, deney ve kontrol grubu öğrencilerinin eleştirel düşünme ve işbirlikli öğrenme düzeyleri denel işlem öncesinden sonrasına anlamlı bir farklılık göstermemiştir [$F_{(1,133)} = .57, p = .44$]. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının eleştirel düşünme ve işbirlikli öğrenme düzeylerini etkilemediğini ortaya koyabilir.

Çalışmada öğrencilerin farklı düşünme beceri düzeyleri farklı düşünme faktörü aracılığıyla ölçülmüş, grupların farklı düşünme faktörü ön test-son test ortalama puanlarına ve standart sapmalarına Tablo 21’de yer verilmiştir.

Tablo 21. *Deney ve Kontrol Grubu BİDÖ Farklı Düşünme Faktörü Ortalama ve Standart Sapma Değerleri*

Grup	Ön test			Son test		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	35.79	3.79	66	34.83	4.55
Kontrol	69	35.35	4.31	69	33.67	6.14

Tablo 21’de deney ve kontrol grubu son test ortalama puanlarının ön test ortalama puanlarına göre düşük olduğu görülebilir. Deney ve kontrol grubunda gözlemlenen bu değişimin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemek için Karışık Ölçümler için ANOVA gerçekleştirilmiş, deney ve kontrol grubu öğrencilerinin farklı düşünme düzeyleri denel işlem öncesinden sonrasına anlamlı bir fark göstermemiştir [$F_{(1,133)} = .57, p = .44$]. Bu bulgu, öğrencilerin deney ya da kontrol grubunda olma durumlarının farklı düşünme düzeylerini etkilemediğini ortaya koyabilir.

Çalışmada öğrencilerin algoritmik düşünme beceri düzeyleri algoritmik düşünme faktörü aracılığıyla ölçülmüş, grupların algoritmik düşünme faktörü ön test-son test ortalama puanlarına ve standart sapmalarına Tablo 22’de yer verilmiştir.

Tablo 22. *Deney ve Kontrol Grubu BİDÖ Ölçeği Algoritmik Düşünme Faktörü Ortalama ve Standart Sapma Değerleri*

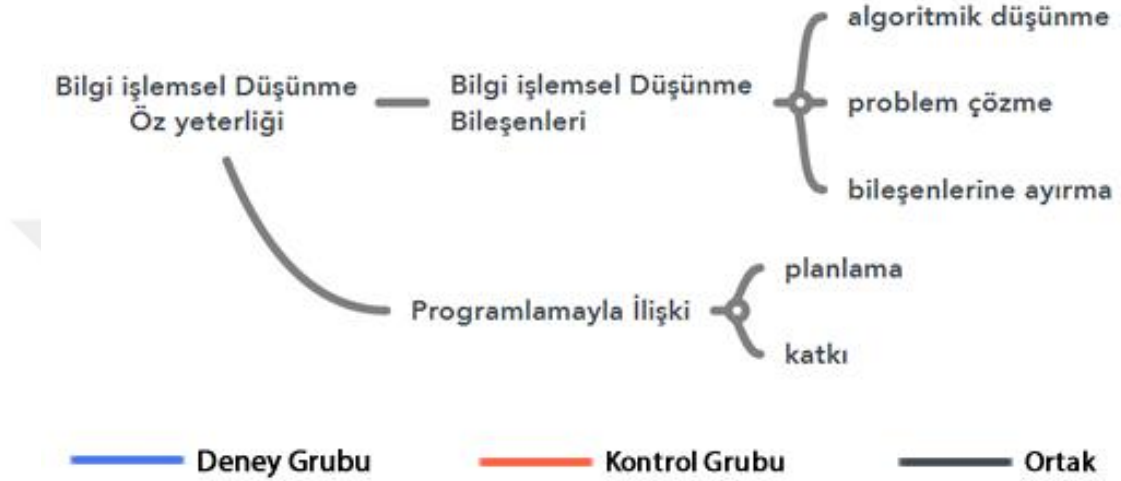
Grup	Ön test			Son test		
	<i>n</i>	$\bar{X}$	<i>S</i>	<i>n</i>	$\bar{X}$	<i>S</i>
Deney	66	19.11	2.02	66	19.35	2.56
Kontrol	69	18.30	2.30	69	18.25	2.83

Tablo 22’de grupların son test puanlarında ön test puanlarına göre artış, kontrol grubunda ise azalış olduğu görülebilir. Deney ve kontrol grubunda gözlemlenen bu değişimlerin gruplar arasında anlamlı bir farklılık gösterip göstermediğini belirlemek için Karışık Ölçümler için ANOVA gerçekleştirilmiş, deney ve kontrol grubu öğrencilerinin farklı düşünme düzeyleri denel işlem öncesinden sonrasına anlamlı bir fark göstermemiştir [$F_{(1,133)} = .25, p = .61$]. Bu bulgu, öğrencilerin deney ya da kontrol

grubunda olma durumlarının algoritmik düşünme düzeylerini etkilemediğini ortaya koyabilir.

4.3.2. Bilgi işlemsel Düşünme Öz Yeterliliğine Yönelik Nitel Bulgular

Araştırmada, bilgi işlemsel düşünme öz yeterliği kategorisi altında bilgi işlemsel düşünme bileşenleri ve programlamayla ilişki olmak üzere iki tema (Şekil 8) yer almış, söz konusu bu temalar aşağıda açıklanmıştır.



Şekil 8. Bilgi işlemsel düşünme öz yeterliği kategorisini oluşturan tema, alt tema ve kodlar

4.3.2.1. Bilgi işlemsel Düşünme Bileşenleri

Bilgi işlemsel düşünme bileşenleri teması, bilgi işlemsel düşünmenin kapsamını oluşturan “algoritmik düşünme”, “problem çözme” ve “bileşenlere ayırma” kodlarından oluşmuştur. “Algoritmik düşünme”, deney ve kontrol grubunda bilgi işlemsel düşünmenin bir ögesi olmakla birlikte programlama sürecinin verimliliğine katkı sağlayan bir düşünme biçimi olarak da ortaya çıkmıştır. Gruplar algoritmik düşünmeyle programlama görevlerini adımlara ayırmış ve çözüm yolları aramışlardır. D-Ö2’nin, “Başta problemi yani sorunu, daha sonra problem konusunda ne yapmamız gerektiğini arkadaşlarla belirledik.” ifadesi deney grubunda bir problemin nasıl ele alındığını özetlemiştir. K-ÖRT’nin, “Soruyu tahtaya yazıyorum daha sonra algoritma ve akış şemasını nasıl çizeceklerini gayet iyi biliyorlar. Algoritmayı yazıyorlar, akış şemasını çıkarıyorlar.” açıklaması da kontrol grubunun problemin çözümüne geçmeden önce algoritma ve akış şeması aracılığıyla problemi analiz ettiğini ortaya koymuştur.

Bilgi işlemsel düşünmenin bir başka bileşeni olan “problem çözmede” ise öğrenciler elektronik ürün geliştirirken problem çözme basamaklarından bazılarını kullanmış, deney grubu kontrol grubuna oranla daha fazla problem çözme basamağını dikkate almıştır. Deney grubunda problem çözme basamaklarından problemi tanımlama basamağına D-Ö5’in “Önce problemin ne olduğunu, nasıl yazabileceğimizi...”, olası çözüm yolları önerme basamağına D-Ö8’in “İşte onu nasıl koda aktarabileceğimizi” ve test etme basamağına D-Ö4’ün “Ondan sonra kodun ilk yapman gereken kısmını yaptıktan sonra hata çıkıyor mu diye kontrol ediyorsunuz.” ifadeleri örnek verilebilir. Deney grubu öğretmeni problem çözme tanımlarken “algoritmayı oluşturup adım adım çözebilme, sonrasında akış şeması oluşturabilme, son aşamada da kod haline getirebilme.” ifadesiyle problem çözme basamaklarına vurgu yapmıştır, kontrol grubu öğretmeni ise “mesela dediğim gibi programın yarısını vereceğim bir kısmını vermeyeceğim bulun çözüm kendiniz, gerekirse internetten” yönergesiyle problem çözmede görevi tamamlamaya odaklanmıştır.

Deney grubunda yer alan öğrenciler denel işlem sürecinde karşılaştıkları problemleri çözerken “bileşenlerine ayırmayı” tercih etmiştir. Öğrenciler ilk olarak problemin çözümü için problemi parçalara bölmeyi tercih etmiş, daha sonra çözmeye çalışmıştır. D-Ö2’nin “Problemi parçalara böldük. Önce ilk sorunu sonra diğerlerini...” ifadesi bu durumu desteklemektedir. D-Ö3’ün “İlk olarak problemi bir bütün halinde çözmeye çalıştık, olmayacağını düşünerek tekrar problemi bileşenlerine ayırmayı tercih ettik.” ifadesi ise deney grubunda sürecin başında problemi bütüncül olarak çözmeye çalışan ve daha sonra parçalara ayırmayı tercih eden öğrenciler olduğunu göstermiştir. Kontrol grubunda ise problemi bütün olarak ele alanların sayısı bileşenlerine ayıranların sayısına oldukça yakındır. Kontrol grubunda problemi bütüncül olarak ele almayı tercih eden öğrencilerden K-Ö1 problemi neden bütüncül olarak ele aldığını, “Bir bütün olarak çözmeye çalıştığımız zaman kafa karışıklığı fazla olmuyor. Ama parça parça ilerlediğimizde bir şey yaptığımız zaman hata verdiğinde bulamıyoruz o hatayı nerede olduğunu.” ifadesiyle açıklamıştır.

4.3.2.2. Programlamayla İlişki

Bilgi işlemsel düşünme öz yeterliği kategorisinin **programlamayla ilişki** teması “planlama” ve “katkı” kodlarından oluşmuştur. Deney ve kontrol grubu öğrencileri elektronik ürünlerini kodlamaya geçmeden önce “planlama” yapmak

amacıyla algoritma ve akış şeması oluşturmuşlardır. Algoritma ve akış şeması geliştirme süreci bilgi işlemsel düşünme bileşenlerinden algoritmik düşünmeyle ilişkili olarak ortaya çıkmış, D-Ö4 bu ilişkiyi “algoritmayla kendi yazdıklarını koda çevirmek daha kolay oluyor.” ifadesiyle özetlemiştir. Ayrıca araştırmanın nitel analizi sırasında programlama ve bilgi işlemsel düşünme öz yeterliği kategorilerinde planlama açısından ortaya çıkan kodlar ortaklık göstermiş, K-ÖRT de “Programlamayla bir problemin çözümünde işin çoğu algoritmada bitiyor.” ifadesiyle planlama aşamasına vurgu yapmıştır.

Denel işlem sürecinde bilgi işlemsel düşünme algoritmik düşünme ve bileşenlere ayırma öğeleri aracılığıyla programlama becerisine *katkı* sağlamıştır. Deney ve kontrol grubunda öğrenciler öncelikle verilen görevlerin planlamasını yapmış, daha sonra anlamlı birimlere bölerek görevi tamamlamayı tercih etmiştir. Algoritmik düşünme; sürecin planlanması ve problemin akılcı bir biçimde çözülmesi için gerekli hazırlıkların yapılmasında, bileşenlere ayırma ise çözümü kolaylaştırma amacıyla hem deney hem de kontrol grubunda kullanılmıştır. D-ÖRT’ye göre, “öğrencilerin önce algoritmayı hazırlayıp, sonra programı yapmaları” da bilgi işlemsel düşünceleri üzerinde olumlu etki yaratmıştır.

4.3.3. Bilgi İşlemsel Düşünme Öz Yeterliği Açısından Nitel ve Nicel

Kanıtların İlişkisi

Araştırmada, BİDÖ ölçeğinde yer alan problem çözme faktörü dışındaki diğer faktörlerde deney ve kontrol gruplarının ortalama puanları açısından istatistiksel anlamlı bir fark bulunmamıştır. Bilgi işlemsel düşünme öz yeterliğine yönelik nitel bulgulara ise bilgi işlemsel düşünme bileşenleri ve programlamayla ilişki olmak üzere iki tema ortaya çıkmış, elde edilen bulgular, problem çözme ve algoritmik düşünme becerisi açılarından örtüşmüştür.

Algoritmik düşünme daha çok programlama sürecinin planlaması ile ilişkilendirilmiş, deney grubunda Pastadan Programa etkinliği ve kontrol grubunda dersler öncesi algoritma ve akış şeması geliştirme araçlarını örneklendirme gibi birçok etkinlik yer almıştır. Her iki grup öğretmeni de algoritma ve akış şemasını programlama sürecinin temel bir bileşeni olarak görmüş, bu bileşen sonraki aşamalar için temel bir gereksinim olarak tanımlanmıştır. Deney ve kontrol grubu öğretmenlerinin her ikisinin de bu aşamaya oldukça önem vermesi, her iki grupta da

algoritmik düşünmeye yönelik birçok etkinliğe yer verilmesi gruplar arasındaki algoritmik düşünme faktör ortalama puanları arasında anlamlı farkın olmamasını açıklayabilir.

Problem çözme ise denel işlem sırasında bilgi işlemsel düşünme ve programlamanın temel süreçlerinden biri olarak ortaya çıkmıştır. Elektronik ürün geliştirme sürecinde deney ve kontrol gruplarının her ikisinde de problem çözme basamakları kullanılmakla birlikte deney grubunda problemi tanımlama, olası çözüm yolları üretme ve çözüm yollarını test etme basamakları daha sistematik bir şekilde yürütülmüştür. Deney grubu öğretmenin hem ders planlarındaki metafora dayalı senaryoları kullanması hem de problem çözme sürecini daha sistematik bir açıdan ele alması gruplar arasındaki BİDÖ problem çözme faktör ortalama puanları arasındaki farkı açıklayabilir.

V. BÖLÜM

SONUÇ, TARTIŞMA VE ÖNERİLER

Bu bölümde, alt problemler çerçevesinde ulaşılan bulgulardan çıkarılan sonuçlar paylaşılmış; alanyazın bağlamında tartışılmış ve değerlendirmeler yapılmıştır. Son olarak, ulaşılan sonuçlardan hareketle uygulamaya ve ileride yapılacak araştırmalara yönelik önerilerde bulunulmuştur.

5.1. Sonuç ve Tartışma

Araştırmada, Lakoff ve Johnson'ın (1980) Kavramsal Metafor Kuramını temele alan ve araştırmacı tarafından geliştirilen metafora dayalı öğretimin öğrencilerin programlama kavramlarını öğrenme, programlama ve bilgi işlemsel düşünme becerilerine etkisi incelenmiştir. Metafora dayalı programlama öğretimi metin tabanlı programlamayla ilk kez karşılaşan öğrencilerin programlama kavramlarını öğrenmelerine ve programlama becerilerine olumlu katkı sağlamış, bilgi işlemsel düşünme becerisinin bileşenlerinden sadece problem çözmede anlamlı bir fark oluşturmuştur. Araştırmada elde edilen bu sonuçları, ayrıntılı olarak metafora dayalı öğretimin *programlama kavramlarını öğrenmeye, programlama ve bilgi işlemsel düşünmeye etkisi* başlıkları altında incelenmiş ve tartışılmıştır.

5.1.1. Metafora Dayalı Öğretimin Programlama Kavramlarını Öğrenmeye Etkisi

Programlama kavramlarını öğrenme süreci, öğretimde ele alınan kavramlar ve bu kavramların nasıl öğretildiğiyle ilişkilidir (Yıldız Durak, 2018). Araştırmada programlama öğretimine yönelik algoritma, akış şeması, döngü, karar, fonksiyon, veri, değişken, dosya ve kayıt gibi kavramlar ele alınmıştır. Bilgisayar bilimi (Kur-1) dersi öğretim programının bu kavramlara yer vermesi, deney ve kontrol grubu öğretmenlerinin bu kavramları programlama süreci için gerekli kavramlar arasında saymaları bu kavramların deney ve kontrol grubunda ortaya çıkmasını sağlamıştır. Alanyazında da programlama öğretiminde bu kavramları inceleyen birçok araştırmaya rastlanabilir (ör. Luxton-Reilly, 2018; Strachey, 2000). Araştırmalardan bazıları kavramlara bütüncül olarak yer verirken (ör. Chibaya, 2019; Ontanon, 2017) birkaç kavramı bir arada inceleyen araştırmalar (ör. Hermans ve diğ., 2018; Perez-Marin ve diğ., 2018) da bulunmaktadır. Deney ve kontrol grubunda ele alınan kavramların alanyazında yer verilen kavramlarla benzerlik göstermesi, söz konusu kavramların öğretimine yönelik bulguların genellenebilirliğini sağlayabilir.

Deney ve kontrol gruplarında programlamaya hazırlık için gerekli kavramlardan elektronik ürünün ortaya çıkmasını sağlayan karmaşık süreçlere kadar kapsamlı bir bağlama yer verilirken planlama sürecine oldukça uzun zaman ayrılmıştır. Grupların planlamaya uzun süre ayırmaları ilgili kavramları öğrenmelerini kolaylaştırmaya, deney ve kontrol grubu arasında programlamanın temel kavramlarını öğrenmede anlamlı bir fark oluşmamasına neden olmuş olabilir. Karar kontrol yapılarına yönelik kavramlar ise nitel bulgulara programlamanın kilit noktası olarak ortaya çıkmış; kavram öğrenme, karar yapıları sonrası zamana bağlı olarak zorlaşmıştır. Nicel bulgular metafora dayalı öğretimin, karar kontrol yapıları, veri yapıları ve dosya işlemlerine yönelik kavramların öğrenilmesinde deney grubu lehine anlamlı farkın oluştuğunu göstermiştir. Programlamanın temellerine yönelik kavramların öğrenilmesinde ise gruplar arasında istatistiksel anlamlı fark ortaya çıkmamıştır. Bu durum dikkatleri nitel bulgulara ortaya çıkan öğrencilerin kavram kazanma becerilerine ve kullanılan öğretim yöntem ve tekniklerine çekmiş, araştırmada elde edilen bulgular aşağıda bu açılardan incelenmiştir.

Kavram kazanma açısından deney ve kontrol grupları kavramları doğrudan ifade etmiş, tanımlarken anlamını söyleme, görevini açıklama, mantığını aktarma ve terim karşılığını bulma gibi yollara başvurmuştur. Kavram kazanmanın ilk aşaması olan ön bilgilere dayalı kazanımlara ise öğrencilerin metin tabanlı programlamayla ilk kez karşılaşmaları nedeniyle ulaşılamamıştır. Ayrıca gruplar, kavramlar yerine sık sık kavrama karşılık gelen kod ifadelerini kullanmıştır. Grupların kavram kazanma becerileri açısından benzer özellikleri taşıması, deney grubu lehine anlamlı farka ilişkin odağı yine kullanılan yöntem ve tekniklere yöneltmiştir.

Araştırmada deney grubunda metafora dayalı öğretim, kontrol grubunda ise sunuş yoluyla öğretim gerçekleşmiştir. Sunuş yoluyla öğretimde kontrol grubu öğretmeni kavramları açıklama ve örnekleme yoluyla kazandırmayı seçerken deney grubunda metafora dayalı öğretimle kaynak ve hedef alan arası yapısal ilişkileri ortaya çıkarmaya odaklı bir kavram öğrenme süreci izlenmiştir. Araştırmada nitel bulgular karar kontrol yapıları, veri yapıları ve dosya işlemleri ile ilişkili kavramların öğretiminde kaynak – hedef alan ilişkilerini temel alan metafora dayalı öğretimin daha etkili olduğunu ortaya koymuştur. Numic (2020), önceki metafor belirleme araştırmalarına dayanarak geliştirilen araçların öğrencilerin kavram kazanmalarında

etkili olduğunu belirtmektedir. Bu araştırmada, öğrencilerin kavram kazanmalarını desteklemek için öncelikle öğretimde metafor kullanan araştırmalar incelenmiş, sonrasında özgün metaforların uygulama sürecinde etkilerini belirlemek için yeni metaforlar geliştirilmiştir. Bu nedenle programlamaya yönelik kavramları kazandırmada önceden belirlenmiş ya da özgün metaforların birlikte ya da ayrı olarak kullanılabileceği düşünülmektedir.

Araştırmada metafora dayalı öğretime uygun olarak hazırlanan ders planlarının temelini oluşturan metafora dayalı öğretim kaynak/hedef alan ve bunlar arasındaki yapısal ilişki ile metafor geliştirme bileşenlerinden oluşmuştur. Metafora dayalı öğretim sürecinde pastacılık, asansör, orman yolu, kış uykusu, ankesörlü telefon, depo ve nüfus kayıt sistemi gibi çeşitli kaynak alanlarından yararlanılmış, bu kaynak alanlar aracılığıyla programlamanın soyut kavramlarının öğrencilere kazandırılması hedeflenmiştir. Soyut kavramların metaforlar aracılığıyla hedef ve kaynak alanlar arası ilişkiler aracılığıyla kazandırılmasına Hermans ve diğerlerinin, (2018) değişken kavramını açıklamada gündelik araçlar kaynak alanından kutu; Perez-Marin ve diğerlerinin (2018) mutfak kaynak alanından yemek tarifi; teknoloji kaynak alanından bilgisayar ekranı, karar verici olarak bilgisayar ve tekrarlayıcı olarak bilgisayar metaforlarını kullanmaları örnek verilebilir. Alanyazında ve araştırmada kullanılan kaynak alanlardan örnekler, programlamanın soyut kavramlarının öğretiminde günlük hayattan çok kapsamlı alanların programlama kavramlarının öğretiminde kullanılabileceğini göstermiştir.

Kaynak ve hedef alan arasında kurulan ilişkinin türü soyut kavramların öğretiminde önemlidir. Alanlar arası ilişkilerin uygun biçimde kurulamaması kavram yanlışlarına neden olabilmektedir (Hermans ve diğ., 2018). Metafora dayalı öğretimde oluşabilecek kavram yanlışlarını önlemek için hedef alanda yer alan kavramı açıklayacak kaynak alanların uygun seçimi yapılmalıdır. Bunun yanında kaynak alan ve hedef alan ilişkilerini doğru tanımlamak ya da öğrencilerden bu ilişkileri betimlemelerini istemek kavram yanlışlarının tespitini de kolaylaştıracaktır. Birçok araştırma (ör. Fanny ve diğ., 2020; Perez-Marin ve diğ., 2018) programlama kavramlarının öğretiminde metaforlar aracılığıyla yapısal ilişkileri incelemiştir. Örneğin, Hermans ve diğerleri (2018) değişken kavramını “kutunun” yapısal özellikleri aracılığıyla anlamlandırmaya çalışmıştır. Alanyazında önceden belirlenmiş

metaforların öğrenme üzerindeki etkilerini inceleyen araştırmalar olduğu gibi, öğretim sırasında ortaya çıkan metaforlar aracılığıyla hedef ve kaynak alan arasında ne tür bir ilişki olduğunu ortaya koyan araştırmalar (ör. Chibaya, 2019; Fanny ve diğ., 2020; Larsson, 2022; Larsson, 2023a; Larsson, 2023b) da yer almaktadır.

Araştırmada ortaya çıkan sonuçlardan bir diğeri, denel işlem sürecinde kullanılan metaforların kavram öğretiminde öğretmenlerin, kavramları anlamlandırmada öğrencilerin kendi metaforlarını geliştirmelerine teşvik etmesidir. Alanyazında, metafora dayalı bir model geliştirilirken (Chibaya, 2019), robotik kavramların öğretiminde (Fanny ve diğ., 2020) ve metaforu temele alan bir araç geliştirmede (Numic, 2020) öğretmen ve öğrencilerin metafor geliştirmeleri için teşvik edildiği örnekler yer almaktadır. Öğrenme ortamında metafor örneklerinin artması öğretmen ve öğrenciyi yeni metaforlar geliştirmeye cesaretlendirmiş olabilir.

Programlamanın soyut kavramlarını özgün yapısal metaforlar aracılığıyla kazandırmayı amaçlayan bu araştırmada, ister özgün ister önceden geliştirilmiş metaforların belirlenmesinde uygun metafor seçiminin önemi ortaya çıkmış, programlama kavramlarını öğretmede yapısal metaforlar dışında nesnelerle deneyime dayanan ontolojik metaforların ya da uzay/mezan algısını temel alan yönelimsel metaforların incelenmesi, öğretmenler ve öğrencilerin kendi metaforlarını geliştirmeleri de programlama kavramlarının etkili öğretimi için gerekli bulunmuştur.

5.1.2. Metafora Dayalı Öğretimin Programlamaya Etkisi

Araştırmanın nitel bulgularını programlama öğretim sürecini oluşturan planlama, uygulama, ölçme ve değerlendirme bileşenleri, öğrencilerin programlama öğrenmelerini etkileyen faktörler ve öğrencilerin programlamaya yönelik tutumları oluşturmuştur.

Deney ve kontrol grubu öğretmenleri hedefleri ve içeriği planlamada öğretim programını temel almalarıyla benzerlik gösterirken eğitim durumları ve ölçme-değerlendirmenin planlanmasında farklılaşmıştır. Öğretimi planlamada hedefler belirlenirken her iki grupta da dersin öğretim programında yer alan kazanımlardan yararlanılmış; öğretmenler, öğrencilerden beklentilerini dersin seçmeli bir ders olması nedeniyle temel düzeyde tutmuştur. Benzer şekilde dersin içeriğine karar verilirken her iki grupta da öğretim programı temele alınmış, ek olarak elektronik kaynaklardan yararlanılmıştır. Denel işlem gereği, deney grubunda gerçekleştirilen eğitim durumları

kontrol grubundan farklılaşmıştır. Değerlendirmenin planlanmasında ise deney grubu elektronik ürünü değerlendirmeyi, kontrol grubu ise yazılı sınavı tercih etmiştir. Deney ve kontrol grubunun süreçte eğitim durumları açısından farklılaşması nicel bulgularda ortaya çıkan deney grubu lehine istatistiksel anlamlı farkı açıklayabilir.

Planlama sürecinden sonra programlama öğretim sürecini oluşturan bir diğer öge uygulamadır. Nitel bulgularda öğretim stratejileri uygulamanın önemli bir bileşeni olarak yerini almıştır. Programlama öğretimi sırasında gruplarda buluş yoluyla, sunuş yoluyla, işbirlikli öğrenme ve farklılaştırılmış öğretim gibi stratejiler kullanılmıştır. Her iki grup öğretmeni öğrencilerin hazır bulunuşluklarını, ilgilerini, yeteneklerini dikkate almakla birlikte bunu planlı bir biçimde gerçekleştirmemiştir. Deney grubu buluş yoluyla öğretim stratejisini temele alarak kaynak ve hedef alanlar arası metafor aracılığıyla bağ kurmayla, sunuş yoluyla öğretim stratejisi kapsamında gösterip yaptırmayı temel alan kontrol grubundan farklılaşmıştır. Deney grubunda kullanılan metafora dayalı öğretim, öğrencilerin hedef ve kaynak alan aracılığıyla ilişkiler kurmalarını kolaylaştırmış, bu durum öğrencilerin programlama becerilerine olumlu yansımıştır.

Deney grubunda uygulanan metafora dayalı öğretim motivasyon ve kod yazmayı kolaylaştırmada kontrol grubuna oranla daha çok katkıda bulunmuştur. Ürün geliştirmede her iki grup da planlama, kod yazma ve hata ayıklama süreçlerinden geçmiş; deney grubunda hedef ve kaynak alan arası kurulan ilişkiler öğrencilerin kod yazmalarına olumlu katkı sağlamıştır. Sanford ve diğerleri (2014) metafor türlerini kataloglamayla, Schetz-Sobrinio ve diğerleri (2019) de yollar ve trafik işaretleri metaforuna dayanan ANGELA isimli bir öğretim aracıyla kod yazmaya metaforların olumlu etkisini ortaya koymuştur. Farida ve diğerleri (2022) ile Hui ve Umar'a (2011) göre metaforik düşünme, programlama derslerini öğrenmeye ve kod yazmaya katkı sağlamaktadır. Deney grubunda kullanılan metaforların kod yazmaya olumlu katkısı, öğrencilerin karmaşık ve anlaşılması zor soyut kavramlara karşılık gelen kodların daha iyi anlamlandırılmasıyla açıklanabilir. Ayrıca Hidalgo- Cespedes ve diğerleri (2018) metaforların kod yazmaya etkisini kalıplaşmış metaforlar olarak tanımlanan alegorilerle karşılaştırmış, alegoriler ve metaforlar arasında bir fark görülmemiştir. Bu durum, kalıplaşmış metaforların kod yazmada özgün metaforlar gibi katkı sağlayabildiğini gösterebilir.

Araştırmada deney ve kontrol grubu öğretmenleri, öğrencilerden beklentilerini programlama öğrenmede zorlanmaları nedeniyle temel düzeyde tutmuştur. Bunun nedeni alanyazında belirtildiği gibi (ör. Bennedsen ve Caspersen, 2019; Santos ve diğ., 2020) programlama öğrenmenin zorluğu olabilir. Programlamanın soyut ve anlaşılması zor kavram ve süreçleri barındırması, öğrencilerin programlamada zorlanmalarına ve dersten erken kopmalarına neden olmaktadır (Major ve diğ., 2011; Papadakis ve Kalogiannakis, 2017). Deney ve kontrol grubu öğrencilerinin zaman zaman metin tabanlı programlamada zorlanmaları kontrol grubunda sürece yayılırken deney grubunda denel işlem aracılığıyla zamanla aşılmıştır. Karar kontrol yapıları öğrencilerin kod yazmada zorlanmalarında kilit noktalardan birini oluşturmuş, öğrenciler programlamada en çok karar kontrol yapılarından sonra zorlanmaya başlamıştır. Corno ve diğerleri (2019) programlamaya yeni başlayanların kodlamada karşılaştıkları If-Then yapısına ait komut hatalarını ayıklamada kullanabilecekleri yapboz metaforunu temel alan görsel bir araç; Esper ve diğerleri (2013) “sihirbazlık” metaforunu temel alan bir Java eklentisi aracılığıyla metaforların etkinliklerini değerlendirdikleri çalışmalarında metaforların metin tabanlı programlamada karşılaşılan zorlukları yenmede etkili olabileceğini göstermiştir. Araştırmanın deney grubunda programlamada zorlanmanın ve dersten erken kopmaların önüne geçilmesi metafora dayalı öğretim etkinliklerinin dersi daha eğlenceli ve daha anlaşılır hale getirmesi ile açıklanabilir.

Araştırma sürecinde programlamaya yönelik öğrenmeler arttıkça deney ve kontrol grubunun her ikisinde de programlamaya yönelik tutumun bilişsel bileşeninde olumlu değişim gözlenmiş, tutumun duyuşsal bileşeninde öğrencilerin programlama merakları artmıştır. Deney grubunda metafora dayalı öğretim, öğrencilerin programlama isteklerini artırmıştır. Dolayısıyla metafora dayalı öğretim, programlamaya yönelik tutuma olumlu katkı sağlamıştır. Araştırmada metin tabanlı programlamayla ilk kez karşılaşan öğrenciler öncelikle kodlarını metin biçiminde yazabilecekleri editörlerle karşılaşmış, editörde nasıl program yazacakları, çalıştıracakları ve hataları nasıl düzeltecekleri konusunda zorlanmıştır. Barmpoutis ve Huynh’a (2019) göre metafora dayalı gerçekleştirilen metin tabanlı programlama öğretimi sözdizimini kolaylaştırmaktadır. Metin tabanlı programlama ile ilk kez

karşılaşan öğrencilerin metin tabanlı programlamaya keskin geçişini metafora dayalı öğretim günlük hayattan daha tanıdık kaynak alanlar aracılığıyla kolaylaştırmıştır.

Denel işlem boyunca deney grubunda uygulanan metafora dayalı öğretim sınıfta olumlu yansımış, bu durum öğrencilerin kod yazmalarını kolaylaştırmış ve dersten kopmalarının önüne geçmiştir. Sınıfta olumlu ortamın oluşmasında öğretmenin kişilik özelliklerinin katkıları gündeme gelebilir. Ancak, metafora dayalı ders planlarının öğretmeni öğretim süreçlerinde aktif kıldığı da dikkate alınmalıdır. Grupların benzer hazır bulunuşluğa ve motivasyona sahip olmaları ve sınıflarda benzer teknik problemlerle karşılaşmaları deney grubunda programlama becerisi lehine çıkan anlamlı farkı metafora dayalı öğretim ile ilişkilendirilmesini sağlamıştır. Metafora dayalı öğretim; programlama becerilerine kod yazmada, metin tabanlı programlamaya geçişte ve dersten erken kopmaların önüne geçmede sağladığı katkılar nedeniyle etkili zihinsel bir araç olarak tanımlanabilir.

5.1.3. Metafora Dayalı Öğretimin Bilgi İşlemsel Düşünmeye Etkisi

Araştırmada gruplar bilgi işlemsel düşünme bileşenlerinden problem çözmede deney grubu lehine anlamlı biçimde farklılaşmıştır. Bilgi işlemsel düşünmenin diğer bileşenlerinden eleştirel düşünme ve işbirlikli öğrenme, farklı düşünme ve algoritmik düşünme açılarından ise gruplar arasında istatistiksel olarak anlamlı bir fark oluşmamıştır.

Problem çözüme, bilgi işlemsel düşünmeyi oluşturan önemli faktörlerden biridir (Maharani ve diğ., 2019) ve metafora dayalı öğretim, problem çözme becerisini geliştirmede etkili olmuştur. Deney grubu problem çözerken daha fazla problem çözme basamağını dikkate almış, metafora dayalı etkinlikler kod yazma ve programlamaya yönelik tutumlarında değişim yaratarak problem çözmelerini kolaylaştırmıştır. Pérez-Marín ve diğerleri (2020) de metafora dayalı öğretimin öğrencilerin bilgi işlemsel düşüncelerine olumlu katkı sağladığını ve probleme odaklanmayı kolaylaştırdığını belirtmektedir. Metafora dayalı öğretimin kavram öğrenme ve programlama üzerindeki olumlu etkisi; karşılaşılan problemleri tanımlamada, farklı çözüm yollarını belirlemede, çözümü uygulamada ve sonuçlarını değerlendirmede katkı sağlamış olabilir.

Araştırmada, bilgi işlemsel düşünmenin diğer bileşenlerinde gruplar arasında anlamlı farkın oluşmama nedenlerine ise algoritmik düşünme ve bileşenlerine

ayırmaya yönelik veriler üzerinden ulaşılmıştır. Programlamanın önemli aşamalarından biri olan algoritma da algoritmik düşünme ile yakın ilişkilidir ve algoritmik düşünmenin uygulamadaki izlerindendir. Algoritma, programlama öğretiminde ilk ele alınan konulardan biridir (Susanti ve diğ., 2021) ve araştırmada öğretmenler, algoritmalara oldukça uzun zaman ayırdıkları için her iki grupta da öğrenciler algoritma hazırlamada yeterlik kazanmıştır. Her iki grubun da algoritma ve akış şeması geliştirme çalışmalarıyla programlamayı planlamaları algoritmik düşünmeye katkı sağlamış, gruplar elektronik ürün geliştirme sürecinde görevlerini adımlara ayırmıştır. Öğrencilerin algoritma geliştirme üzerinde uzun süre çalışması ve her iki grup öğretmenin algoritma geliştirmeye önem vermesi öğrencilerin algoritmik düşünme becerilerinde gelişim sağlamakla birlikte gruplar arasında istatistiksel açıdan anlamlı fark oluşmamıştır.

Bileşenlerine ayırma, bilgi işlemsel düşünmeyi diğer becerilerden ayıran önemli bileşenlerden biridir (Rich ve diğ., 2018). Gruplar arasındaki problemi bileşenlerine ayırmaya yönelik farklılık ise bilgi işlemsel düşünme puanları üzerinde etki yaratmamıştır. Deney grubu öğrencileri denel işlemin başında verilen problemlerin çözümüne bütüncül olarak başlamış fakat zamanla problemi bileşenlerine ayırmayı tercih etmiştir. Kontrol grubu öğrencilerinin neredeyse yarısı, zihinsel karışıklık yaşamamak için problemi bütüncül olarak çözmüştür. Selby ve Woollard'a (2014) göre büyük problemler, karmaşık sistemler ya da görevlerle uğraşırken bileşenlerine ayırma gereklidir. Bir problemi bileşenlerine ayırmayı, bir görevi adımlara ayırmaktan farklı kılan şey ise problemlerin ayrıştırılmış parçalarının daha sonra adımlara ayrılabilmesi ile açıklanabilir (Rich ve diğ., 2018). Yani bileşenlerine ayırmada adımlardan daha büyük görevlerden bahsedilmektedir. Deney grubunda yer alan öğrencilerin tamamının verilen problemi çözmede problemi bileşenlerine ayırması; bilgi işlemsel düşünme bileşenlerinden problem çözmede deney grubunun kontrol grubuna göre anlamlı farkı açıklayabilir.

Metafora dayalı programlama öğretimi öğrencilerin eleştirel düşünme ve işbirlikli öğrenmelerinde deney grubu lehine anlamlı bir fark yaratmamıştır. Alanyazında sıklıkla bilgi işlemsel düşünme ile ilişkilendirilen eleştirel düşünmenin temelini bir problemi daha derin bir seviyede düşünmek ve mevcut bilgi ve becerileri kullanarak ya da uyarlayarak problemi değerlendirmek oluşturur (Doleck ve diğ.,

2017). Sternberg'e (1986) göre eleştirel düşünme insanların sorunlarını çözmek, karar vermek ve yeni kavramları öğrenmek için kullandıkları zihinsel süreçleri, stratejileri ve temsilleri içerir. Tsirkunova'ya (2013) göre kavramsal metaforlar, eleştirel düşünme becerilerinin öğretiminde etkili bir araç olarak tanımlanmakla birlikte bu araştırma sonuçları deney ve kontrol grubu öğrencilerinin eleştirel düşünme becerilerinde metafora dayalı öğretimin anlamlı bir fark oluşturmadığını ortaya koymuştur. Bu durum her iki grupta da verilen programlama görevlerinin programlama öğrenmeyi sadeleştirmek için derin düşünmeye sevk edecek durumların öğrencilere sunulmamış olması ve öğretmenlerin öğrencilerden derse yönelik beklentilerinin temel düzeyde tutulmasıyla açıklanabilir.

İşbirlikli öğrenme, bilgi işlemsel düşünmeyi oluşturan bileşenleri doğrudan etkilemektedir ve bu nedenle önemlidir. Programlama, bilgi işlemsel düşünmeyi geliştiren etkili bir yoldur ve öğretiminde Lye ve Koh'a (2014) göre işbirlikli, eşli ve bireysel programlama gibi çeşitli yaklaşımlar kullanılmaktadır. Araştırmada denel işlem sürecinde verilen elektronik ürünün kodlanmasında eşli programlama kullanılmış, problemlere işbirlikli öğrenme aracılığıyla çözümler aranmıştır. Her iki grupta da eşli programlama ve işbirlikli öğrenmenin birlikte kullanımı ve nitel bulgularda metafora dayalı öğretimin işbirlikli öğrenmede anlamlı fark yaratacak kanıtlar içermemesi, deney grubu lehine anlamlı farkın oluşmamasını açıklayabilir.

5.2. Öneriler

Araştırmadan elde edilen sonuçlara dayalı öneriler; “programlama öğretimine” ve “ileride yapılacak araştırmalara” yönelik olmak üzere iki başlıkta sunulmuştur:

5.2.1. Programlama Öğretimine Yönelik Öneriler

- Metafora dayalı öğretim uygun kaynak – hedef alan seçimi yapılmış, alanlar arası ilişkileri doğru şekilde tanımlanmış metaforlar aracılığıyla planlı ve kapsamlı programlar geliştirmede alternatif bir öğretim tekniği olarak kullanılabilir.

- Metafora dayalı öğretim, kavram öğretimini daha etkili hale getirdiği, dersleri daha ilgi çekici kıldığı, akılda kalıcılığı ve derse olan katılımı arttırdığı için öğrencilerin karar kontrol yapıları ile ilgili ve sonrası kavramları öğrenmede yaşadıkları zorlukları aşmalarında ve dersten erken kopmalarının önüne geçmede kullanılabilir.

- Araştırmada, metafora dayalı ders planlarının hazırlanmasında pastacılık, makine, doğal yaşam, teknolojik aletler, depo ve kamu kuruluşu gibi birçok farklı kaynak alandan yararlanmıştır. Kavram kazandırmada kullanılabilecek metaforlar bu kaynak alanlar ile sınırlı olmadığından kavram öğretimi için uygun farklı kaynak alanları belirlenebilir, öğretim programları uygun kaynak alan seçimi ve alanlar arası ilişkilendirmede öğretmenlere rehberlik edebilir.

- Kavram yanlışları, kavram öğretiminde karşılaşılan yaygın problemlerden biridir. Öğretmenler, kavram yanlışlarını önlemede ve gerektiğinde oluşan kavram yanlışlarını ortadan kaldırmada uygun yöntem ve teknikleri belirlemek ve öğrencilerin kavramı nasıl anlamlandırdıklarını ortaya koymak için birtakım becerilere sahip olmalıdır. Bu nedenle kavram yanlışları, olası nedenleri ve kavram yanlışlarını önlemede metafora dayalı öğretimin katkıları mesleki gelişim programlarının bir parçası olabilir.

- Bilgiyi yapılandırmada öğrencinin aktif olarak yer alması öğrenmeyi kolaylaştırabilir ve kalıcılığı arttırabilir. Bu nedenle öğrencilerin kavram öğrenme süreçlerinde kendi yaşantılarına dayanan metaforlar geliştirmeleri kavram öğrenmelerini destekleyebilir. Öğrencilerin kendi metaforlarını geliştirmelerini destekleyecek unsurlardan biri de öğretmenlerin sınıfta kendi metaforlarını geliştirmede istekli olmalarıdır. Bu nedenle öğretim programlarında öğrencilerin ve öğretmenlerin kendi metaforlarını geliştirmelerini destekleyecek açıklamalara yer verilebilir.

- Metafora dayalı öğretim, kavram öğretiminin yanında bir elektronik ürünün planlanmasında, geliştirilmesinde ve değerlendirilmesinde kullanılabilir. Metaforların kullanımı, kavram öğretiminin ötesine geçerek proje tabanlı öğrenme ve işbirlikli öğrenmeyle birlikte programlama ve bilgi işlemsel düşünme becerileri uygulamalarına genişletilerek geniş kapsamlı katkı sağlayabilir. Kavram öğretiminde kullanılan metafor örnekleri senaryolar aracılığıyla bir programlama görevine dahil edilebilir, uygulamada kullanımı kavram öğrenimini destekleyebilir.

- Araştırmada öğretmenler, öğrencilerden beklentilerini kavram ve söz dizimi öğretiminde ve programlama için gerekli üst düzey becerileri kazandırmada yaşanan sorunlar nedeniyle temel düzeyde tutmuştur. Metafora dayalı öğretimin bu problemlerin çözümüne yönelik katkıları düşünüldüğünde, öğrencilerden beklentiler

daha üst düzeylere çıkarılabilir, öğretim programı kazanımlarının belirlenmesinde bu durum dikkate alınabilir.

- Programlamayla ilk kez karşılaşan ya da metin tabanlı programlamaya geçişte zorluk yaşayan öğrenciler metin tabanlı programlamaya geçişte zorlanabilmektedir. Metafora dayalı öğretim, öğrencilerin kaynak alana ait deneyimlerini hedef alana aktarmalarını kolaylaştırarak metin tabanlı programlamaya geçişlerini kolaylaştırmada kullanılabilir.

- Araştırma sonuçları öğretmenlerin kavram öğretiminden çok programlama dilinin söz dizimine odaklandıklarını, öğrencilerin de kavramlar yerine program kodlarını ifade etmeyi tercih ettiklerini göstermiştir. Bu durum öğrenci öğrenmelerinin programlama dili ile sınırlı kalmasına yol açabilmektedir. Öğretim programlarında, öğrencilerin beceri temelli bir öğrenme gerçekleştirebilmeleri için kavram öğrenmenin öneminden bahsedilebilir ve öğretmenlerin mesleki gelişim çalışmalarında bu durum dikkate alınabilir.

- Araştırma, deney ve kontrol grubunda derslerin zaman zaman başka öğretmenlerin teknik destek ihtiyaçları nedeniyle bölündüğünü, her iki grup öğretmenin bilişim teknolojileri rehber öğretmeni görevleri nedeniyle diğer ders öğretmenlerine yardımcı olmak için sınıfı terk ettiklerini ortaya koymuştur. Bu durum öğrencilerin dikkatinin dağılmasına neden olmaktadır ve yasal düzenlemelerle öğretmenlerden ders süreleri boyunca yardım talep edilmesinin önüne geçebilir.

5.2.2. İleride Yapılacak Araştırmalar için Öneriler

- Araştırmada metafor kullanımının programlama kavramlarını öğrenmeye, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliğine etkisi anadolu lisesi okul türünde ve 9. sınıf düzeyinde incelenmiştir. Programlama öğretiminde metafor kullanımının etkileri diğer lise türleri ve farklı sınıf düzeylerinde incelenebilir, sonraki araştırmalarda metin tabanlı programlamaya yeni başlayan ilkökul, ortaokul ve yükseköğretim öğrencileri üzerinde de çalışılabilir.

- Araştırma bulguları, 9. sınıfta uygulanan bilgisayar bilimi (Kur-1) dersi programlama becerileri ile sınırlıdır. Benzer araştırmalarda, metafora dayalı öğretimin etkisi öğretim programında yer alan başka beceriler ya da içerik kullanılarak test edilebilir.

- Araştırma Türkiye’de, Ege bölgesinde yer alan okullarla gerçekleştirilmiştir. Benzer araştırmalar farklı bölge ve ülkelerde gerçekleştirilebilir. Kültürün ve farklı eğitim sistemlerinin programlama öğretiminde kullanılan metaforlara etkisi incelenerek metaforların benzerlik ya da farklılıkları tanımlanabilir.

- Araştırmada, metafora dayalı öğretimin etkililiği kontrol grubunda kullanılan sunuş yoluyla öğretim stratejisine göre belirlenmiştir. Başka araştırmalarda bu stratejisinin dışında öğretim stratejilerinin kullanıldığı öğretim süreçlerinde metafora dayalı öğretimin kavram öğrenmeye, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliğe olan etkileri karşılaştırmalı araştırılabilir.

- Araştırmada çağdaş metafor kuramlarından Kavramsal Metafor Kuramı temele alınarak metafor kullanımı araştırmanın bağımlı değişkenleri açısından incelenmiştir. Başka araştırmalarda diğer çağdaş metafor kuramlarından yararlanılarak oluşturulmuş öğrenme yaşantıları tasarlanabilir ve etkileri belirlenebilir.

- Metafor kullanımının, araştırmada incelenen bağımlı değişkenler üzerindeki etkileri dışında 2023 Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programında alan becerileri olarak tanımlanan bilimsel süreç, bilgi işlemsel süreç, üst bilişsel süreç, veri tanıma-işleme-analiz becerileri üzerindeki etkileri incelenebilir.

- Araştırmada, araştırmacı tarafından geliştirilen özgün metaforlar kullanılmıştır. Başka araştırmalarda öğretmen ve öğrencilerin programlama öğretiminde kullandıkları ya da basılı ve dijital kaynaklarda yer alan metaforlar durum çalışması aracılığıyla nitel olarak belirlenebilir. Belirlenen metaforlar arasından uygun bulunanların programlama öğretimine etkilerine yönelik nicel desende araştırmalar yürütülebilir.

- Programlama öğretiminde karşılaşılan kavram yanlışlarını önlemek ve ortadan kaldırmak için bu yanlışların oluşma nedenleri derinlemesine araştırılabilir.

- Metafora dayalı öğretimin metin tabanlı programlama öğretimindeki etkileri, programlamaya yönelik farklı ilgi ve yetenek düzeyindeki öğrencilerin programlama kavramlarını kazanmalarına, programlama becerilerine ve bilgi işlemsel düşünme öz yeterliklerine etkileri araştırılabilir.

- Programlama öğretimini gerçekleştirecek uygulayıcıların öğretim süreçlerini planlarken, uygularken ve değerlendirirken yararlanabilmeleri için metafora dayalı

öğrenmenin nasıl gerçekleştiğini ortaya koyabilecek derinlikli nitel çalışmalar gerçekleştirilebilir.

- Metafora dayalı öğretimi verimli ve etkili kılmak için uygulayıcıların taşıması gereken özellikler alanyazın taraması ya da uzman görüşleri aracılığıyla tanımlanabilir; bu özellikler, uygulayıcıların program yönelimleri açısından incelenebilir.

- Bu araştırma karma yöntem deneysel desenle yürütülmüştür. Öğretmenlerin mesleki gelişimlerini desteklemek, araştırma becerilerini geliştirmek ve böylelikle kendi öğretim süreçlerinin iyileşmesine katkıda bulunmak amacıyla metafora dayalı öğretime yönelik eylem araştırmaları yapılabilir.



KAYNAKÇA

- Abesadze, S., & Nozadze, D. (2020). Make 21st century education: The importance of teaching programming in schools. *International Journal of Learning and Teaching*, 6(3), 158-163.
- Ahlgren, K., Golden, A., & Magnusson, U. (2021). Metaphor in education: A multilingual and Scandinavian perspective. *Metaphor And The Social World*, 11(2), 196-211. <https://doi.org/10.1075/msw.00015.ahl>
- Ahmady, S., Yaghmaei, M., Arab, M., & Monajemi, A. (2016). Metaphor in education; hidden but effective. *Journal of Medical Education*, 15(1), 52-57. <https://doi.org/10.22037/jme.v15i1.10401>
- Akyol, C. (2019). Metaforun kullanım alanları ve faydaları. B. Kılcan (Ed.), *Metafor ve eğitimde metaforik çalışmalar için bir uygulama rehberi* (s. 49-85) içinde. Pegem Akademi.
- Alammary, A. (2019). Blended learning models for introductory programming courses: A systematic review. *PloS One*, 14(9), 1-26. <https://doi.org/10.1371/journal.pone.0221765>
- Alper, A. ve Öztürk, S. (2019). Programlama öğretimindeki ters-yüz öğretim yönteminin öğrencilerin başarılarına, bilgisayara yönelik tutumuna ve kendi kendine öğrenme düzeylerine etkisi. *Bilim Eğitim Sanat ve Teknoloji Dergisi*, 3(1), 13-26.
- Arocena, I., Huegun-Burgos, A., & Rekalde-Rodriguez, I. (2022). Robotics and Education: A systematic review. *TEM Journal*, 11(1), 379-387.
- Arslan, D., & Aytaç, A. (2010). İlkokuma yazma öğretimi dersine ilişkin tutum ölçeği geliştirilmesi. *İlköğretim Online*, 9(3), 841-850.
- Atman Uslu, N. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1), 19-31.
- Aubusson, P. J., Harrison A. G., & Ritchie, S.M. (2006). Metaphor and Analogy. In P. J. Aubusson et al. (Eds.), *Metaphor and Analogy in science education* (pp. 1-9). Springer.
- Aydoğdu, Ş. (2020). Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine etkisi. *Eğitim Teknolojisi Kuram ve Uygulama*, 10(1), 303-320. <https://doi.org/10.17943/etku.649585>
- Ayverdi, L., Şahin, E., & Sarı, U. (2023). A STEM activity for gifted students: Biodegradable smart packaging design through physical computing. *Journal of Inquiry Based Activities*, 13(1), 54-79.
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future. Computer programming and coding: Priorities, school curricula and initiatives across Europe*. European Schoolnet. <http://www.eun.org/resources/detail?PublicationID=481>

- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191-215. <https://doi.org/10.1037/0033-295X.84.2.191>
- Barmpoutis, A., & Huynh, K. (2019). Name tags and pipes: Assessing the role of metaphors in students' early exposure to computer programming using emoticon coding. In S. Nazir et al. (Eds), *Advances in Human Factors in Training, Education, and Learning Sciences* (pp. 194-202). Springer.
- Barr, P., Biddle, R., & Noble, J. (2002). A taxonomy of user-interface metaphors. In S. Jones & M. Masoodian (Eds.), *Proceedings of the SIGCHI-NZ Symposium on Computer-Human Interaction* (pp.25-30). Association for Computing Machinery. <https://doi.org/10.1145/2181216.2181221>
- Barut, E. ve Kuzu, A. (2017). Türkiye ve İngiltere bilişim teknolojileri öğretim programlarının amaç, kazanım, etkinlik, ölçme ve değerlendirme süreçleri açısından karşılaştırılması. *Trakya Üniversitesi Eğitim Fakültesi Dergisi*, 7(2),721-745. <https://doi.org/10.24315/trkefd.303156>
- Beger, A., & Jakel, O. (2015). The cognitive role of metaphor in teaching science: Examples from physics, chemistry, biology, psychology and philosophy. *Philosophical Inquiries*, 3(1), 89-112. <https://doi.org/10.4454/philing.v3i1.116>
- Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *Association for Computing Machinery Inroads*, 10(2), 30-36. <https://doi.org/10.1145/3324888>
- Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers Education*, 6, 499–528. <https://doi.org/10.1007/s40692-019-00147-3>
- Botha, E. (2009). Why metaphor matters in education. *South African Journal of Education*, 29(4). 431- 444.
- Brist, A. H. (2012). *The effect of a contextual approach to chemistry instruction on students' attitudes, confidence, and achievement in science* [Master's thesis, Montana State University]. <https://scholarworks.montana.edu/items/3d798bf2-6ecd-4793-8c86-9bfec4b97fa8>
- Brooks, J. G. & Brooks, M. G. (1999). *In search of understanding: The case for constructivist classrooms*. Association for Supervision and Curriculum Development
- Burke, Q. (2016). Mind the metaphor: Charting the rhetoric about introductory programming in K-12 schools. *On the Horizon*, 24(3), 210-220. <https://doi.org/10.1108/OTH-03-2016-0010>
- Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 433-438). ACM. <https://doi.org/10.1145/2157136.2157264>
- Busetto, L., Wick, W., & Gumbinger, C. (2020). How to use and assess qualitative research methods. *Neurological Research and Practice*, 2(14), 1-10. <https://doi.org/10.1186/s42466-020-00059-z>

- Butt, A. (2014). Student views on the use of a flipped classroom approach: Evidence from Australia. *Business Education & Accreditation*, 6(1), 33-43.
- Butterworth, J., & Thwaites, G. (2013). *Thinking skills: Critical thinking and problem solving*. Cambridge University Press.
- Cambridge Dictionary, (2023). Metaphor. <https://dictionary.cambridge.org/adresinden> 7 Kasım 2023 tarihinde alınmıştır.
- Campbell, D., & Stanley, J. (1963). *Experimental and quasi-experimental designs for research*. Rand McNally & Company.
- Campbell-Larsen, J. (2012). Metaphors We Teach By. In *EFL Teacher Journeys Conference Proceedings* (pp.22-28). Teacher Education and Development SIG.
- Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, 30(1), 22-46. <https://doi.org/10.1080/08993408.2019.1648119>
- Cao, Y., Porter, L., & Zingaro, D. (2016). Examining the value of analogies in introductory computing. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (pp. 231-239). Association for Computing Machinery. <https://doi.org/10.1145/2960310.2960313>
- Celepcolu, M., & Boyer, K. E. (2018). Thematic analysis of students' reflections on pair programming in CS1. In *49th ACM Technical Symposium on Computer Science Education* (pp.771-776). Association for Computing Machinery. <https://doi.org/10.1145/3159450.3159516>
- Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2), 1-14.
- Chibaya, C. (2019). A metaphor-based approach for introducing programming concepts. In *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)* (pp. 1-8). Institute of Electrical and Electronics Engineers.
- Chou, C. H., Su, Y. S., & Chen, H. J. (2019). Interactive teaching aids integrating building blocks and programming logic. *Journal of Internet Technology*, 20(6), 1709-1719.
- Cienki, A. (2016). Analysing metaphor in gesture: A set of metaphor identification guidelines for gesture (MIG-G). In *The Routledge handbook of metaphor and language* (pp. 149–165). Routledge
- Coblentz, M., Aldrich, J., Myers, B. A., & Sunshine, J. (2018). Interdisciplinary programming language design. In E. G. Boix & R. P. Gabriel (Eds.), In *Proceedings of the 2018 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (pp. 133-146). Association for Computing Machinery. <https://doi.org/10.1145/3276954.3276965>
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates.

- Coleman, S. A., & Nichols, E. (2011). Embedding inquiry based learning into programming via paired assessment. *Innovation in Teaching and Learning in Information and Computer Sciences*, 10(1), 72-77.
- Corno, F., Russis, L. D., & Monge Roffarello, A. (2019). My iot puzzle: Debugging if-then rules through the jigsaw metaphor. In A. Malizia, S. Valtolina, A. Morch, A. Serrano, A. & A. Stratton (Eds.), *International Symposium on End User Development* (pp. 18-33). Springer. https://doi.org/10.1007/978-3-030-24781-2_2
- Cortazzi, M., & Jin, L. (1999). Cultural mirrors. In E. Hinkel (Ed.), *Culture in second language teaching and learning* (pp.196-219). Cambridge University Press.
- Cortina, T. J. (2015). Reaching a broader population of students through “unplugged” activities. *Communications of the ACM*, 58(3), 25-27. <http://dx.doi.org/10.1145/2723671>
- Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and Conducting Mixed Methods Research* (3rd ed.). Thousand Oaks, CA: SAGE.
- Crick, T., Knight, C., Watermeyer, R., & Goodall, J. (2020). The impact of COVID-19 and “Emergency Remote Teaching” on the UK computer science education community. In J. Maguire & Q. Cutts (Eds.), *United Kingdom & Ireland Computing Education Research Conference* (pp. 31-37). Association for Computing Machinery.
- Cronje, J. (2020). Towards a new definition of blended learning. *Electronic Journal of e-Learning*, 18(2), 114-121. <https://doi.org/10.34190/EJEL.20.18.2.001>
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T. ..., & Woollard, J. (2015). *Computational Thinking A Guide for Teachers*. Retrieved July 06, 2020 from <https://community.computingschool.org.uk/resources/2324/single>.
- Csorba, D. (2015). Metaphor in Science Education. Implications for teacher education. *Procedia-Social and Behavioral Sciences*, 180, 765-773.
- Cücü, H. ve Dağ, F. (2023). Sınıf öğretmenleri açısından bilgi işlemsel düşünme ve kodlama öğretiminin kapsamının belirlenmesine yönelik bir inceleme. *Milli Eğitim Dergisi*, 52(239), 1807-1844. <https://doi.org/10.37669/milliegitim.1148733>
- Çakır, R. (2019). Programlama öğretimi ve düşünme becerileri. Özgen Korkmaz (Ed.), *Programlama öğretimi yaklaşımları* (s. 15-36) içinde. Nobel Akademik.
- Çatlak, Ş., Tekdal, M. ve Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Öğretim Teknolojileri ve Öğretmen Eğitimi Dergisi*, 4(3), 13-25.
- Çelenk, K. (2019). Evaluation of perceptions of secondary school students about the concept of "music" by metaphors. *Journal of Education and Training Studies*, 7(9), 70-89. <https://doi.org/10.11114/jets.v7i9.4304>
- Çetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, 47, 70-80. <https://doi.org/10.1016/j.jmathb.2017.06.004>

- Çetin, İ. (2018). Bilgi işlemsel düşünme tanımı ve kapsamı. Y. Gülbahar ve H. Karal (Ed.), *Kuramdan uygulamaya programlama öğretimi* (s. 41-78) içinde. Pegem Akademi.
- Çiçekler, A. N. ve Aydın, T. (2019). Kavramsal metafor kuramı ve belagat: Karşılaştırmalı bir inceleme. *RumeliDE Dil ve Edebiyat Araştırmaları Dergisi*, 16, 14-26. <https://doi.org/10.29000/rumelide.616880>
- Danesi, M. (2007). A conceptual metaphor framework for the teaching of mathematics. *Studies in Philosophy And Education*, 26, 225-236. <https://doi.org/10.1007/s11217-007-9035-5>
- Dangwal, K. L. (2017). Blended learning: An innovative approach. *Universal Journal of Educational Research*, 5(1), 129-136. <https://doi.org/10.1016/j.sbspro.2013.09.248>
- Demir, M. (2009). Batı metaforu ve doğu istiaresinin mukayeseli olarak incelenmesi. *Türkbilig*, 18, 64-90.
- Demir, R., Öztürk, N., & Dökme, İ. (2011). The views of the teachers taking in-service training about alternative measurement and evaluation techniques: The sample of primary school teachers. *Procedia-Social and Behavioral Sciences*, 15, 2347-2352. <https://doi.org/10.1016/j.sbspro.2011.04.105>
- Demirbilek, N., & Korkmaz, C. (2021). Study concept as a metaphor from the lenses of university students. *African Educational Research Journal*, 9(1), 227-236.
- Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Denny, P., MacNeil, S., Savelka, J., Porter, L., & Luxton-Reilly, A. (2024). Desirable characteristics for ai teaching assistants in programming education. In *Proceedings of Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 1-7). Association for Computing Machinery.
- Derisma, D. (2020). The usability analysis online learning site for supporting computer programming course using system usability scale (SUS) in a university. *International Journal of Interactive Mobile Technologies*, 14(9), 182-195. <https://doi.org/10.3991/ijim.v14i09.13123>
- Derus, S. R. M., & Ali, A. Z. M. (2012). Difficulties in learning programming: Views of students. In *1st international conference on current issues in education (ICCIE 2012)* (pp. 74-78). ICCIE.
- Doleck, T., Bazelaïs, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4, 355-369.
- Dongo, T., Reed, A. H., & O'Hara, M. (2016). Exploring pair programming benefits for MIS majors. *Journal of Information Technology Education: Innovations in Practice*, 15, 223-239.
- Doolittle, P. E. (2014). Complex constructivism: A theoretical model of complexity and cognition. *International Journal of Teaching And Learning in Higher Education*, 26(3), 485-498.

- Doppelt, Y. (2005). Assessment of project-based learning in a mechatronics context. *Journal of Technology Education*, 16(2), 7-24.
- Duban, N., & Küçükylmaz, E. A. (2008). Primary education pre-service teachers opinions regarding to the use of alternative measurement-evaluation methods and techniques in practice schools. *Elementary Education Online*, 7(3), 769-769.
- Effenberger, T., & Pelanek, R. (2019). Measuring students' performance on programming tasks. In *Sixth (2019) ACM Conference on Learning@ Scale* (pp.1-4), Association for Computing Machinery. <https://doi.org/10.1145/3330430.3333639>
- Ekici, G. (2016). Biyoloji öğretmeni adaylarının mikroskop kavramına ilişkin algılarının belirlenmesi: Bir metafor analizi çalışması. *Kırşehir Eğitim Fakültesi Dergisi*, 17(1), 615-636.
- Ernest, P. (1994). Varieties of constructivism: Their metaphors, epistemologies and pedagogical implications. *Hiroshima Journal of Mathematics Education*, 2, 1-22.
- Erümit, A. K., Şahin, G. ve Karal, H. (2020). YAP programlama öğretim modelinin öğrencilerin bilgi-işlemsel düşünme becerilerine etkisi. *Kastamonu Eğitim Dergisi*, 28(3), 1529-1540. <https://doi.org/10.24106/kefdergi.3915>
- Erümit, K. A., Karal, H., Şahin, G., Aksoy, D. A., Aksoy, A., & Benzer, A. İ. (2019). A model suggested for programming teaching: Programming in seven steps. *Education and Science*, 44(197), 155-183. <http://dx.doi.org/10.15390/EB.2018.7678>
- Eryılmaz, S. (2010). Web ortamında öge gösterim kuramına göre tasarlanan kavram öğretiminin uygulaması: programlamada dizi kavramının öğretimi. *Gazi Üniversitesi Ticaret ve Turizm Eğitim Fakültesi Dergisi*, (2), 100-127.
- Eryılmaz, S. ve Deniz, G. (2019). Türkiye'de programlama eğitimi ile ilgili yapılan çalışmaların incelenmesi: Bir betimsel analiz çalışması. *Eğitimde Kuram ve Uygulama*, 15(4), 319-338. <https://doi.org/10.17244/eku.645387>
- Esper, S., Foster, S. R., & Griswold, W. G. (2013). CodeSpells: Embodying the metaphor of wizardry for programming. In *18th ACM conference on Innovation and Technology in Computer Science Education* (pp. 249-254). Association for Computing Machinery. <https://doi.org/10.1145/2462476.2465593>
- Estácio, B., Oliveira, R., Marczak, S., Kalinowski, M., Garcia, A., Prikladnicki, R., & Lucena, C. (2015). Evaluating collaborative practices in acquiring programming skills: Findings of a controlled experiment. In *29th Brazilian Symposium on Software Engineering* (pp. 150-159). IEEE.
- European Commission (2020). *Digital Education Action Plan (2021-2027)*. <https://education.ec.europa.eu/focus-topics/digital-education/action-plan>
- European Commission. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice*. <http://www.eun.org/resources/detail?publicationID=861>.
- Fadiana, M., Amin, S. M., & Lukito, A. (2018). The seventh grade students' generalization strategies of patterns. *Journal of Education and Learning (EduLearn)*, 12(3), 333-339. <https://doi.org/10.11591/edulearn.v12i3.8997>

- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12-28. <https://doi.org/10.1002/cae.22255>
- Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., ... & Quille, K. (2019). An international study piloting the measuring teacher enacted computing curriculum (METRECC) instrument. In *Working Group Reports on Innovation and Technology in Computer Science Education* (pp. 111-142). Association for Computing Machinery. <https://doi.org/10.1145/3344429.3372505>
- Fanny, B., Julie, H., & Anne-Sophie, C. (2020). Developing a critical robot literacy for young people from conceptual metaphors analysis. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-7). IEEE.
- Farida, F., Supriadi, N., Andriani, S., Pratiwi, D. D., Suherman, S., & Muhammad, R. R. (2022). STEM approach and computer science impact the metaphorical thinking of Indonesian students'. *Revista de Educación a Distancia (RED)*, 22(69). <http://dx.doi.org/10.6018/red.493721>
- Fidan, A. (2016). *Scratch ile programlama öğretiminde oyunlaştırmanın öğrenci katılımına etkisi* [Doktora Tezi, Uludağ Üniversitesi]. Yüksek Öğretim Kurulu Başkanlığı Tez Merkezi.
- Fincher, S., & Knox, D. (2013). The porous Classroom: Professional practices in the computing curriculum, *Computer*, 46(9), 44-51.
- Fink, A. S. (2000). The role of the researcher in the qualitative research process. A potential barrier to archiving qualitative data. *Forum Qualitative Sozialforschung Forum: Qualitative Social Research*, 1(3), 1-15.
- Fisler, K., Schanzer, E., Weimar, S., Fetter, A., Renninger, K. A., Krishnamurthi, S., ... & Koerner, C. (2021). Evolving a K-12 curriculum for integrating computer science into mathematics. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 59-65). Association for Computing Machinery.
- Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. *Journal of Educational Technology & Society*, 19(3), 38-46.
- Font, V., Bolite, J., & Acevedo, J. (2010). Metaphors in mathematics classrooms: analyzing the dynamic process of teaching and learning of graph functions. *Educational Studies in Mathematics*, 75(2), 131-152. <https://doi.org/10.1007/s10649-010-9247-4>
- Forišek, M., & Steinová, M. (2012). Metaphors and analogies for teaching algorithms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 15-20). Association for Computing Machinery. <https://doi.org/10.1145/2157136.2157147>
- Fox, R. (2001). Constructivism examined. *Oxford Review of Education*, 27(1), 23-35. <https://doi.org/10.1080/03054980125310>

- Funcke, M. (2014). *Developing high-fidelity mental models of programming concepts using manipulatives and interactive metaphors* [Doctoral dissertation, Rhodes University].
- Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In I. Kalaš ve R. T. Mittermeir (Eds.), *International conference on informatics in schools: Situation, evolution, and perspectives* (pp.155-164). Springer. https://doi.org/10.1007/978-3-642-24722-4_14
- Gagnon, G. W., & Collay, M. (2001). *Designing for learning: Six elements in constructivist classrooms*. Corwin Press.
- García-Peñalvo, F. J., & Cruz-Benito, J. (2016). Computational thinking in pre-university education. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 13-17). Association for Computing Machinery. <https://doi.org/10.1145/3012430.3012490>
- Gary, K. (2015). Project-Based Learning. *Computer*, 48(9), 98-100.
- Gaspar, A., & Langevin, S. (2007). Restoring “coding with intention” in introductory programming courses. In *The 8th ACM SIGITE Conference on Information Technology Education*, (pp. 91-98). Association for Computing Machinery. <https://doi.org/10.1145/1324302.1324323>
- Gibbs, K., & Beamish, W. (2020). Differentiated instruction: A programming tool for inclusion. In *Inclusive Theory And Practice in Special Education* (pp. 174-191). IGI Global.
- Gibson, Cristina. B., & Mary E. Zellmer-Bruhn, (2001). Metaphors and meaning: An intercultural analysis of the concept of teamwork. *Administrative Science Quarterly*, 46(2), 274–303. <http://dx.doi.org/10.2307/2667088>
- Gil Fonseca, N., Macedo, L., & Mendes, A. J. (2018). Supporting differentiated instruction in programming courses through permanent progress monitoring. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 209-214). Association for Computing Machinery. <https://doi.org/10.1145/3159450.3159578>
- Goode, J., Ivey, A., Johnson, S. R., Ryoo, J. J., & Ong, C. (2021). Rac(e)ing to computer science for all: How teachers talk and learn about equity in professional development. *Computer Science Education*, 31(3), 374-399.
- Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 493-498). Association for Computing Machinery.
- Gökoğlu, S. (2017). Programlama eğitiminde algoritma algısı: Bir metafor analizi. *Cumhuriyet Uluslararası Eğitim Dergisi*, 6(1), 1-14.

- Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, E. Barendsen ve C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19-34). Bloomsbury Publishing.
- Guba, E. G., & Lincoln, Y. S. (1982). Epistemological and methodological bases of naturalistic inquiry. *Educational Communication and Technology Journal*, 30(4), 233-252.
- Gunawan, G., Harjono, A., Sahidu, H., & Herayanti, L. (2017). Virtual laboratory to improve students' problem-solving skills on electricity concept. *Jurnal Pendidikan IPA Indonesia*, 6(2), 257-264.
- Gülbahar, Y. ve Kalelioğlu, F. (2018). Bilişim teknolojileri ve bilgisayar bilimi: Öğretim programı güncelleme süreci. *Millî Eğitim Dergisi*, 47(217), 5-23.
- Gülbahar, Y., & Kalelioğlu, F. (2017). Competencies of high school teachers and training needs for computer science education. In *Proceedings of the 6th Computer Science Education Research Conference* (pp. 26-31). Association for Computing Machinery. <https://doi.org/10.1145/3162087.3162092>
- Güneş, C. ve Tezcan, R. (2019). Metafor nedir, ne değildir? B. Kılcan (Ed.), *Metafor ve eğitimde metaforik çalışmalar için bir uygulama rehberi* (s. 1-14) içinde. Pegem Akademi.
- Hall, T. (2002). Differentiated instruction. www.cast.org/publications/ncac/ncac-diffinstruc.html adresinden 7 Kasım 2023 tarihinde alınmıştır.
- Hamlen, K., Sridhar, N., Bievenue, L., Jackson, D.K., & Lalwani, A. (2018). Effects of teacher training in a computer science principles curriculum on teacher and student skills, confidence, and beliefs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 741-746). Association for Computing Machinery. <https://doi.org/10.1145/3159450.3159496>
- Hampe, B. (2017). Embodiment and discourse: Dimensions and dynamics of contemporary metaphor theory. In B. Hampe (Ed.), *Metaphor: Embodied cognition and discourse* (pp.3-24). Cambridge University Press.
- Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *2016 IEEE Frontiers in Education Conference* (pp.1-9). Institute of Electrical and Electronics Engineers.
- Hendrik, H., & Hamzah, A. (2021). Flipped classroom in programming course: A systematic literature review. *International Journal of Emerging Technologies in Learning (iJET)*, 16(2), 220-236. <https://doi.org/10.3991/ijet.v16i02.15229>
- Hermans, F., Swidan, A., Aivaloglou, E., & Smit, M. (2018). Thinking out of the box: comparing metaphors for variables in programming education. In A. Mühling & Q. Cutts (Eds.), *Proceedings of the 13th workshop in primary and secondary computing education* (pp.1-8). Association for Computing Machinery. <https://doi.org/10.1145/3077618>

- Hidalgo- Cespedes, J., Marín- Raventos, G., Lara- Villagran, V., & Villalobos- Fernández, L. (2018). Effects of oral metaphors and allegories on programming problem solving. *Computer Applications in Engineering Education*, 26(4), 852-871. <https://doi.org/10.1002/cae.21927>
- Hong, Q. N., Pluye, P., Fabregues, S., Bartlett, G., Boardman, F., Cargo, M., ..., & Vedel, I. (2019). Improving the content validity of the mixed methods appraisal tool: a modified e-Delphi study. *Journal of clinical epidemiology*, 111, 49-59. <https://doi.org/10.1016/j.jclinepi.2019.03.008>
- Hromkovic, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, 10(1-2), 111-124.
- Hubálovský, S. (2012). Research of methods of a multidisciplinary approach in the teaching of algorithm development and programming. In M. Cápay, M. Mesárošová, V. Palmárová (Eds.), *9th international scientific conference on distance learning in applied informatics-DIVAI* (pp. 147-156).
- Hui, T. H., & Umar, I. N. (2011). Does a combination of metaphor and pairing activity help programming performance of students with different self-regulated learning level? *Turkish Online Journal of Educational Technology-TOJET*, 10(4), 121-129.
- Hung, D. W. (2002). Metaphorical ideas as mediating artifacts for the social construction of knowledge: Implications from the writings of Dewey and Vygotsky. *International Journal of Instructional Media*, 29(2), 197.
- ISTE (2019). The ISTE national educational technology standards (NETS) and performance indicators for students. Retrieved April 01, 2020 from <http://www.iste.org/standards/nets-for-students>.
- Jamrozik, A., McQuire, M., Cardillo, E. R., & Chatterjee, A. (2016). Metaphor: Bridging embodiment to abstraction. *Psychonomic Bulletin & Review*, 23, 1080-1089. <https://doi.org/10.3758/s13423-015-0861-0>
- Jenkins, J. R., & Pany, D. (1978). Standardized achievement tests: How useful for special education? *Exceptional Children*, 44(6), 448-453. <https://doi.org/10.1177/001440297804400606>
- Jiménez-Peris, R., Pareja-Flores, C., Patiño-Martínez, M., & Velázquez-Iturbide, J. Á. (1997). The locker metaphor to teach dynamic memory. In *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education* (pp. 169-173). Association for Computing Machinery.
- Jumankuziev, U. (2023). Using new approaches to teaching programming languages in secondary schools. *Galaxy International Interdisciplinary Research Journal*, 11(5), 280–283.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, (52), 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>.
- Kalelioğlu, F. (2018). Türkiye’de programlama öğretimi. Y. Gülbahar ve H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* (s. 67-89) içinde. Pegem Akademi.

- Kalelioğlu, F. (2020). Bilgisayarsız bilgisayar bilimi öğretimi. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (s. 183-206) içinde. Pegem Akademi.
- Kalelioğlu, F. ve Gülbahar, Y. (2015). *Bilgi işlemsel düşünme nedir ve nasıl öğretilir?* [Sempozyum Bildirisi]. 3. Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumu, Trabzon, Türkiye.
- Kan, A. (2007). Ödev ve Projeler. H. Atılgan (Ed.), *Eğitimde ölçme ve değerlendirme* (s. 269-292) içinde. Anı Yayıncılık.
- Karaman, S., Anders, A., Boulet, M., Connor, J., Gregson, K., Guerra, W., ... & Vivilecchia, J. (2017). Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT. In *IEEE Integrated STEM Education Conference (ISEC)* (pp.195-203). Institute of Electrical and Electronics Engineers.
- Karasar, N. (2009). *Bilimsel Araştırma Yöntemleri*. Nobel Yayınları.
- Kert, S.B. (2018). Programlama öğretimi için pedagojik yaklaşımlar. Y. Gülbahar ve H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* (s. 37-65) içinde. Pegem Akademi.
- Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., McGinnis, J.R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology*, 29, 174–188. <https://doi.org/10.1007/s10956-019-09798-4>
- Kim, D., Choi, J., Jung, I. H., & Choi, C. (2018). Pedagogy of programming education for higher education using block based programming environment. In A. Marcus & W. Wang (Eds.), *International Conference of Design, User Experience, and Usability* (pp.39-50). Springer. https://doi.org/10.1007/978-3-319-91803-7_3
- Konan, F. (2020). *Programlama öğretimine yönelik bir içerik analizi* [Yüksek lisans tezi, Erzincan Binali Yıldırım Üniversitesi]. Yüksek Öğretim Kurulu Başkanlığı Tez Merkezi.
- Koohang, A., & Harman, K. (2005). Open source: A metaphor for e-learning. *Informing Science*, 8, 75-86. <http://dx.doi.org/10.28945/488>
- Korkmaz, Ö. (2019). Programlama becerilerinin ölçülmesi. Ö. Korkmaz (Ed.) *Programlama öğretimi yaklaşımları* (s. 235-259) içinde. Nobel Akademik.
- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, 14(4), 1-28. <https://doi.org/10.1145/2662412>
- Kövecses, Z. (2010). *Metaphor- A Practical Introduction*. Oxford University Press.
- Krippendorff, K. (1993). Major metaphors of communication and some constructivist reflections on their use. *Cybernetics & Human Knowing*, 2(1), 1-27.

- Kukul, V. ve Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. 8. *Uluslararası Bilgisayar & Öğretim Teknolojileri Sempozyumu Bildiriler Kitabı* içinde (s. 58-63). <http://dx.doi.org/10.13140/RG.2.1.4563.7606>
- Kuo, Y. C., & Chang, Y. C. (2023). Adaptive teaching of flipped classroom combined with concept map learning diagnosis-an example of programming design course. *Education and Information Technologies*, 28(7), 8665-8689. <https://doi.org/10.1007/s10639-022-11540-4>
- Kuzu, A. ve Türk, M. (2018). Fiziksel Programlama. Y. Gülbahar ve H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* (s. 339-388) içinde. Pegem Akademi.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM Sigcse Bulletin*, 37(3), 14-18. <http://dx.doi.org/10.1145/1067445.1067453>
- Lakoff, G. (1993). The contemporary theory of metaphor. In A. Ortony (Ed.), *Metaphor and Thought* (pp. 202-251). Cambridge University Press.
- Lakoff, G. ve Johnson, M. (2015). *Metaforlar: Hayat, anlam ve dil* [Metaphors we live by], (G. Y. Demir, Çev.). İthaki.
- Lakoff, G., & Johnson, M. (1980). Conceptual metaphor in everyday language. *The Journal of Philosophy*, 77(8), 453-486.
- Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. Basic Books.
- Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, 30(6), 45-52.
- Lancor, R. (2014). Using metaphor theory to examine conceptions of energy in biology, chemistry, and physics. *Science & Education*, 23(6), 1245-1267. <https://doi.org/10.1007/s11191-012-9535-8>
- Landis, J. R., & Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, 33(2), 363-374.
- Lane, R. M. (2018). An ecology for learning and teaching: Niches for theories within the habitat of learning. In M. Hanne & A. A. Kaal (Eds.), *Narrative and Metaphor in Education* (pp. 46-59). Routledge.
- Langston, W. (2002). Violating orientational metaphors slows reading. *Discourse Processes*, 34(3), 281-310. https://doi.org/10.1207/S15326950DP3403_3
- Larsson, A. (2022). Reading the code between the lines – exploring the structure of metaphors in educational programming resources. *Nordic Studies in Science Education*, 18(3), 305-322. <https://doi.org/10.5617/nordina.8774>
- Larsson, A., & Stolpe, K. (2023a). Hands on programming: teachers' use of metaphors in gesture and speech make abstract concepts tangible. *International Journal of Technology and Design Education*, 33, 901-919. <https://doi.org/10.1007/s10798-022-09755-0>

- Larsson, A., & Stolpe, K. (2023b). Exploring the metaphoric nature of programming teachers' reflections on action-a case study with teaching in mind. *International Journal of Technology and Design Education*, 34, 585-602. <https://doi.org/10.1007/s10798-023-09826-w>
- Lazarinis, F., Karachristos, C. V., Stavropoulos, E. C., & Verykios, V. S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies*, 24(2), 1237-1249. <https://doi.org/10.1007/s10639-018-9823-2>
- Leary, D. E. (1990). Psyche's muse: The role of metaphor in the history of psychology. In D. E. Leary (Ed.), *Metaphors in the history of psychology* (pp. 1-78). Cambridge University Press.
- LeCompte, M. D., & Goetz, J. P. (1982). Problems of reliability and validity in ethnographic research. *Review of Educational Research*, 52, 31-60. <https://doi.org/10.3102/00346543052001031>
- Lee, N. T. S., Kurniawan, O., & Choo, K. T. W. (2021). Assessing programming skills and knowledge during the COVID-19 pandemic: An experience report. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 352-358).
- Levine, P. M. (2005). Metaphors and images of classrooms. *Kappa Delta Pi Record*, 41(4), 172-175. <https://doi.org/10.1080/00228958.2005.10532066>
- Li, X. (2010). Conceptual metaphor theory and teaching of English and Chinese idioms. *Journal of Language Teaching and Research*, 1(3), 206-210.
- Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K- 12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, 50(4), 1979-1995. <https://doi.org/10.1111/bjet.12685>
- López-Pernas S., Gordillo A., Barra E., & Quemada J. (2019). Examining the use of an educational escape room for teaching programming in a higher education setting. In *Institute of Electrical and Electronics Engineers (IEEE) Access* (pp. 31723-31737).
- Low, G. (2008). Metaphor and education. In R. W. Gibbs (Ed.), *The Cambridge handbook of metaphor and thought* (pp. 212-231). Cambridge.
- Lukose, J., & Mammen, K. J. (2018). Enhancing academic achievement in an introductory computer programming course through the implementation of guided inquiry-based learning and teaching. *Asia-Pacific Forum on Science Learning & Teaching* 19(2), 1-36.
- Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., ... & Szabo, C. (2018). Introductory programming: a systematic literature review. In *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education* (pp. 55-106). <https://doi.org/10.1145/3293881.3295779>

- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in human behavior*, 41, 51-61.
- Lynch, H. L., & Fisher-Ari, T. R. (2017). Metaphor as pedagogy in teacher education. *Teaching and Teacher Education*, 66, 195-203. <https://doi.org/10.1016/j.tate.2017.03.021>
- Mabie, A. K., McGill, M., & Huerta, B. (2023). A systematic literature review examining the impacts of integrating Computer Science in K-5 settings. In *2023 ASEE Annual Conference & Exposition* (pp. 3-20).
- Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem solving in the context of computational thinking. *Infinity Journal*, 8(2), 109-116.
- Major, L., Kyriacou, T., & Brereton, P. (2011). Experiences of prospective high school teachers using a programming teaching tool. In *11th Koli Calling International Conference on Computing Education Research* (pp. 126-131). Association for Computing Machinery. <https://doi.org/10.1145/2094131.2094161>
- Malik, S. I., Shakir, M., Eldow, A., & Ashfaq, M. W. (2019). Promoting algorithmic thinking in an introductory programming course. *International Journal of Emerging Technologies in Learning (iJET)*, 14(1), 84-94.
- McCandless, B. (2012). The use and misuse of metaphor in education and education reform. *Education*, 132(3), 538-547.
- MEB (1998). İlköğretim okulları seçmeli bilgisayar dersi (1-2-3-4-5) öğretim programı. <http://tebligler.meb.gov.tr/index.php/tuem-sayilar/viewcategory/62-1998> adresinden 20 Aralık 2020 tarihinde alınmıştır.
- MEB (2006a). İlköğretim seçmeli bilgisayar (4-8. Sınıflar) dersi öğretim programı. http://computereducation.web.tr/wpcontent/uploads/2017/10/bilgisayar_1_8_programi.pdf adresinden 20 Aralık 2020 tarihinde alınmıştır.
- MEB (2006b). Ortaöğretim bilgi ve iletişim teknolojisi dersi öğretim programı. (Yayın no. 2588). <http://tebligler.meb.gov.tr/index.php/tuem-sayilar/viewcategory/70-2006> adresinden 21 Aralık 2020 tarihinde alınmıştır.
- MEB (2012). Ortaokul ve imam hatip ortaokulu bilişim teknolojileri ve yazılım dersi (5, 6, 7 ve 8. Sınıflar) öğretim programı. <http://yunus.hacettepe.edu.tr/~htuzun/courses/bto202016spring/Bili%3Fim%20Tehnolojileri%20ve%20Yaz%3Fl%3Fim%20Dersi%20%3F%3Fretim%20Program%3F.pdf> adresinden 21 Aralık 2020 tarihinde alınmıştır.
- MEB (2018a). Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 5 ve 6. sınıflar). <http://mufredat.meb.gov.tr/> adresinden 22 Aralık 2020 tarihinde alınmıştır.
- MEB (2018b). Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 7 ve 8. sınıflar). <http://mufredat.meb.gov.tr/> adresinden 22 Aralık 2020 tarihinde alınmıştır.
- MEB (2018c). Ortaöğretim bilgisayar bilimi dersi (kur 1-2) öğretim programı. <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=335> adresinden 22 Aralık 2020 tarihinde alınmıştır.

- MEB (2023). Ortaöğretim bilişim teknolojileri ve yazılım dersi öğretim programı. <https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=1353> adresinden 28 Mart 2023 tarihinde alınmıştır.
- MEB (2024). Ortaöğretim matematik dersi öğretim programı (Hazırlık, 9, 10, 11 ve 12. sınıflar). <https://mufredat.meb.gov.tr/> adresinden 29 Haziran 2024 tarihinde alınmıştır.
- Menon, P., & Kovalchick, L. (2020). Using a concept map to represent the composition of knowledge in an introductory programming course. *Information Systems Education Journal*, 18(3), 4-17.
- Michaelson, G. (2015). Teaching programming with computational and informational thinking. *Journal of Pedagogic Development*, 5(1), 51-65.
- Michaelson, G. (2018). Microworlds, objects first, computational thinking and programming. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp.31-48). Springer. https://doi.org/10.1007/978-3-319-93566-9_3
- Miles, M., B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. (2nd ed.). Thousand Oaks, CA: Sage.
- Milner, W. W. (2010). A broken metaphor in Java. *ACM SIGCSE Bulletin*, 41(4), 76-77.
- Mladenovic, M., Boljat, I., & Zanko, Z. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, 23(4), 1483-1500. <https://doi.org/10.1007/s10639-017-9673-3>
- Moape, T., Ojo, S., & Van Wyk, E. (2018). A metaphor-driven interactive multimedia simulation for teaching and learning of programming concepts. In P. Cunningham & M. Cunningham (Eds.), *2018 IST-Africa Week Conference* (pp. 1-8). Institute of Electrical and Electronics Engineers.
- Moreno-León, J., Román-González, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1684-1689). Institute of Electrical and Electronics Engineers.
- Morgan, G. (2006). *Images of organization*. Sage.
- Morrison, C., Villar, N., Hadwen-Bennett, A., Regan, T., Cletheroe, D., Thieme, A., & Sentance, S. (2019). Physical programming for blind and low vision children at scale. *Human-Computer Interaction*, 36(5-6), 535-569. <https://doi.org/10.1080/07370024.2019.1621175>
- Müller, C., & Mildenerberger, T. (2021). Facilitating flexible learning by replacing classroom time with an online learning environment: A systematic review of blended learning in higher education. *Educational Research Review*, 34, 100394. <https://doi.org/10.1016/j.edurev.2021.100394>
- Ni, L., Bausch, G., & Benjamin, R. (2023). Computer science teacher professional development and professional learning communities: A review of the research literature. *Computer Science Education*, 33(1), 29-60. <https://doi.org/10.1080/08993408.2021.1993666>

- Niebert, K., & Gropengiesser, H. (2015). Understanding starts in the mesocosm: conceptual metaphor as a framework for external representations in science teaching. *International Journal of Science Education*, 37(5-6), 903-933. <https://doi.org/10.1080/09500693.2015.1025310>
- Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1), 463-484.
- Nouri, J., Zhang, L., Mannila, L., & Noren, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. <https://doi.org/10.1080/20004508.2019.1627844>
- Novak, J. D. (1990). Concept mapping: A useful tool for science education. *Journal of research in science teaching*, 27(10), 937-949. <https://doi.org/10.1002/tea.3660271003>
- Numanoğlu, M. ve Keser, H. (2017). Programlama öğretiminde robot kullanımı-mbot örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6(2), 497-515.
- Numic, M. (2020). *Metaphor-based tangible toolkit for programming education* [Master's thesis, Ostfold University College]. <https://hiof.brage.unit.no/hiof-xmlui/handle/11250/2660420?locale-attribute=en>
- OECD (2021). *The State of School Education: One Year into the COVID Pandemic*. <https://doi.org/10.1787/201dde84-en>.
- Oliveira, G. A. A., Assunção, O. B., & Prates, R. O. (2019). Using cultural viewpoint metaphors in the analysis of computational thinking teaching. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems* (pp. 1-13). Association for Computing Machinery.
- Olmo-Munoz, J., Cózar-Gutierrez, R., & Gonzalez-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers ve Education*, 15, 1-19. <https://doi.org/10.1016/j.compedu.2020.103832>
- Olsson, M., Mozelius, P., & Collin, J. (2015). Visualisation and gamification of e-Learning and Programming Education. *Electronic Journal of E-learning*, 13(6), 441-454.
- Oluk, A., Korkmaz, Ö. ve Oluk, H. A. (2018). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve bilgi-işlemsel düşünme becerilerine etkisi. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 9(1), 54-71. <https://doi.org/10.16949/turkbilmat.399588>
- Ontanon, S., Zhu, J., Smith, B. K., Char, B., Freed, E., Furqan, A., ... & Valls-Vargas, J. (2017). Designing visual metaphors for an educational game for parallel programming. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2818-2824). Association for Computing Machinery. <https://doi.org/10.1145/3027063.3053253>

- Özelmacı, Ş. ve Çakır, U. (2017). Metaforlar ve benzerlileri ile ilgili söylenenler. Kılcan, B. (Ed.), *Metafor ve eğitimde metaforik çalışmalar için bir uygulama rehberi* (s. 1-14) içinde. Pegem Akademi.
- Özer, S. (2017). Türkçede vücut organ adı olarak “kulak” sözcüğünün kavramsal içeriğine bilişsel yaklaşım. *Turkish Studies*, 12(7), 291-310. <http://dx.doi.org/10.7827/TurkishStudies>.
- Papadakis, S., & Kalogiannakis, M. (2017). Mobile educational applications for children: what educators and parents need to know. *International Journal of Mobile Learning and Organisation*, 11(3), 256-277. <https://doi.org/10.1504/IJMLO.2017.085338>
- Parmar, D., Isaac, J., Babu, S. V., D'Souza, N., Leonard, A. E., Jörg, S., ... & Daily, S. B. (2016). Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. In *2016 IEEE Virtual Reality (VR)* (pp. 131-140). Institute of Electrical and Electronics Engineers.
- Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, 22, 421-443. <https://doi.org/10.1007/s10639-016-9475-z>
- Peng, J., Sergis, S., Wang, M., & Sampson, D. (2017). Combining smart web-based learning environments with teaching and learning analytics to support reflection on project-based programming education. In *26th International Conference on World Wide Web Companion* (pp. 379–385). International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/3041021.3054158>
- Perçin, F. (2019). Programlama öğretiminde ters yüz öğretim yönteminin öğrencilerin başarılarına, teknoloji tutumlarına ve bireysel yenilikçilik düzeylerine etkisi [Yüksek Lisans Tezi, Marmara Üniversitesi]. Yüksek Öğretim Kurulu Başkanlığı Tez Merkezi.
- Pérez-Marín, D., Hijón-Neira, R., & Martín-Lope, M. (2018). A methodology proposal based on metaphors to teach programming to children. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1), 46-53.
- Pérez-Marin, D., Hijon-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105849, 1-10. <https://doi.org/10.1016/j.chb.2018.12.027>
- Polat, H., & Karabatak, S. (2022). Effect of flipped classroom model on academic achievement, academic satisfaction and general belongingness. *Learning Environments Research*, 25, 159-182. <https://doi.org/10.1007/s10984-021-09355-0>
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and Education*, 128, 365-376. <https://doi.org/10.1016/j.compedu.2018.10.005>

- Pragglejaz Group. (2007). MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1), 1–39. <https://doi.org/10.1080/10926480709336752>
- Prust, C. J., Kelnhofer, R. W., & Petersen, O. G. (2015). The flipped classroom: It's (still) all about engagement. In *2015 ASEE Annual Conference & Exposition* (pp. 26-1534). 10.18260/p.24872
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-24. <https://doi.org/10.1145/3077618>
- Ramazanoğlu, M. (2021). Robotik kodlama uygulamalarının ortaokul öğrencilerinin bilgisayara yönelik tutumlarına ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 25(1), 163-174.
- Rampin, R., & Rapmpin, V. (2021). Taguette: Open-source qualitative data analysis. *Journal of Open Source Software*, 6(68), 3522. <https://doi.org/10.21105/joss.03522>
- Restrepo-Calle, F., Ramírez-Echeverry, J. J., & González, F. A. (2018). Unicode: interactive system for learning and automatic evaluation of computer programming skills. In *Proceedings of EDULEARN18 Conference* (pp. 6888-6898). IATED.
- Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, D. (2018). Decomposition: A k-8 computational thinking learning trajectory. In *2018 ACM Conference on International Computing Education Research* (pp.124-132). Association for Computing Machinery. <https://doi.org/10.1145/3230977.3230979>
- Rodrigues, L., Toda, A. M., Oliveira, W., Palomino, P. T., Avila-Santos, A. P., & Isotani, S. (2021). Gamification works, but how and to whom? an experimental study in the context of programming lessons. In *Proceedings of the 52nd ACM technical symposium on computer science education* (pp. 184-190). ACM. <https://doi.org/10.1145/3408877.3432419>
- Ruiz de Mendoza Ibanez, F. J., & Pérez Hernandez, L. (2011). The contemporary theory of metaphor: Myths, developments and challenges. *Metaphor and symbol*, 26(3), 161-185. <https://doi.org/10.1080/10926488.2011.583189>
- Saban, A. (2004). Giriş düzeyindeki sınıf öğretmenleri adaylarının “öğretmen” kavramına ilişkin ileri sürdükleri metaforlar. *Türk Eğitim Bilimleri Dergisi*, 2(2), 131-155.
- Saez-Lopez, J. M., Roman-Gonzalez, M., & Vazquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers ve Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of research in programming teaching tools: An initial review. *Procedia-Social and Behavioral Sciences*, 103, 127-135. <https://doi.org/10.1016/j.sbspro.2013.10.317>
- Sammur, C. (2010). Concept Learning. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 205-208). Springer.

- Sanford, J. P., Tietz, A., Farooq, S., Guyer, S., & Shapiro, R. B. (2014). Metaphors we teach by. In J. Dougherty, K. Nagel, A. Decker, K. Eiselt (Eds.), *The 45th ACM technical symposium on Computer science education* (pp. 585-590). Association for Computing Machinery. <https://doi.org/10.1145/2538862.2538945>
- Santos, S. C., Tedesco, P. A., Borba, M., & Brito, M. (2020). Innovative approaches in teaching programming: A systematic literature review. In *Proceedings of the 12th International Conference on Computer Supported Education* (pp. 205-214).
- Schez-Sobrinho, S., García, M. A., Gomez, C., Vallejo, D., Molina, A. I., Lacave, C., ..., & Redondo, M. Á. (2019). ANGELA: A novel approach of graphic notation based on the metaphor of road signs to facilitate the learning of programming. In M. A. C. Gonzalez, F. J. R. Sedano, C. F. Llamas, & F. J. Garcia-Penalvo (Eds.), *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 822-829). Association for Computing Machinery. <https://doi.org/10.1145/3362789.3362871>
- Schneider, G. M., & Gersting, J. (2018). *Invitation to computer science* (6th ed.). Cengage Learning.
- Sedgewick, R., Wayne, K., & Dondero, R. (2015). *Introduction to programming in Python: An interdisciplinary approach*. Addison-Wesley Professional.
- Selby, C. C., & Woollard, J. (2013). Computational thinking: The developing definition. Retrieved July 4, 2024 from https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22, 469-495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. In E. Barendsen, P. Hubwieser (Eds.), *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 113-114), Association for Computing Machinery. <https://doi.org/10.1145/3137065.3137084>
- Sentance, S., Dorling, M., McNicol, A., & Crick, T. (2012). Grand challenges for the UK: upskilling teachers to teach computer science within the secondary curriculum. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, (pp. 82-85).
- Seppälä, O. (2012). *Advances in assessment of programming skills* [Doctoral dissertation, Aalto University]. <http://lib.tkk.fi/Diss/2012/isbn9789526047201/isbn9789526047201.pdf>
- Seto, K. (199). Distinguishing metonymy from synecdoche. In K. U. Panther ve G. Radden, (Eds.), *Metonymy in language and thought* (pp.91-120). John Benjamins Publishing.
- Shim, J., Kwon, D., & Lee, W. (2016). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education*, 60(2), 164-172.

- Shin, S., Park, P., & Bae, Y. (2013). The effects of an information-technology gifted program on friendship using scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), 246-249. <http://dx.doi.org/10.7763/IJCCE.2013.V2.181>
- Shuell, T. J. (1990). Teaching and learning as problem solving. *Theory into practice*, 29(2), 102-108. <https://doi.org/10.1080/00405849009543439>
- Shuhidan, S. M. (2012). *Probing the minds of novice programmers through guided learning* [Doctoral dissertation, RMIT University]. <https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/Probing-the-minds-of-novice-programmers-through-guided-learning/9921861240001341>
- Skorczynska, H. (2014). Metaphor and education: Reaching business training goals through multimodal metaphor. *Procedia-Social and Behavioral Sciences*, 116, 2344-2351. <https://doi.org/10.1016/j.sbspro.2014.01.570>
- Smith, K., & Hill, J. (2019). Defining the nature of blended learning through its depiction in current research. *Higher Education Research and Development*, 38(2), 383–397. <https://doi.org/10.1080/07294360.2018.1517732>
- Sontay, G., & Karamustafaoğlu, O. (2022). Harmanlanmış öğrenme modeli üzerine yayınlanan ulusal araştırmaların incelenmesi. *Ulusal Eğitim Akademisi Dergisi*, 6(2), 145-155.
- Sternberg, R. J. (1986). *Critical Thinking: Its Nature, Measurement, and Improvement*. ERIC.
- Strachey, C. (2000). Fundamental concepts in programming languages. *Higher-Order and Symbolic Computation*, 13, 11-49.
- Susanti, W., Jama, J., Krismadinata, Ramadhani, D., Nasution, T. (2021). An overview of the teaching and learning process basic programming in algorithm and programming courses. *Turkish Journal of Computer and Mathematics Education*, 12(2), 2934-2944.
- Şeyihoğlu, A. ve Gencer, G. (2011). Hayat bilgisi öğretiminde “metafor” tekniğinin kullanımı. *Türk Fen Eğitimi Dergisi*, 8(3), 83-100.
- Şimşek, İ. (2018). Dünyada programlama öğretimi. Y. Gülbahar ve H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* (s. 37-65) içinde. Pegem Akademi.
- Tashakkori, A., & Teddlie, C. (2003). *SAGE handbook of mixed methods in social & behavioral research*. Sage. <https://doi.org/10.4135/9781506335193>
- Tay, D. (2016). The nuances of metaphor theory for constructivist psychotherapy. *Journal of constructivist psychology*, 30(2), 165-181. <https://doi.org/10.1080/10720537.2016.1161571>
- TDK (2023). Metafor. <https://sozluk.gov.tr/> adresinden 07.11.2023 tarihinde alınmıştır.
- Thomas, G. P. (2006). Metaphor, students' conceptions of learning and teaching, and metacognition. In P. J. Aubusson (Ed.), *Metaphor and analogy in science education* (pp. 105-117). Springer.

- Thomas, G. P., & McRobbie, C. J. (2001). Using a metaphor for learning to improve students' metacognition in the chemistry classroom. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 38(2), 222-259. [https://doi.org/10.1002/1098-2736\(200102\)38:2%3C222::AID-TEA1004%3E 3.0.CO;2-S](https://doi.org/10.1002/1098-2736(200102)38:2%3C222::AID-TEA1004%3E 3.0.CO;2-S)
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, 162, 1-23. <https://doi.org/10.1016/j.compedu.2020.104083>
- Tomlinson, C. A. (2003). *Fulfilling the promise of the differentiated classroom: Strategies and tools for responsive teaching*. Association for Supervision and Curriculum Development.
- Tsirkunova, S. A. (2013). Conceptual metaphor as a means for teaching critical thinking skills. *International Journal of Humanities and Social Science*, 3(16), 44-48.
- TTKB (2018). Ortaöğretim bilgisayar bilimi dersi (kur 1, kur 2) öğretim programı. <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=335> adresinden 01 Kasım 2021 tarihinde alınmıştır.
- Tuleja, E. A. (2017). Constructing image-based culture definitions using metaphors: Impact of a cross-cultural immersive experience. *Journal of Teaching in International Business*, 28(1), 4-34. <https://doi.org/10.1080/08975930.2017.1316226>
- Türker, P. M., & Pala, F. K. (2020). The effect of algorithm education on students' computer programming self-efficacy perceptions and computational thinking skills. *International Journal of Computer Science Education in Schools*, 3(3), 19-32. <https://doi.org/10.21585/ijcses.v3i3.69>
- Umar, I. N., & Hui, T. H. (2012). Learning style, metaphor and pair programming: Do they influence performance? *Procedia-Social and Behavioral Sciences*, 46, 5603-5609. <https://doi.org/10.1016/j.sbspro.2012.06.482>
- Underwood, B. J. (1966). Some relationship between concept learning and verbal learning. In H. J. Klausmeier & C. W. Harris (Eds.), *Analyses of Concept Learning* (pp. 51-63). Academic Press. <https://doi.org/10.1016/B978-1-4832-3127-3.50009-2>
- Uysal, F. G. (2016). 6. ve 7. sınıf öğrencilerinde kesirler konusunda metafor yardımıyla kavram oluşturma [Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi]. Yüksek Öğretim Kurulu Başkanlığı Tez Merkezi.
- Ülgen, G. (2004). *Kavram Geliştirme: Kuramlar ve Uygulamalar* (3. baskı). Pegem yayıncılık.
- Üzümcü, Ö. ve Bay, E. (2018). Eğitimde yeni 21. yüzyıl becerisi: Bilgi işlemsel düşünme. *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler Dergisi*, 3(2), 1-16.

- Vaníček, J. (2015). Programming in Scratch using inquiry-based approach. In A. Brodník & J. Vahrenhold (Eds.), *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 82-93). Springer. https://doi.org/10.1007/978-3-319-25396-1_8
- Venter, M. (2020). Gamification in STEM programming courses: State of the art. In A. Cardoso, G. R. Alves & M. T. Restivo (Eds.), *IEEE Global Engineering Education Conference* (pp. 859-866).
- Vinayakumar, R., Soman, K., & Menon, P. (2018). Digital storytelling using scratch: engaging children towards digital storytelling. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp.1-6). IEEE.
- Voinohovska, V. (2020). Development of students' computational thinking through creative digital storytelling in block-based programming environment. In L. Gómez Chova, A. López Martínez & I. Candel Torres (Eds.), *EDULEARN20 Proceedings* (pp. 272-278). IATED Academy. <https://doi.org/10.21125/edulearn.2020.0133>
- Wallace, S. (2001). Guardian angels and teachers from hell: Using metaphor as a measure of schools' experiences and expectations of General National Vocational Qualifications. *International Journal of Qualitative Studies in Education*, 14(6), 727-739. <https://doi.org/10.1080/09518390110078404>
- Wang, H. Y., Huang, I., & Hwang, G. J. (2016). Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students. *Journal of Computers in Education*, 3(1), 33-45. <https://doi.org/10.1007/s40692-015-0047-9>
- Wang, T. C., Mei, W. H., Lin, S. L., Chiu, S. K., & Lin, J. M. C. (2009). Teaching programming concepts to high school students with alice. In *Proceedings of the 39th IEEE Frontiers in Education Conference*, (pp. 1-6). <http://dx.doi.org/10.1109/FIE.2009.5350486>
- Wang, Y. (2010). On concept algebra for computing with words (CWW). *International Journal of Semantic Computing*, 4(3), 331-356. <https://doi.org/10.1142/S1793351X10001061>
- Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, 22, 445-468. <https://doi.org/10.1007/s10639-016-9493-x>
- Wegner, E., & Nückles, M. (2015). Training the brain or tending a garden? Students' metaphors of learning predict self-reported learning patterns. *Frontline Learning Research*, 4, 95-109. <http://dx.doi.org/10.14786/flr.v3i4.212>
- Wegner, E., Burkhart, C., Weinhuber, M., & Nückles, M. (2020). What metaphors of learning can (and cannot) tell us about students' learning. *Learning and Individual Differences*, 80, 101884. <https://doi.org/10.1016/j.lindif.2020.101884>

- Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-25. <https://doi.org/10.1145/3089799>
- Wing, J. M. (2006). Computational thinking. *Communications of the Association for Computing Machinery*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
- Yağcı, M. (2019). Valid and reliable tool for examining computational thinking skills. *Education and Information Technologies*, 24, 929-951. <https://doi.org/10.1007/s10639-018-9801-8>
- Yıldırım, A. ve Şimşek, H. (2011). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri* (8. baskı). Seçkin Yayınevi.
- Yıldız Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, 34(6), 740-752. <https://doi.org/10.1111/jcal.12281>
- Yılmaz, Ö. ve Üstün, A. B. (2021). App Inventor ve alternatif blok tabanlı mobil uygulama geliştirme platformlarının karşılaştırmalı incelenmesi. *Disiplinlerarası Eğitim Araştırmaları Dergisi*, 5(9), 1-11.
- Yurdakök, E. A. (2022). *Fiziksel programlama aracı destekli programlama öğretiminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine ve özyeterlik algılarına etkisi* [Yüksek Lisans Tezi, Başkent Üniversitesi]. Yüksek Öğretim Kurulu Başkanlığı Tez Merkezi.
- Yükseltürk, E. ve Altıok, S. (2016). Bilişim teknolojileri öğretmen adaylarının programlama öğretiminde scratch aracının kullanımına ilişkin algıları. *Mersin University Journal of the Faculty of Education*, 12(1), 39-52. <https://doi.org/10.17860/efd.94270>
- Zeren, G. ve Yapıcı, M. (2014). Öğretmen adaylarının renklere ilişkin metaforları. *The Journal of Academic Social Science Studies*, 25(1), 165-175. doi: <http://dx.doi.org/10.9761/JASSS2150>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>
- Zhu, J., & Van Brummelen, J. (2021). Teaching students about conversational ai using convo, a conversational programming agent. In *2021 IEEE symposium on visual languages and human-centric computing* (pp. 1-5). Institute of Electrical and Electronics Engineers.
- Zohrabi, M. (2013). Mixed method research: Instruments, validity, reliability and reporting findings. *Theory and Practice in Language Studies*, 3(2), 254-262.

EKLER

Ek-1. “Pastadan Programa” Ders Planı

Etkinlik Adı	Pastadan Programa
Konu	Programlamanın Temelleri
Kazanımlar	1.3.1.1 Belirli bir problemi çözmek üzere geliştirdiği algoritmayı metin tabanlı programlama aracını kullanarak hatasız bir programa dönüştürür. 1.3.1.3 Yapısal ve söz dizimsel kurallara uygun şekilde programlar geliştirir. 1.3.1.4 Değişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir. 1.3.1.5 Girdilerin belirlendiği ve çıktıların gözlemlendiği programlar geliştirir.
Kavramlar	Programlama Metin tabanlı programlama Programlama Aracı Algoritma Değişken Sabit Girdi Çıktı
Süre	5 ders saati
Durum (Station):	
Öğretmen öğrencilere bir pastacı tarafından yapılan ve pastanın oluşum süreçlerini ayrıntılı biçimde açıklayan bir video izletir. Daha sonra öğrencilere “Sizce içinde bulunduğumuz ‘Programlama’ ünitesinde neden böyle bir video izletmiş olabilirim?” sorusunu yöneltir. Öğrencilerden görüşleri alınır ve öğretmen nedene ilişkin açıklamalarını şimdilik etkinlik sonuna bırakacağını belirterek aşağıdaki senaryoyu anlatmaya başlar. Senaryo aynı zamanda öğrencilerin görebileceği şekilde ekrandan da yansıtılmaktadır. PROGRAMLAMA PASTACILIKTIR Kadir, bu sene uzun zamandır istediği ve pastacılık hakkında yeni şeyler öğrenebileceğini düşündüğü aşçılık okulunu kazanmıştı. İlerde iyi bir pasta ustası olmayı planlıyordu. Ayrıca Ankara’da öğrencilik yıllarını birlikte geçirebileceğini düşündüğü bir de ev arkadaşı bulmuştu. Kadir’in ev arkadaşı olan Onur ise, bilişim teknolojilerini kullanmaya meraklı ve ilerde yazılım mühendisi olmak isteyen azimli bir gençti. Bu iki arkadaş birlikte zaman geçirmekten ve hayalleri doğrultusunda çalışmaktan büyük zevk alıyorlardı. Bir gün Onur eve geldiğinde Kadir’i elinde <i>bir kâğıt (tarif)</i> ve masada onlarca malzeme ile düşünürken buldu. Kadir onu görünce çok sevindi ve Onur’a “<i>Merhaba Onur, tam da sana ihtiyacım olan bir zamanda geldin. Benim de yarınki ödevim olan çikolatalı pastayı yaparken yardım edebilecek birine ihtiyacım vardı</i>” dedi. Onur’un yetiştirmesi gereken bir yazılım ödevi vardı fakat bir saat dinlenmek ve çikolatalı pasta yapımına yardım etmek ona da ilginç gelmişti. Birlikte pasta yapımına başladılar. Onur’un ilk dikkatini çeken Kadir’in elindeki kâğıt oldu. Kadir’e “<i>Elindeki kâğıda bakabilir miyim? Bu kadar sık baktığına göre çok önemli şeyler yazıyor olmalı.</i>” dedi gülümseyerek. Onur kâğıda baktığında iki bölüm gördü. Birinci bölümde <i>Malzemeler</i> ikinci bölümde ise <i>Hazırlanışı</i> yazıyordu (Öğretmen burada çikolatalı pastaya ilişkin bir <i>tarifi</i> öğrencilere gösterir). Kadir öncelikle pastanın hamuru için gerekli malzemeleri kontrol etmek ve doğrulamak için Onur’dan tarifteki malzemeleri okumasını istedi. Onur’un tereyağı ve yumurta ile pek arası olmadığını bildiği için tereyağından 25 gram ve yumurtadan bir tane azaltmıştı. Ne de olsa ödevi daha sonrası afiyetle yiyeceklerdi.	

Malzemeler

- 350 gr bitter çikolata (isteğe göre değişebilir),
- 1 su bardağı un,
- 1 tatlı kaşığı nişasta,
- 1 tatlı kaşığı kabartma tozu,
- 125 gr tereyağı (isteğe göre değişebilir),
- 3 yumurta (isteğe göre değişebilir),
- 1 paket vanilin
- 3 çay bardağı süt.
- 1 su bardağı krema
- 1 tatlı kaşığı toz şeker (isteğe göre değişebilir)

Kadir, Onur'a sence ilk olarak ne yaparak işe başlamalıyız diye sordu? (Bu soru aynı zamanda öğrencilere de sorularak görüşleri alınır). Mutfakla pek arası olmayan Onur "Hepsini bir kaba koyup, karıştırılabilir dedim gülerim". Kadir ise gülümseyerek "Maalesef, pastacılıkta temel kural kâğıtta yazan hazırlanış sırasına uygun davranmaktır. Elimizdeki kâğıtta yer alan tarife uygun hareket etmeliyiz. İşe kâğıtta ilk sırada yazan çikolatayı eritmekle başlamamız gerekiyor" dedi. (Öğretmen pastanın hazırlanışını gösteren tarifi tekrar gösterir).

Hazırlanışı:

- Çikolatayı eritin
- Un, nişasta ve kabartma tozunu bir kaptaki karıştırın
- Tereyağı ve toz şekeri ayrı bir kaptaki çırpın
- Yumurta, vanilin, süt ve erimiş çikolatayı ekleyerek karıştırın
- Un karışımını üzerine ekleyerek karıştırın
- Önceden 220 derecede ısıtılmış fırında 35-40 dakika pişirin
- Fırından çıkan keki üç kat olacak şekilde kesin
- Sıcak kremaya çikolatayı ekleyin
- Her kata çikolata krema karışımından sürün
- Kekin üzerini çikolata ile kaplayın
- 10 dakika sonra buzdolabına koyun ve 2 saat bekleyin.

Kadir ile Onur tarifte yer alan sıralamaya dikkat ederek bir çikolatalı pastayı yaptılar. Son olarak tarifte yazan pasta yapımında ihtiyaç duyulan araçlar kısmı yer almaktaydı. Onur buradaki araçlara göz atarken listede "Fırın, kek kalıbı, çırpıcı, servis tabağı, spatula gibi ifadeler dikkatini çekmişti (Öğretmen araçları gösteren tarifi tekrar gösterir). Kadir "Bunlar genel olarak pasta yapmak için ihtiyacımız olan araçlar" dedi. Daha sonra dağınıklığı toplamaya başladılar ve çikolatalı pastadan birer dilim aldılar...

Senaryoyu anlattıktan sonra öğretmen öğrencilerden senaryoda gördükleri metinlerden altı çizili olanları bir yere not almaları için zaman tanır. Ortaya şu şekilde bir liste çıkması beklenmektedir.

- tarif
- malzemeler
- hazırlanışı
- 350 gr bitter çikolata (isteğe göre değişebilir)
- 125 gr tereyağı (isteğe göre değişebilir),
- 1 tatlı kaşığı toz şeker (isteğe göre değişebilir)
- 3 yumurta
- 1 su bardağı un
- 3 çay bardağı süt
- Maalesef, pastacılıkta temel kural kâğıtta yazan hazırlanış sırasına uygun davranmaktır. Elimizdeki kâğıtta yer alan tarife uygun hareket etmeliyiz.
- "Bunlar genel olarak pasta yapmak için ihtiyacımız olan araçlar"
- Çikolatalı pasta

Öğretmen, öğrencilerden yukarıdaki listeyi incelemelerini ister ve dersin başında sorduğu "Sizce içinde bulunduğumuz programlama ünitesinde neden böyle bir video izletmiş ve böyle bir senaryo anlatmış olabilirim?" sorusunu tekrar sorar ve öğrenci görüşlerini alır.

Gruplama (Grouping)

Etkinlik sonunda öğrencilerden metin tabanlı programlama aracılığıyla hazırlanmış “Kek hazırlama programını” oluşturmaları beklenmektedir. Bu nedenle öğrenciler programlama öğretiminde yaygın olarak kullanılan eşli programlamanın gerekliliklerine uygun olarak ikili gruplara ayrılırlar. Grupları oluştururken öğrencilerin seçiminde programlama becerileri birbirine yakın ve uyum içinde çalışabilecek öğrenciler seçilmeye özen gösterilir. Süreç sonunda grupta yer alan öğrenciler hem bireysel hem de eşleriyle ortak değerlendirmeye tabi tutulacaktır.

Köprü (Bridge)

Öğretmen öncelikle kendisinin yaptığı bir pasta deneyimi ya da pasta anısı varsa onu anlatır. Daha sonra sınıfa “Daha önce çikolata pasta yapan var mı?” diye sorarak öğrencilerden pasta deneyimi olan varsa kısaca deneyimlerini aktarmasını ister. Sonrasında ise öğretmen öğrencilere bir kek hazırlama programı hakkında onlardan beklediği göreve ilişkin aşağıdaki açıklamaları yapar.

Kek Hazırlama Programında sizden beklenenler:

1. Kek yapmak için gerekli malzemeleri ve hazırlanışını araştırınız.
2. Oluşturacağınız programda kullanıcıdan hangi bilgilerin isteneceğini, hangi bilgilerin sabit ve hangilerinin değişebileceğine karar veriniz. Bunu yaparken çikolatalı pasta örneğindeki malzeme listesini ve hazırlanışını inceleyerek yardım alabilirsiniz.
3. Hazırlayacağınız programın algoritma ve akış şemasını hazırlayınız.
4. Kodlama işlemini gerçekleştiriniz.
5. Programdan, çıktı olarak kullanıcı tarafından girilen ve sizin tarafınızdan programda sabit olarak tanımlanmış ürünleri listelemesi ve oluşan kekin nasıl hazırlandığını yazılı olarak göstermesi beklenmektedir.
6. Programlama aracı olarak Python ile gelen IDLE’ı kullanınız.

Öğretmen beklenen programa ilişkin açıklamaları yaptıktan sonra öğrencilerden programda beklenenler listesinde altı çizili olan sözcükleri not almalarını ister. Şu şekilde bir liste oluşması beklenmektedir;

- hangi bilgilerin kullanıcıdan isteneceği,
- hangi bilgilerin sabit, hangilerinin değişebileceği,
- algoritma,
- çıktı,
- Programlama aracı.

Oluşan liste ile daha önce öğretmenin anlattığı senaryodan altı çizili olan liste karşı karşıya getirilir. Öğrencilere aşağıdaki tablo sunularak iki tablo arasındaki olası benzerliklere ilişkin öğrenci görüşleri alınır.

Pastacılığa ilişkin liste	Kek hazırlama programına ilişkin liste
1. Program yazmak pasta yapmaktır! 2. Tarif 3. Malzemeler 4. Hazırlanışı 5. 350 gr bitter çikolata (isteğe göre değişebilir) 6. 1 su bardağı un 7. Maalesef, pastacılıkta temel kural kâğıtta yazan hazırlanış sırasına uygun davranmaktır. Elimizdeki kâğıtta yer alan tarife uygun hareket etmeliyiz. 8. “Bunlar genel olarak pasta yapmak için ihtiyacımız olan araçlar” 9. Çikolatalı pasta	a. Kullanıcıdan hangi bilgilerin isteneceği b. Hangi bilgilerin sabit hangilerinin değişebileceği c. Algoritma d. Çıktı e. Programlama aracı

Sorular (Questions) Öğretmen <i>algoritma, girdi, sabit, değişken, çıktı, programlama aracı</i> temel kavramlarını içeren bir kod örneğini sınıfta sunar ve öğrencilere kod içerisinde bu kavramların yerlerini işaretlemeleri için sorular sorar. Daha sonra öğretmen Kadir’in Onur’a söylediği “<i>Maalesef, pastacılıkta temel kural kâğıtta yazan hazırlanış sırasına uygun davranmaktır. Elimizdeki kâğıtta yer alan tarife uygun hareket etmeliyiz</i>” ifadesi ile programlamada yapısal ve sözdizimsel kurallar arasında ilişkiye dikkat çeker. Öğrencilere aşağıdaki ifadeler verilerek programlama ile pastacılık arasındaki ilişkilere dair görüşleri alınır. • Problemi çözmeyi planladığınız programın <i>tarifini</i> yapar mısınız? (Algoritma ile ilişkilendirilir). • Programda değişken ve sabit tanımlamada bellek yönetimi açısından <i>ölçülü</i> davranmak gerekir. • Sonuca götürecek çıktılar iyi <i>süzülmüş</i> girdilerin ürünüdür. Öğretmen, öğrencilere “Kek hazırlama programı için onlardan beklenenler” listesine göz atmalarını ister. Grup arkadaşlarıyla hazırlayacakları programda bu kavramların anlamlarını tartışmaları ve programlamaya başlamadan önce program hakkında planlama yapmalarını ister. Öğrencilere programı kodlamaları için zaman tanınır. Öğretmen bu sürede öğrencilerden gelecek sorulara rehberlik yapar.
Sergileme (Exhibit) Öğrencilerden hazırladıkları kek hazırlama programını kendi isimleri ile açtıkları klasörlerde (e-gelişim dosyalarında) dönem boyunca saklamaları istenir. Ayrıca her grubun hazırladığı ürün, grupta yer almayan iki öğrenci tarafından Kek Hazırlama Programı Ürün Değerlendirme Formu aracılığıyla değerlendirilir. Öğrenci e-gelişim dosyalarında yer alan ürünler ayrıca dönem sonunda öğrenciler tarafından sunulacaktır.
Yansıtma (Reflections) Öğretmen gönüllü olan öğrencilerden sözel olarak hem etkinlik süreci hem de programı hazırlamada yaşadıkları deneyimler hakkında görüşlerini dile getirmelerini ister. Bu süreçte öğrencilere programlamaya yönelik kavramların ve programlama sürecinin pastacılık ile ilişkilendirilmesine ilişkin görüşleri alınır. Ayrıca öğrencilerin grup içindeki davranışları grupta yer alan öğrenciler tarafından Grup içi Değerlendirme Formu aracılığıyla elde edilecek ve e-gelişim dosyalarına eklenecektir. Öğrencilerden her etkinlik için ayrılan sürenin son saati proje saati olarak ayrılır ve bu etkinlikteki öğrenmelerini proje görevlerine aktarmaları istenir. Proje görevi için öğrencilerden programın algoritma ve akış şemalarını oluşturmaları, programın girdi ve çıktılarına karar vermeleri ve veri girişi kontrolüne gerçekleştirecek kodlamayı yapmaları beklenmektedir.

Kek Hazırlama Programı Ürün Değerlendirme Formu

..... grubu tarafından hazırlanan programı ve oluşturulan algoritmayı aşağıdaki ifadeler açısından inceleyerek uygun gördüğünüz alanları X ile işaretleyiniz.

Algoritma

- () Programın başlangıcı ve bitişi belirtilmiş,
- () Algoritma oluşturmada işlem sırasına dikkat edilmiş,
- () İşlem basamakları numaralandırılmış,
- () Uygun yazım dili kullanılmış.

Program

- () Program dışarıdan veri girişine izin vermektedir.
- () Değişken tanımlanmıştır.
- () Sabitler tanımlanmıştır.
- () Program hatasız şekilde geliştirilmiş.
- () Program doğru sonuç üretmektedir.

Değerlendirmeyi gerçekleştiren öğrenciler;

1.
2.

Grup İçi Değerlendirme Formu		
Bu form, yaptığınız çalışmada ve daha sonraki çalışmalarınızda size rehberlik edebilecek geri bildirimler sunabilmek amacıyla sizden veri toplamayı amaçlamaktadır. Bu nedenle kendiniz ve grup arkadaşınız açısından aşağıdaki ölçütleri dikkate alarak 1-5 arası düzeyde değerlendiriniz. 1- Hiçbir zaman 2- Nadiren 3- Bazen 4- Sık sık 5- Her zaman		
	Ben	Grup Arkadaşım
Grup içinde çalışmaya isteklidir.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Yapılacak etkinliğe hazırlıklı gelir.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Görevini zamanında yerine getirir.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Grup içinde rahatlıkla iletişim kurabilir.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Grup içi görüşlere saygıyla yaklaşır.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Grup içinde uyumlu bir dil kullanır.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Verilen problemi anlamaya çalışır.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Problemin çözümüne etkin katılır.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Problemin çözümüne farklı bakış açıları getirir.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()
Çözümü analiz eder ve yorumlar.	1 () 2 () 3 () 4 () 5 ()	1 () 2 () 3 () 4 () 5 ()

Ek-2. Kavram Haritası Geliştirme ve Değerlendirme (KHGD) Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **çıkma** hakkında sahipsiniz. **Çalışmayı yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Kavram Haritası Geliştirme (Öğrenci)

Sevgili öğrenciler;

Sizden “Programlama” ünitesini kapsayan bir kavram haritası hazırlamanız beklenmektedir. Bunun için öncelikle programlamaya yönelik kavramları hatırlayınız ve listeleyiniz. Daha sonra listelediğiniz kavramları ve olası ilişkilerini düşünerek görselleştiriniz ve varsa örneklerinizi ekleyiniz. Son olarak kavramlar ve örneklerini bağlantı sözcükleri kullanarak ilişkilendiriniz. Oluşturduğunuz kavram haritanız, kavram sayısı, kavramlar arası ilişkiler, bağlantı sözcükleri ve örnekler açılarından değerlendirilecektir. Kavram haritasını tamamlamanız için önerilen süre 30 dakikadır.

Başarılar dilerim.

Erkan GEÇİTLİ
erkangecitli01@gmail.com

Rumuz:.....

Programlama

Kavram Haritası Değerlendirme Formu (Uzman)

“Programlama” ünitesindeki kavramları içerecek biçimde hazırlanan öğrenci kavram haritalarını aşağıdaki her bir boyut altındaki ölçütlerden sadece birini işaretleyerek değerlendiriniz. Puanlamayı “Programlama” ünitesinde yer alan “programlamanın temelleri”, “program kontrolü”, “veri türleri” ve “dosya işlemleri” için ayrı ayrı yapınız ve son sayfada yer alan tabloya not ediniz.		
Programlamanın Temelleri		
Kavram Genişliği (Nicelik)	7 ya da 7’den fazla kavrama yer verme.	4
	5 - 6 arası kavrama yer verme.	3
	3 - 4 arası kavrama yer verme.	2
	1 - 2 arası kavrama yer verme.	1
	Kavrama yer vermeme.	0
Derinlik (Nitelik-İlişkilendirme)	7 ya da 7’den fazla kavramı birbiriyle ilişkilendirme.	4
	5 - 6 arası kavramı birbiriyle ilişkilendirme.	3
	3 - 4 arası kavramı birbiriyle ilişkilendirme.	2
	2 kavramı birbiriyle ilişkilendirme.	1
	İlişkiye yer vermeme.	0
Derinlik (Nitelik-Bağlantı sözcüklerinin kullanımı)	7 ya da 7’den fazla kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	4
	5 - 6 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	3
	3 - 4 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	2
	2 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	1
	Bağlantılı sözcük tanımlamama.	0
Derinlik (Nitelik-örnek kullanımı)	7 ya da 7’den fazla kavramı örnekleme.	4
	5 - 6 arası kavramı örnekleme.	3
	3 - 4 arası kavramı örnekleme.	2
	1 - 2 arası kavramı örnekleme.	1
	Örneğe yer vermeme.	0
Program Kontrolü		
Kavram Genişliği (Nicelik)	4 ya da 4’ten fazla kavrama yer verme.	4
	3 kavrama yer verme.	3
	2 kavrama yer verme.	2
	1 kavrama yer verme.	1
	Kavrama yer vermeme.	0
Derinlik (Nitelik-İlişkilendirme)	5 ya da 5’ten fazla kavramı birbiriyle ilişkilendirme.	4
	4 kavramı birbiriyle ilişkilendirme.	3
	3 kavramı birbiriyle ilişkilendirme.	2
	2 kavramı birbiriyle ilişkilendirme.	1
	İlişkiye yer vermeme.	0
Derinlik (Nitelik-Bağlantı sözcüklerinin kullanımı)	5 ya da 5’ten fazla kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	4
	4 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	3
	3 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	2
	2 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	1
	Bağlantılı sözcük tanımlamama.	0
Derinlik (Nitelik-örnek kullanımı)	4 ya da 4’ten fazla kavramı örnekleme.	4
	3 kavramı örnekleme.	3
	2 kavramı örnekleme.	2
	1 kavramı örnekleme.	1
	Örneğe yer vermeme.	0

Veri Yapıları		
Kavram Genişliği (Nicelik)	6 ya da 6'dan fazla kavrama yer verme.	4
	4 – 5 arası kavrama yer verme.	3
	2 - 3 arası kavrama yer verme.	2
	1 arası kavrama yer verme.	1
	Kavrama yer vermeme.	0
Derinlik (Nitelik-İlişkilendirme)	6 ya da 6'dan fazla kavramı birbiriyle ilişkilendirme.	4
	5 - 6 arası kavramı birbiriyle ilişkilendirme.	3
	3 - 4 arası kavramı birbiriyle ilişkilendirme.	2
	2 kavramı birbiriyle ilişkilendirme.	1
	İlişkiye yer vermeme.	0
Derinlik (Nitelik-Bağlantı sözcüklerinin kullanımı)	7 ya da 7'den fazla kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	4
	5 - 6 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	3
	3 - 4 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	2
	2 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	1
	Bağlantılı sözcük tanımlamama.	0
Derinlik (Nitelik-örnek kullanımı)	7 ya da 7'den fazla kavramı örnekleme.	4
	5 - 6 arası kavramı örnekleme.	3
	3 - 4 arası kavramı örnekleme.	2
	1 - 2 arası kavramı örnekleme.	1
	Örneğe yer vermeme.	0
Dosya İşlemleri		
Kavram Genişliği (Nicelik)	7 ya da 7'den fazla kavrama yer verme.	4
	5 - 6 arası kavrama yer verme.	3
	3 - 4 arası kavrama yer verme.	2
	1 - 2 arası kavrama yer verme.	1
	Kavrama yer vermeme.	0
Derinlik (Nitelik-İlişkilendirme)	7 ya da 7'den fazla kavramı birbiriyle ilişkilendirme.	4
	5 - 6 arası kavramı birbiriyle ilişkilendirme.	3
	3 - 4 arası kavramı birbiriyle ilişkilendirme.	2
	2 kavramı birbiriyle ilişkilendirme.	1
	İlişkiye yer vermeme.	0
Derinlik (Nitelik-Bağlantı sözcüklerinin kullanımı)	7 ya da 7'den fazla kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	4
	5 - 6 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	3
	3 - 4 arası kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	2
	2 kavramı birbiriyle uygun bağlantı sözcüklerini kullanarak tanımlama.	1
	Bağlantılı sözcük tanımlamama.	0
Derinlik (Nitelik-örnek kullanımı)	7 ya da 7'den fazla kavramı örnekleme.	4
	5 - 6 arası kavramı örnekleme.	3
	3 - 4 arası kavramı örnekleme.	2
	1 - 2 arası kavramı örnekleme.	1
	Örneğe yer vermeme.	0
Programlamanın Temelleri Konusuna İlişkin Toplam Puan		
Program Kontrolü Konusuna İlişkin Toplam Puan		
Veri Türleri Konusuna İlişkin Toplam Puan		
Dosya İşlemleri Konusuna İlişkin Toplam Puan		

Ek-3. Şifrelenmiş Mesajlar Oluşturma Uygulaması



Burak, idealist bir doktor olarak bir yandan insanları iyileştirmek için çabalarken diğer yandan da uzun zamandır çocukluk hayali olan onları nasıl ölümsüz yapabileceğine ilişkin araştırmalar yapıyordu. Neredeyse bütün ömrünü buna adanmıştı. Şimdiye kadar insanların ölümsüz olabilmelerini destekleyecek dört madde bulmuştu ancak sonuncuya bir türlü ulaşamıyordu. Umudunu kesmeye başlamıştı. Bu nedenle bulduklarını herkesle paylaşmak ve beşinci maddeyi bulmalarına yardım etmek için onlara bu bilgileri miras olarak bırakmak istiyordu.

Büyük bir emekle elde ettiği bilgileri paylaşacağı doğru kişileri bulmaya kafa yormaya başladı. En sonunda bu bilgiyi kullanacak insanları bulmanın yolunun bu bilgilere ulaşmak için az da olsa çaba sarf etmiş insanlar olmaları gerektiğine karar verdi. İnsanlara bu bilgileri şifrelenmiş olarak sunacaktı. Böylelikle ilk dört maddedeki şifreyi çözmeyi başaranlar beşinci maddeyi de bulmak için de çaba gösterecekti. Bu maddeleri isteyen herkesle aşağıdaki gibi paylaştı:

Ölümsüz olabilmek için;

1. Uçioknok dguöğpöğ
2. Ğaçhröm hiçhuümç
3. Çızıünsğı açöa
4. Mfdfaf trbsrb gfpnz

Burak aynı zamanda bu bulmacayı çözmeye hevesli insanlara bir ipucu bırakmıştı. *“Birinci madde 3, ikinci madde 4, üçüncü madde 5 ve dördüncü madde ise 6 ile ilişkiliydi”*. Bu ipucu kullanılarak ilk dört madde çözülebilirdi. Peki, beşinci madde ne olmalıydı?



Gelin öncelikle bu ilk dört maddenin ne olabileceğini bulmaya çalışarak başlayalım. Ayrıca beşinci maddenin ne olabileceğine yönelik de öneriler oluşturalım. Hatta doktor Burak’ın kullandığı şifreleme sistemini bir bilgisayar programı aracılığıyla herkes tarafından kullanılabilir hale getirelim.



Sizden beklenenler:

1. Doktor Burak’ın ölümsüzlük çalışmaları sonrası bulduğu dört maddenin neler olduğunu belirleyiniz.
2. Ölümsüzlük için gerekli olduğunu düşündüğünüz beşinci madde sizce ne olabilir?
3. Doktor Burak’ın şifrelemede kullandığı şifreleme sistemini Python programlama dilini kullanarak programlarken;
 - a. Programa ait akış şeması ve algoritmayı hazırlayınız.
 - b. Programda gerekli olabilecek veri tanımlamalarını yapınız ve bu veri yapılarına uygun fonksiyonları belirleyiniz.
 - c. Programa dışarıdan veri girişi yapılmasını ve sonucu ekranda gösterilmesini sağlayınız.
 - d. Programdan üretilen verilerin bir dosyaya kaydedilmesini sağlayınız.
 - e. Programda karar kontrol yapıları, döngüler ve fonksiyonları kullanmaya dikkat ediniz.

Hazırladığınız ürünler aşağıda belirtilen Elektronik Ürün Değerlendirme Formu aracılığıyla değerlendirilecektir. Bu nedenle formu dikkatle inceleyiniz.

Ek-4. Elektronik Ürün Geliştirme Süreci Puanlama Yönergesi

1. Algoritma ve Akış Şeması Oluşturma	Puan
Programlama öncesi verilen probleme uygun akış şeması ve algoritma uygun biçimde oluşturulmuş.	4
Programlama öncesi verilen probleme uygun akış şeması ve algoritma oluşturulmuş ancak eksiklikler (sıralama hataları, öge eksikliği, gereksiz basamaklar) içermektedir.	3
Programlama öncesi verilen probleme uygun akış şeması ve algoritmadan yalnızca birisi doğru olacak biçimde oluşturulmuştur.	2
Programlama öncesi verilen probleme uygun akış şeması ve algoritmadan yalnızca birisi oluşturulmuş fakat eksiklikler (sıralama hataları, öge eksikliği, gereksiz basamaklar) içermektedir.	1
Programlama öncesi verilen probleme uygun akış şeması ya da algoritma oluşturulmamıştır.	0
2. Veri Girişi ve Çıkışı	
Program veri giriş / çıkışına olanak sağlamaktadır ve veri girişi kontrolü (boş veri girişi ya da farklı türde veri girişi kontrolleri) yapılmaktadır.	4
Program veri giriş / çıkışına olanak sağlamaktadır ve veri girişi kontrolü sınırlı bir biçimde (boş veri girişi ya da farklı türde veri girişi kontrollerinden birini) yapılmaktadır.	3
Program, veri giriş / çıkışına olanak sağlamaktadır fakat veri girişi kontrolü yapılmamaktadır.	2
Program, veri giriş ya da çıkışından birine olanak sağlamaktadır.	1
Program veri giriş / çıkışına mümkün kılmamaktadır.	0
3. Karar Yapıları	
Programda karar vermede çok yönlü koşul ifadesi kullanılmış ve kodları çalışır durumdadır.	4
Programda karar vermede zincirleme durum ifadesi kullanılmış ve kodları çalışır durumdadır.	3
Programda karar vermede çok yönlü koşul ifadesi kullanılmış fakat kodları yanlış sonuçlar üretmektedir.	2
Programda karar vermede zincirleme durum ifadesi kullanılmış fakat kodları yanlış sonuçlar üretmektedir.	1
Programda karar verme için herhangi bir kod yer almamaktadır.	0
4. Tekrarlı Yapılar	
Programda tekrar eden işlemlerde kullanabileceği döngü çeşidine karar verebilmiş, kodları yazmış ve sonucu ekrana yansıtabilmektedir.	4
Programda tekrar eden işlemlerde kullanabileceği döngü çeşidine karar verebilmiş ve kodları yazmış fakat ekrana yansıtamamaktadır.	3
Programda tekrar eden işlemlerde kullanabileceği döngü çeşidine karar verebilmiş, kodları yazmış ve fakat hatalı sonuç üretmektedir.	2
Programda tekrar eden işlemlerde uygun döngü çeşidine karar verememiş, kodları yazmış ve ekrana yansıtamamıştır.	1
Programda tekrar eden bir yapıya ait bir kod yer almamaktadır.	0
5. Fonksiyonlar	
Programda parametre alan bir fonksiyon tanımlanmış, fonksiyon programın herhangi bir yerinden çağrılabilir ve çalışabilir durumdadır.	4
Programda parametre alan bir fonksiyon tanımlanmış, fonksiyon programın herhangi bir yerinden çağrılabilir fakat yanlış sonuç üretmektedir.	3
Programda parametre alan bir fonksiyon tanımlanmış, fakat programın herhangi bir yerinden çağırılmamaktadır.	2
Programda parametre almayan bir fonksiyon tanımlanmış, programın herhangi bir yerinden çağırılmamaktadır.	1
Programda fonksiyon yer almamaktadır.	0

6. Veri Yapıları	Puan
Amacına uygun veri türü tanımlamaları yapılmış, bu türe ait temel fonksiyonlar kullanılmış ve çalışabilir durumdadır.	4
Amacına uygun veri türü tanımlamaları yapılmış, bu türe ait temel fonksiyonlar kullanılmış fakat yanlış sonuç üretmektedir.	3
Amacına uygun veri türü tanımlamaları yapılmış fakat bu türe ait temel fonksiyonlar kullanılmamıştır.	2
Amaca uygun olmayan bir veri türü tanımlaması yapılmış ve amaca uygun türe ait fonksiyon kullanılmamıştır.	1
Programda veri türü tanımlamaları ve bu türe ait temel fonksiyonlar yer almamaktadır.	0
7. Dosya İşlemleri	
Program harici bir dosyaya veri yazabilmiş, harici bir dosyadan veriyi okuyabilmiş ve ekrana yazdırabilmiştir.	4
Program harici bir dosyaya veri yazabilmiş, harici bir dosyadan veriyi okuyabilmiş ve ekrana yazdıramamıştır.	3
Program harici bir dosyaya veri yazabilmiş, harici bir dosyadan veriyi yanlış okuyabilmiş ve ekrana yazdırmıştır.	2
Program harici bir dosyaya veri yazabilmiş fakat harici dosyadan veriyi okuyamamış ve ekrana yazdıramamıştır.	1
Program harici dosyaya yazma, okuma ve ekrana yazdırma işlemi gerçekleştirememektedir.	0
8. Proje Geliştirme	
Proje sürecinde planlama yapılmış, probleme yönelik çözüm üretilmiş ve süreçte işbirlikli olarak görev alınmıştır.	4
Proje sürecinde planlama yapılmış, probleme yönelik çözüm üretilmiş fakat süreçte işbirlikli olarak görev alınmamıştır.	3
Proje sürecinde planlama yapılmamış, probleme yönelik çözüm üretilmiş ve süreçte işbirlikli olarak görev alınmıştır.	2
Proje sürecinde planlama yapılmış, probleme yönelik çözüm üretilmemiş ve süreçte işbirlikli olarak görev alınmamıştır.	1
Proje sürecinde planlama yapılmamış, probleme yönelik çözüm üretilmemiş ve süreçte işbirlikli olarak görev alınmamıştır.	0

Ek-5. Denel İşlem Öncesi Öğretmen Görüşme Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **cıkma** hakkında sahipsiz. **Çalışmayı yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Değerli meslektaşım;

Bu araştırmanın metin tabanlı programlama öğretiminde yaşanan sorunlara çözüm üretebileceği ve lise düzeyinde metafora dayalı programlama öğretim uygulamalarıyla alanyazına katkıda bulunabileceği umulmaktadır. Sizinle yapılacak görüşmeden elde edilecek verilerle bilgisayar bilimi (Kur-1) dersi “Programlama” ünitesinin var olan öğretim süreçleri betimlenerek söz konusu ünitenin metafora dayalı planlanmasında işe koşulabilecek çalışmaların neler olabileceği kararlaştırılacaktır. Böylelikle “Programlama” ünitesinin öğretimine yönelik var olan öğretim uygulamalarına alternatif denencel bir öğretim planı oluşturulabilecektir. Görüşmeden elde edilen veriler, sadece bu araştırmada kullanılacak ve kişisel bilgiler kesinlikle gizli tutulacaktır. Veri kaybına yol açmamak ve zamanı daha iyi kullanmak için izin verirsiniz görüşmeyi kaydetmek istiyorum. Görüşme sonunda istemediğiniz bazı açıklamalarınızı söylebiliriz. Bu araştırmaya katılmayı kabul ettiğiniz için şimdiden çok teşekkür ederim. Görüşmeye başlamadan önce sormak istediğiniz bir soru varsa, önce bunu yanıtlamak isterim. İzin verirsiniz görüşmeye başlamak istiyorum.

Erkan GEÇİTLİ
erkangecitli01@gmail.com

Görüşme Tarihi ve Saati ____/____/____, Saat: ____

Görüşme Yeri _____

Görüşme Süresi (Başlangıç) _____ / (Bitiş) _____

HAZIRLIK / GİRİŞ SORULARI

1. Kaç yıldır bilgisayar bilimi dersi öğretmenliği yapıyorsunuz?
2. Ne kadar süredir bulunduğunuz okulda öğretmenlik yapıyorsunuz?
3. Lise düzeyinde kaç yıldır metin tabanlı programlama eğitimi veriyorsunuz?

İÇERİĞE YÖNELİK SORULAR

1. “Programlama” ünitesi sonunda öğrencilerinizin hangi nitelikleri kazanmış olmalarını bekliyorsunuz? Örnekler verebilir misiniz?

Sonda:

Değerler açısından

Bireysel gelişim açısından

Beceriler açısından

Kazanımlar açısından

2. Programlama ünitesinin içeriğinde hangi kavramları, temaları / konuları ya da süreçleri ele alıyorsunuz, neden?

3. “Programlama” ünitesinin öğrenme-öğretme süreçlerinde neler yapıyorsunuz?

Alternatif: Programlama ünitesini nasıl işliyorsunuz?

Sonda:

Kavram / konu / süreçlerin öğretimi açısından

Yöntem ve teknikler açısından

Öğretim materyalleri açısından

Rol ve sorumluluklar açısından

Fiziksel ortam açısından

Sınıf dışı etkinlikler açısından

4. “Programlama” ünitesinde hem öğretimin niteliğini hem de öğrencilerinizin kazanımlarını ve düzeyini nasıl belirliyorsunuz?

Alternatif: Programlama ünitesinde nasıl bir ölçme-değerlendirme yaklaşımı kullanıyorsunuz?

5. Bu dönem “Programlama” ünitesinin öğretiminde geçmiş yıllardan farklı olarak yapmak istediğiniz çalışmalar var mı? Varsa, ne tür çalışmalar yapmayı planlıyorsunuz, neden?



Ek-6. Denel İşlem Sonrası Öğretmen Görüşme Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **cıkma** hakkında sahipsiz. **Çalışmayı yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Değerli meslektaşım;

Bu araştırmanın metin tabanlı programlama öğretiminde yaşanan sorunlara çözüm üretebileceği ve lise düzeyinde metafora dayalı programlama öğretim uygulamalarıyla alanyazına katkıda bulunabileceği umulmaktadır. Yapılacak görüşmeden elde edilecek veriler araştırma kapsamında hazırlanan ünite planının nasıl çalıştığını ve çıktılarının oluşma nedenini anlamaya yardımcı olacaktır. Veriler araştırmanın alt problemleri dikkate alınarak analiz edilecek ve araştırmanın rapor edilmesinde kullanılacaktır. Görüşmeden elde edilen veriler, sadece bu çalışmada kullanılacak ve kişisel bilgiler kesinlikle gizli tutulacaktır. Veri kaybına yol açmamak ve zamanı daha iyi kullanmak için izin verirsiniz görüşmeyi kaydetmek istiyorum. Görüşme sonunda istemediğiniz bazı açıklamalarınızı söylebiliriz. Bu araştırmaya katılmayı kabul ettiğiniz için şimdiden çok teşekkür ederim. Görüşmeye başlamadan önce sormak istediğiniz bir soru varsa, önce bunu yanıtlamak isterim. İzin verirsiniz görüşmeye başlamak istiyorum.

Erkan GEÇİTLİ
erkangecitli01@gmail.com

Görüşme Tarihi ve Saati ____/____/____, Saat: ____

Görüşme Yeri _____

Görüşme Süresi (Başlangıç) _____ / (Bitiş) _____

Sorular

1. Sınıfta programlama kavramlarının öğretimine yönelik nasıl bir yol izlediniz?

Sonda:

Kullanılan yöntem/teknik açısından
Kullanılan materyaller açısından
Etkinliklere ayrılan süre açısından
Ölçme-değerlendirme açısından

2. Bilgisayar bilimi (Kur-1) dersinde Programlama ünitesi öncesi üniteler ile “Programlama” ünitesini karşılaştırdığınızda kavram öğretimine yönelik bir farklılığa gittiniz mi? Yanıtınız evet ise nasıl?

Sonda:

Kullanılan yöntem/teknik açısından
Kullanılan materyaller açısından
Etkinliklere ayrılan süre açısından
Ölçme-değerlendirme açısından

3. Programlamaya yönelik kavramları öğretme sürecinizde öğrencilerin ne tür etkinliklere daha fazla katılım gösterdiklerini söyleyebilirsiniz? Neden?
4. Programlamaya yönelik kavramları öğretme sürecinizi düşündüğünüzde öğrencilerin ne tür etkinliklere katılımda zorluk yaşadığını söyleyebilirsiniz? Neden?

5. “Programlama” ünitesi sürecinde gerçekleştirdiğiniz hangi etkinliklerin öğrencilerin programlamaya yönelik tutumlarında bir değişim yarattığını düşünüyorsunuz, nasıl?
Sonda:
Bilişsel açıdan (düşünce ve inançlarında)
Duyuşsal açıdan (sevme, ilgi duyma, zevk alma, sıkılma, korku, kaygı vb.)
Davranış açıdan (yeni programlar yazma, daha uzun vakit geçirme vb.)
6. (Beşinci soruya yanıt evet ise) Gerçekleştirdiğiniz etkinliklerin hangi özelliklerinin programlaya yönelik tutumda olumlu/olumsuz değişime neden olduğunu düşünüyorsunuz?
7. Öğrencilerin elektronik ürün geliştirme süreci sonunda programlama becerilerinde bir değişim olduğunu düşünüyor musunuz? Nedenleri hakkında neler söyleyebilirsiniz?
8. Programlama ünitesinde gerçekleştirilen hangi etkinliklerin öğrencilerin programlama becerilerine katkı sağladığını düşünüyorsunuz? Neden?
Sonda:
Problemi tanımlama
Probleme çözüm üretme
Algoritma ve akış şeması oluşturma
Kodlama
Hataları ayıklama
Programı test etme
Programı değerlendirme aşamaları açılarından.
9. “Programlama” ünitesi süresi boyunca öğrencilerin bilgi işlemsel düşünme becerilerinde bir değişim gözlemlediniz mi? Nasıl?
Sonda:
Problemleri bileşenlerine ayırma
Örüntü tanıma ve genelleme
Soyutlama
Algoritmalar açılarından.
10. Programlama ünitesinde gerçekleştirilen hangi etkinliklerin öğrencilerin bilgi işlemsel düşünme becerilerine katkı sağladığını düşünüyorsunuz? Neden?
11. Uyguladığınız ünite planının güçlü yönleri konusunda neler söyleyebilirsiniz?
12. Uygulanan ünite planının hangi yönlerden iyileştirilmesi gerektiğini düşünüyorsunuz? Önerileriniz nelerdir?

Ek-7. Denel İşlem Sonrası Öğrenci Görüşme Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **cıkma** hakkında sahipsiz. **Çalışmayı yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Sevgili öğrenci;

Bu araştırmanın metin tabanlı programlama öğretiminde yaşanan sorunlara çözüm üretebileceği ve lise düzeyinde metafora dayalı programlama öğretim uygulamalarıyla alanyazına katkıda bulunabileceği umulmaktadır. Bu nedenle sizinle yapacağım görüşmede amacım araştırma kapsamında hazırlanan ünite planının nasıl çalıştığını ve çıktıların oluşma nedenini anlamaya çalışmaktır. Elde edilen veriler araştırmanın alt problemleri dikkate alınarak analiz edilecek ve araştırmanın rapor edilmesinde kullanılacaktır. Veri kaybına yol açmamak ve zamanı daha iyi kullanmak için izin verirseniz görüşmeyi kaydetmek istiyorum. Görüşme sonunda istemediğiniz bazı açıklamalarınızı söylebiliriz. Bu araştırmaya katılmayı kabul ettiğiniz için şimdiden çok teşekkür ederim. Görüşmeye başlamadan önce sormak istediğiniz bir soru varsa, önce bunu yanıtlamak isterim. İzin verirseniz görüşmeye başlamak istiyorum.

Erkan GEÇİTLİ
erkangecitli01@gmail.com

Görüşme Tarihi ve Saati ____/____/____, Saat: ____

Görüşme Yeri _____

Görüşme Süresi (Başlangıç) _____ / (Bitiş) _____

Hazırlık Soruları

1. Bilgisayar bilimi (Kur-1) dersi haricinde bilişim teknolojilerine yönelik bir ders aldınız mı? Evet ise, açıkla mısınız?

Sonda:

Hangi ders?

Ne zaman?

Ne kadar süre?

2. Bilgisayar bilimi (Kur-1) dersi haricinde programlama aracılığıyla bir ürün geliştirme deneyiminiz oldu mu? Evet ise, ürünü geliştirme sürecini açıkla mısınız?

İçeriğe Yönelik Sorular

3. Bu ünite boyunca programlama kavramlarını öğrenmeniz için gerçekleştirilen etkinlikleri nasıl değerlendiriyorsunuz?

Sonda:

Yöntem/teknik açısından

Öğretim materyalleri açısından

Etkinliklere ayrılan süre açısından

Derse etkin katılım açısından

Rol ve sorumluluklarınız açısından

4. Programlamaya yönelik kavramları öğrenme sürecinizde gerçekleştirilen etkinliklerden hangilerinden hoşlandınız/ beğendiniz? Neden?

5. Programlamaya yönelik kavramları öğrenme sürecinizi düşündüğünüzde bu süreçte sizce değişmesi gereken şey/şeyler var mıdır? Neleri değiştirmek isterdiniz?
6. Bilgisayar bilimi (Kur-1) dersinde “Programlama” ünitesi haricindeki ünitelerdeki kavram öğrenme sürecinizle “Programlama” ünitesinde kavram öğrenme süreciniz arasında bir farklılık gözlemlediniz mi? Yanıtınız evet ise nasıl?
- Sonda:**
Kullanılan yöntem/teknik açısından
Kullanılan materyaller açısından
Etkinliklere ayrılan süre açısından
Derse etkin katılım açısından
7. Elektronik ürün oluşturmanız için size verilen senaryoda yer alan problemin çözümünde nasıl bir yol izlediniz?
- Sonda:**
Problemi bileşenlerine ayırma
Örüntü tanımlama ve genelleme
Soyutlama
Algoritma açılarından.
8. Geliştirdiğiniz elektronik ürünü programlama sürecinizde neler yaptınız/yaşadınız?
- Sonda:**
Problemi tanımlama açısından
Probleme çözüm üretme açısından
Algoritma ve akış şeması oluşturma
Kodlama
Hataları ayıklama
Programı test etme
Programı değerlendirme açılarından.
9. Bir bilgisayar programı ya da mobil uygulamayı kullanırken nasıl geliştirildiğini merak eder misiniz? Neden?
- Sonda:**
Kimlerin geliştirdiğini,
Teknik özelliklerini,
Nasıl bir süreçle oluşturulduğunu,
Hangi programlama dilini kullandıklarını,
Nasıl yaygınlaştırıldığını vb.
10. Programlama aracılığıyla “Programlama” ünitesinde geliştirdiğiniz elektronik üründen farklı olarak yeni bir ürün geliştirmek ister miydiniz? Neden?
11. Çok yakın bir arkadaşına programlama sürecinde hissettiklerini anlatsan neler söyledin?
12. Programlama ünitesinde gerçekleştirilen etkinliklerin programlama becerinize katkısının olduğunu düşünüyor musunuz? Yanıtınız evet ise nasıl katkısı oldu?
- Sonda:**
Problemi tanımlama açısından
Probleme çözüm üretme açısından
Algoritma ve akış şeması oluşturma
Kodlama
Hataları ayıklama
Programı test etme
Programı değerlendirme açılarından.

Ek-8. Gözlem Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **cıkma** hakkında sahipsiz. **Çalışmayı** **yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Gözlemin amacı: Metafora dayalı programlama öğretiminin kavram öğrenme, programlamaya yönelik tutum, programlama ve bilgi işlemsel düşünme boyutlarına etkilerini betimlemektir.

Araştırma alt problemleri

1. Metin tabanlı programlama öğretiminde metafor kullanımı, programlama kavramlarını öğrenmeyi nasıl etkilemektedir?
2. Metin tabanlı programlama öğretiminde metafor kullanımının programlama becerilerine etkisi nedir?
3. Metin tabanlı programlama öğretiminde metafor kullanımının bilgi işlemsel düşünme becerilerine etkisi nedir?

Veri Toplama: Deney ve kontrol grubunda yer alan öğrencilere aşağıda belirtilen boyutlar açısından toplam 36 ders saati gözlem yapılacaktır (18 ders saati deney ve 18 ders saati kontrol grubu). Gözlem yapılacak şubeler her hafta farklılaşacak ve deney ve kontrol grubunda yer alan her şubede denel işlemin başında, ortasında ve sonunda en az birer gözlem gerçekleştirilmiş olacaktır. Gözlem notları öğretmenlerin dersi planlama biçimleri dikkate alınarak tutulacaktır.

Kavram öğrenme: Öğrencilerin kavram öğrenme sürecinde kavramı edinme biçimlerine, kavramı öğrenme düzeylerine, kavramın özelliklerine, kavram haritalama, kavram kazanma ve kavram oluşturma süreçlerine, kavrama ait ön bilgilerine ve kavram kargaşasına yönelik veriler.

Programlama becerileri: Öğrencilerin programlama sürecini oluşturan öğeler (planlama, uygulama ve değerlendirme) açısından problem çözme yaklaşımı, algoritma oluşturma, kodlama, çalıştırma, test etme ve değerlendirmelerine yönelik veriler.

Bilgi işlemsel düşünme becerisi: Öğrencilerin bir problemin çözümünde bilişim teknolojileri araçlarını kullanma ve strateji geliştirme süreçlerinde problemi formüle etmeleri, elde edilen verileri analiz ve organize etmeleri, veriyi soyutlamaları, algoritmik düşünme ve çözümü otomatikleştirmeleri, çözümü tanımlamaları, çözümünü analiz etmeleri ve uygulamaları, başka problemlere genellemeleri ve transfer etmelerine yönelik veriler.

Metafor Kullanımı: Öğrenme ortamında öğrencilerin ve öğretmenlerin sınıfta kullandıkları metaforların türü (yapısal, yönelim ve ontolojik) ve bileşenlerine (hedef, kaynak ilişki) yönelik veriler.

Gözlem verilerinin kodlamasında kullanılacak geçici kod listesi

Kavram öğrenme:

- Edinme biçimine göre
 - Tecrübeyle ilgili
 - Metaforik kavramlar
- Öğrenme düzeyi
 - Somut
 - Tanıma
 - Sınıflama
 - Soyut
- Kavramın özellikleri
 - Ortak özellikler
 - Farklı anlamlar
 - Yeniden tanımlama
 - Bireyden bireye değişim

- Kavramın orijinali
 - Soyut/somut kavram
- Kavram kazanma
- Kavram öğrenme koşulları
 - Zaman
 - Kavram geliştirme stratejisi
 - Bilgiyi seçme
 - Bilgiyi merkeze alma
 - Dikkat ve odaklaşma
 - Uyarıcıların seçimi ve organizasyonu
 - Olumlu örnekler
 - Olumsuz örnekler
- Genelleme
- Ayırt etme
- Ön bilgiler
- Kavram kargaşası

Programlama becerisi

- Planlama
 - Problem çözme yaklaşımı
 - Analiz çizelgesi
 - Etkileşim çizelgesi
 - GSÇ çizelgesi
 - Algoritma
 - Akış şeması
- Uygulama
 - Algoritma / Akış şeması oluşturma
 - Kod yazma
 - Veri Giriş/Çıkışı
 - Karar yapıları
 - Tekrarlı yapılar
 - Fonksiyonlar
 - Dosya işlemleri
 - Programı çalıştırma
- Değerlendirme
 - Program test etme
 - Program hatası
 - Çözümü değerlendirme

Bilgi işlemsel düşünme becerisi

- Problemi Formüle etme
- Veri analizi
- Veri organizasyonu
- Veriyi soyutlama
- Algoritmik düşünme
- Çözümü otomatikleştirme
- Çözümü tanımlama
- Çözümü uygulama
- Başka problemlere genelleme
- Transfer etme

Metafor kullanımı

- Türü
 - Yapısal(taşıyıcı) metaforlar
 - Yönelim metaforları
 - Ontolojik metaforlar
- Bileşenler açısından
 - Hedef
 - Kaynak
 - Semantik özellikler (ilişki)

Gözlem notlarının kaydedilmesinde yararlanılan çapraz tablo

	Gözlem no	
	Gözlem Tarihi	
	Gözlem Yapılan Süre	
	Gözlem Yapılan Yer	
Dersin Aşamaları	Açıklayıcı Notlar	Yansıtıcı Notlar
Durum (Situation)		
Gruplama (Grouping)		
Köprü (Bridge)		
Sorular (Questions)		
Sergileme (Exhibit)		
Yansıtma (Reflections)		

Ek-9. Süreç Değerlendirme Formu

Sizi Erkan GEÇİTLİ tarafından yürütülen “Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması” başlıklı **araştırmaya** davet ediyoruz. Bu araştırmaya katılıp katılmama kararını vermeden önce, araştırmanın neden ve nasıl yapılacağını bilmeniz gerekmektedir. Bu nedenle bu formun okunup anlaşılması büyük önem taşımaktadır. Eğer anlayamadığınız ve sizin için açık olmayan şeyler varsa, ya da daha fazla bilgi isterseniz bana sorunuz.

Bu çalışmaya katılmak tamamen **gönüllülük** esasına dayanmaktadır. Çalışmaya **katılmama** veya katıldıktan sonra herhangi bir anda çalışmadan **çıkma** hakkında sahipsiz. **Çalışmayı yanıtlamanız, araştırmaya katılım için onam verdiğiniz** biçiminde yorumlanacaktır. Size verilen **formlardaki** soruları yanıtlarken kimsenin baskısı veya telkini altında olmayın. Bu formlardan elde edilecek kişisel bilgiler tamamen gizli tutulacak ve yalnızca araştırma amacı ile kullanılacaktır.

Ders:	Bilgisayar bilimi (Kur-1)
Ünite:	Programlama
Konu:	
Tarih:	

Bu konuda yer alan programlama kavramlarını öğrenmenizde hangi etkinliklerin etkisi olduğunu düşünüyorsunuz, neden? Olumlu etkisi olan etkinlikler ve neden(ler)i? Olumsuz etkisi olan etkinlikler ve neden(ler)i?
Hangi etkinliklerin programlamaya yönelik tutumunuzu etkilediğini düşünüyorsunuz? Neden? Olumlu etkisi olan etkinlikler ve neden(ler)i? Olumsuz etkisi olan etkinlikler ve neden(ler)i?
Gerçekleştirilen etkinliklerin sizin daha iyi programlar oluşturmaya katkı sağladı mı? Nasıl?
Size verilen senaryodaki probleme çözüm üretme konusunda hangi aşamadasınız ve problemin çözümü için nasıl yol izliyorsunuz?

Ek-10. İl Milli Eğitim Müdürlüğünden Alınan Araştırma İzni



T.C.
MANİSA VALİLİĞİ
İl Milli Eğitim Müdürlüğü

Sayı : E-46949512-605.01-23335100
Konu : Araştırma İzni

31.03.2021

EGE ÜNİVERSİTESİ REKTÖRLÜĞÜNE

İlgi: a) Milli Eğitim Bakanlığı Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarih ve 1563890 sayılı 2020 / 2 No'lu Genelgesi,
b) 16.03.2021 tarih ve 88884 sayılı yazınız.

Üniversiteniz Eğitim Bilimleri Enstitüsü Eğitim Bilimleri Anabilim Dalı Eğitim Programları ve Öğretim Bilim Dalı Doktora Programı öğrencisi Erkan GEÇİTLİ'nin "Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması" konu başlıklı tez çalışması kapsamında Yunusemre İlçe Milli Eğitim Müdürlüğüne bağlı Fatih Anadolu Lisesi ve Dündar Çiloğlu Anadolu Lisesi 9. sınıf öğrencileri ile Bilişim dersi öğretmenlerine yönelik yapmak istediği anket çalışması izni ile ilgili olarak, Müdürlük Makamından alınan 26.03.2021 tarih ve 23147864 sayılı onay yazısı, yazımız ekindedir.

Söz konusu çalışmanın; 2020 - 2021 eğitim öğretim yılında, araştırmanın yüz yüze eğitim-öğretime ara verilmesi göz önüne alınarak örgün eğitimin tam olarak başlamasıyla birlikte ilgili müdürlüğün izni ile denetimi ilçe milli eğitim müdürlüğü ve okul/kurum idaresinde olmak üzere, yazımız ekinde bulunan onaylı bir örneği Müdürlüğümüzde muhafaza edilen veri toplama araçlarının kullanılması koşuluyla, kurum faaliyetlerini aksatmadan, gönüllülük esasına dayalı olarak uygulanması Müdürlüğümüzce uygun görülmüştür.

Bilgilerinizi ve araştırma tamamlanmasından itibaren en geç 30 gün içerisinde araştırma sonucunu içeren bir kitap ve iki adet CD'nin Müdürlüğümüz Strateji Şubesine teslim edilmesini arz ederim.

Mustafa DİKİCİ
İl Milli Eğitim Müdürü

Ekler:
Onay Yazısı (1 sayfa)
Ölçekler (5 sayfa)

Bu belge güvenli elektronik imza ile imzalanmıştır.

Adres : Nişancıpaşa Mh. Atatürk Blv. No:36/A Şehzadeler/MANİSA

Belge Doğrulama Adresi : <https://www.turkiye.gov.tr/meb-ebys>

Bilgi için: AR-GE Birimi T.ATLI (Dahili: 105)

Telefon No : 0 (236) 231 46 08

Uyvan : Şef

E-Posta: strateji45@meb.gov.tr

İnternet Adresi: manisa.meb.gov.tr

Faks:2362311251

Kep Adresi : meb@hs01.kep.tr

Bu evrak güvenli elektronik imza ile imzalanmıştır. <https://evraksorgu.meb.gov.tr> adresinden b7f40-558e-336b-b69b-5c5a kodu ile teyit edilebilir.



Ek-11. Etik Kurul Onayı



EGE ÜNİVERSİTESİ SOSYAL VE BEŞERİ BİLİMLER BİLİMSEL ARAŞTIRMA VE YAYIN ETİĞİ KURULU

KARAR BELGESİ

Ek-2

YÜRÜTÜCÜNÜN ADI SOYADI / KURUMU	Doç. Dr. Bünyamin YURDAKUL / Eğitim Fakültesi
DANIŞMANIN ADI SOYADI / KURUMU	-
DİĞER ARAŞTIRMACILAR	Erkan GEÇİTLİ / Eğitim Bilimleri Enstitüsü
ARAŞTIRMANIN TÜRÜ	☐ Yüksek Lisans Tezi ☒ Doktora Tezi ☐ Özgün Araştırma
ARAŞTIRMANIN BAŞLIĞI	Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması
BİLİRKİŞİ GÖRÜŞÜ	Yok
KARARIN ALINDIĞI TOPLANTI TARİHİ	03.02.2021
TOPLANTI / KARAR SAYISI	03 / 06
	PROTOKOL NO: 786
KARAR	Araştırma OYBİRLİĞİ ile etik açıdan uygun bulunmuştur.

Prof. Dr. Mehmet ERSAN

Kurul Başkanı

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Hülya YILMAZ
Kurul Başkan Yardımcısı

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Fazlı GÖKÇEK
Kurul Üyesi

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Sonia AMADO
Kurul Üyesi

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Mustafa MUTLUER
Kurul Üyesi

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Metin KARADAĞ
Kurul Üyesi

Elektronik onaylıdır (online toplantı yolu ile)

Prof. Dr. Nadim MACİT
Kurul Üyesi

Elektronik onaylıdır (online toplantı yolu ile)

**Ek-12. Araştırmanın 2021-2022 Eğitim-Öğretim Yılında da Uygulanabileceğine
Dair İl Milli Eğitim Müdürlüğü İzni**



T.C.
MANİSA VALİLİĞİ
İl Milli Eğitim Müdürlüğü

Sayı : E-46949512-605.01-33089136
Konu : Araştırma İzni

27.09.2021

MÜDÜRLÜK MAKAMINA

- İlgi: a) Milli Eğitim Bakanlığı Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarih ve 1563890 sayılı 2020 / 2 No'lu Genelgesi,
b) Ege Üniversitesi Rektörlüğü Öğrenci İşleri Daire Başkanlığı Diploma Hizmetleri Şube Müdürlüğünün 16.03.2021 tarih ve 88884 sayılı sayılı yazısı.
c) Ege Üniversitesi Rektörlüğü Öğrenci İşleri Daire Başkanlığı Diploma Hizmetleri Şube Müdürlüğünün 17.09.2021 tarih ve 321914 sayılı yazısı.

İlgi (b) yazı da belirtilen; Ege Üniversitesi Eğitim Bilimleri Enstitüsü Eğitim Bilimleri Anabilim Dalı, Eğitim Programları ve Öğretim Bilim Dalı Doktora Programı öğrencisi Erkan GEÇİTLİ'nin "Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması" konu başlıklı çalışması kapsamında, Yunussemre İlçe Milli Eğitim Müdürlüğüne bağlı Fatih Anadolu Lisesi ve Dünder Çiloğlu Anadolu Lisesi 9. sınıf öğrencileri ile Bilişim dersi öğretmenlerine yönelik 2020-2021 eğitim-öğretim döneminde yapmak istediği araştırma, 26.03.2021 tarih ve 23147864 sayılı Makam Oluru ile uygun görülmüştür.

İlgi (c) yazı ile 2021-2022 eğitim-öğretim yılında da uygulanması istenen araştırma, ilgi (a) Genelgenin 23. maddesi doğrultusunda uygulanması Müdürlüğümüzce uygun görülmektedir.

Makamlarınızca da uygun görülmesi halinde olurlarınıza arz ederim.

Coşkun DEMİR
Şube Müdürü

OLUR

Mustafa DİKİCİ
İl Milli Eğitim Müdürü

Bu belge güvenli elektronik imza ile imzalanmıştır.

Adres : Nisancıpaşa Mh. Atatürk Biv. No:36/A Şehzadeler/MANİSA

Belge Doğrulama Adresi : <https://www.turkiye.gov.tr/meb-ebys>

Telefon No : 0 (236) 231 46 08

E-Posta: istatistik45@meb.gov.tr

Kep Adresi : meb@hs01.kep.tr

Bilgi için: Strateji Geliştirme - C.Sunay BULUT

İnternet Adresi: www.manisa.meb.gov.tr

Unvan : Memur

Faks: 2362311251

Bu güvenli elektronik imza ile imzalanmıştır. <https://www.kocim.meb.gov.tr> adresinden 4e3b-69c7-3ab4-85a9-bb65 kodu ile teyit edilebilir.



Ek-13. Araştırmada Gerçekleşen Değişikliklere Yönelik Etik Kurul Onayı

Ege Üniv. Evrak Tarih ve Sayısı: 29.11.2021-E.432225



T.C.
EGE ÜNİVERSİTESİ REKTÖRLÜĞÜ
Eğitim Bilimleri Enstitüsü Müdürlüğü
Enstitü Sekreterliği

İVEDİ

Sayı : E-10705136-050.06.04-432225
Konu : Kararlar

EĞİTİM BİLİMLERİ ANABİLİM DALI BAŞKANLIĞINA

İlgi : 25.11.2021 tarihli ve 430787 sayılı yazı.

Ege Üniversitesi Rektörlüğü Hukuk Müşavirliği Sosyal ve Beşeri Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulu Anabilim Dalının 25.11.2021-E.43087 sayılı yazısına istinaden, Anabilim Dalımız Eğitim Programları ve Öğretim Bilim Dalı Doktora Programı 94180000002 numaralı öğrencisi Erkan GEÇİTLİ'nin "Metin Tabanlı Programlama Öğretiminde Metafor Kullanımının Etkileri: Bir Karma Yöntem Araştırması" başlıklı tez çalışması kapsamında yapılan değişiklikler ve ekleri incelenmiş olup değişiklik talebi Sosyal ve Beşeri Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulu tarafından oybirliği ile etik açıdan uygun bulunmuştur.

Söz konusu durumun ivedi bir şekilde danışman ve öğrenciye bildirilmesi hususunda bilgelerinizi ve gereğini rica ederim.

Prof. Dr. Hakan ATILGAN
Müdür V.

Bu belge, güvenli elektronik imza ile imzalanmıştır.

Belge Doğrulama Kodu : BSN11L1Y1Z

Belge Takip Adresi : <https://www.turkiye.gov.tr/ege-universitesi-ebys>

Adres: Ege Üniversitesi Rektörlüğü Gençlik Cad. No:12 35040 Bornova/İzmir

Telefon: +90 (232) 311 21 10 Faks: +90 (232) 339 90 90

Web: www.ege.edu.tr

E-posta Adresi: egeuniversitesi@egeuniversitesi.hr03.ksp.tr

Bilgi için: İlkay Tuzioğlu

Unvanı: Büro Personeli



Bu belge, güvenli elektronik imza ile imzalanmıştır.