

**T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DENEYSEL ÇALIŞMA SONUÇLARININ SİMULE EDİLMESİ VE
DEĞERLENDİRİLMESİ İÇİN YAZILIM GELİŞTİRME: İNŞAAT
MÜHENDİSLİĞİ UYGULAMASI**

Nurgün GENÇEL

**Danışman
Yrd. Doç. Dr. Kubilay TAŞDELEN**

**YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
ISPARTA – 2015**

© 2015 [Nurgün GENÇEL]

TEZ ONAYI

Nurgün GENÇEL tarafından hazırlanan “Deneysel çalışma sonuçlarının simule edilmesi ve değerlendirilmesi için yazılım geliştirme: İnşaat Mühendisliği Uygulaması” adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Elektronik Bilgisayar Eğitimi Anabilim Dalı**’nda **YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

Danışman	Yrd. Doç. Dr. Kubilay TAŞDELEN Süleyman Demirel Üniversitesi	
Jüri Üyesi	Doç. Dr. Cengiz ÖZEL Süleyman Demirel Üniversitesi	
Jüri Üyesi	Yrd. Doç. Dr. Muharrem İLKUÇAR Mehmet Akif Ersoy Üniversitesi	

Enstitü Müdürü	Prof.Dr.Ahmet ŞAHİNER	
-----------------------	------------------------------	-------

TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Nurgün GENÇEL

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR.....	v
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ	2
3. MATERYAL VE YÖNTEM	5
3.1. Visual Studio	5
3.2. Kullanılan Malzeme.....	8
3.2.1. Çimento	8
3.2.2. Agregat	9
3.2.3. Silis Dumanı.....	9
3.2.4. Süper Akışkanlaştırıcı.....	10
3.2.5. Çelik Tel.....	11
3.3. Beton Karışımları.....	11
3.4. Yazılım Tasarımı	12
4. ARAŞTIRMA BULGULARI VE TARTIŞMA.....	18
4.1. Yazılım Çalışma İşlem Sırası ve Akış Diagramı	18
4.2. Yazılımın Akış Diagramı.....	21
4.3. Yazılımın Çalıştırılması.....	26
5. SONUÇ VE ÖNERİLER	28
KAYNAKLAR	30
EKLER.....	32
EK A. Yük ve Deformasyon Verileri.....	32
EK B. Yazılım Kodları	39
ÖZGEÇMİŞ.....	68

ÖZET

Yüksek Lisans Tezi

DENEYSEL ÇALIŞMA SONUÇLARININ SİMULE EDİLMESİ VE DEĞERLENDİRİLMESİ İÇİN YAZILIM GELİŞTİRME: İNŞAAT MÜHENDİSLİĞİ UYGULAMASI

Nurgün GENÇEL

**Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Bilgisayar Eğitimi Anabilim Dalı**

Danışman: Yrd. Doç. Dr. Kubilay TAŞDELEN

Günümüzde bilgisayar ve yazılım teknolojisinin gelişmesiyle hayatımızın her alanında çeşitli yazılımları kullanmaktayız. Yeni geliştirilen her teknoloji gibi yazılım teknolojisininde gelişim sürecindeki ilk adımı ihtiyaç analizdir.

Bu tez çalışmasında inşaat alanında ihtiyaç analizi yapılarak yazılım teknolojisi ile daha hızlı çalışılabilecek olan bir dalı saptanmıştır. Betonun kırılma mekaniği depreme dayanıklı binalar, savunma amaçlı yapılar, reaktörler, iklim ve çevre koşullarının etkisinde yıpranmaya maruz binalar gibi ekonomik açıdan maliyeti yüksek yapılar için oldukça büyük bir öneme sahiptir. Betonun kırılma enerjisi, beton yapıların göçmesinin anlaşılması için gerekli önemli bir kırılma karakteristiğidir. Ancak betonun kırılma enerjisini hesaplamak kolay değildir.

Çalışmada deformasyon kontrollü test cihazından gerekli dataları aldıktan sonra bu dataları istenilen parametrelere göre işleyen ve sonuçları grafik ve metin olarak veren bir yazılım hazırlandı. Bu yazılım sayesinde zaman tasarrufu sağlanarak daha sağlıklı sonuçlar elde edildi.

Anahtar Kelimeler: C#, kırılma enerjisi, beton, yazılım

2015, 63 sayfa

ABSTRACT

M.Sc. Thesis

SOFTWARE DEVELOPMENT FOR SIMULATION AND EVALUATION OF EXPERIMENTAL STUDY RESULTS: CIVIL ENGINEERING APPLICATION

Nurgün GENÇEL

**Süleyman Demirel University
Graduate School of Applied and Natural Sciences
Department of Electronics-Computer Education**

Supervisor: Asst. Prof. Dr. Kubilay TAŞDELEN

Today, we have used several softwares in our life by development in computers and software technologies. The first step is the analysis of the needs for developing software technologies as it is in new developed technologies.

In this study, it is detected how faster progressed can be obtained by software technology after the need analysis in construction sector. Fracture mechanic of concrete is highly important for structures such as earthquake resistant structures, military facilities, nuclear power plants and structures in severe environment that cost high. Fracture energy of concrete is an important characteristic of concrete for understanding collapse of structures. But, it is not easy to calculate fracture energy of concrete.

The aim of this study is to write a software processing data obtained from deformed controlled machine according to parameters desired by user and then presenting results in text and graphic. By this software, time will be saved and accurately results will be obtained.

Keywords: C#, Fracture energy, concrete, software

2015, 63 pages

TEŞEKKÜR

Çalışmalarımın her aşamasında bilgi birikimiyle beni yönlendiren danışmanım Yrd. Doç. Dr. Kubilay Taşdelen'e ve Süleyman Demirel Üniversitesi öğretim üyesi Doç. Dr. Cengiz Özel'e sonsuz teşekkürlerimi sunarım.

Çalışmamın uygulama kısmında yardımlarını esirgemeyen Bozok Üniversitesi İnşaat Mühendisliği Bölümüne teşekkürlerimi sunarım.

Tezimin her aşamasında beni yalnız bırakmayan aileme eşime sevgi ve saygılarımı sunarım.

Nurgün GENÇEL
ISPARTA, 2015

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Visual studio çalışma alanı.....	8
Şekil 3.2. Çatlak genişliğini ölçmek için kullanılan düzenek.....	11
Şekil 3.3. Kiriş numunelerin kırıldığı eğilme çerçevesi.....	12
Şekil 3.4. Yazılım geliştirme süreci	13
Şekil 3.5. Betonun eğilme yüklemesi altındaki davranışı	14
Şekil 3.6. Kirilis yazılımında kullanılan anasayfa ve kod sayfaları	16
Şekil 3.7. Kirilis yazılımı geliştirme ortamı.....	17
Şekil 4.1. Yük-deformasyon eğrisi.....	18
Şekil 4.2. Yük-CMOD ve FL diyagramları	23
Şekil 4.3. Akış diagramı birinci bölüm	21
Şekil 4.4. Akış diagramı ikinci bölüm	22
Şekil 4.5. Kirilis yazılımı ekran görüntüsü	24
Şekil 4.6. Kirilis yazılımı kontrol bölümü ekran görüntüsü	24
Şekil 4.7. Giriş parametreleri ekran görüntüsü	25
Şekil 4.8. Kirilis yazılımı mesajlar bölümü ekran görüntüsü.....	25
Şekil 4.9. Kirilis yazılımı sonuç değerleri bölümü ekran görüntüsü.....	25
Şekil 4.10. Kirilis yazılımına veri yükleme anı görüntüsü	26
Şekil 4.11. Kirilis yazılımına değerleri ekleme anı görüntüsü.....	26
Şekil 4.12. Kirilis yazılımına sonuç değerleri alma görüntüsü.....	27

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1. Portland çimentosunun (CEM I 42.5 R) özellikleri	8
Çizelge 3.2. Agregaların fiziksel özellikleri	9
Çizelge 3.3. Silis dumanının kimyasal ve fiziksel özellikleri	9
Çizelge 3.4. Silis dumanı elek analiz sonuçları	10
Çizelge 3.5. Çelik tellerin özellikleri	10
Çizelge 3.6. Agregalar karışım oranları (hacimce)	11
Çizelge 4.1. Beton deney sonuçları	18

SİMGELER VE KISALTMALAR DİZİNİ

a	Çentik derinliği
A	Betonun yoğunluğu
ASP	Active Server Pages
B	Kiriş genişliği
CMOD	Çatlak ağzı genişliği
COM	Commercial
D	Kiriş yüksekliği
DLL	Dynamic Link Library
E	Elastisite modülü
F_L	Eğilme dayanımı
f_{net}	Net eğilme dayanımını
f_t	Betonun çekme dayanımı
G_f	Kırılma enerjisi
L	Numune boyu (kiriş boyu)
l_{ch}	Karakteristik boy
XML	Extensible Markup Language
.NET	Network

1. GİRİŞ

Günümüzde yaşam standardımız bilgisayar teknolojisi ile paralel olarak hızla değişim göstermektedir. Bilgisayar ve bilgisayar yazılımları hemen hemen hayatımızın her alanına girdiği gözlemlenmektedir. Yazılım ve donanımı kullanılabilir kılan, çeşitli görevleri yerine getirmek ve var olan bir problemi çözmek maksadıyla tasarlanmaktadır. Programlama ise belirli oranlarda bir araya getirildiği yazılımın üretim sürecine verilen addır.

Ülkemizde inşaat sektörü kalkınmadaki can damarlarından biridir. Dolayısıyla son yıllarda inşaat sektöründeki yazılım teknolojilerinin kullanımı artmıştır. Bugün inşaat alanındaki birçok disiplini destekleyecek çok değişik teknolojiler mevcuttur. Bu teknolojiler bilgisayar destekli tasarımdan proje yönetimine, görselleştirmeden maliyet tahminine kadar çok değişik alanlarda kullanılmaktadır (Tanyer ve Pekeriçli, 2008).

İnşaat biliminde, yapı bilimi her geçen gün gelişmektedir. Yapı biliminde laboratuvar alanında en çok çalışılan dallardan biri betondur. Yapılan binaların sağlamlığı için beton üzerinde çeşitli çalışmalar yapılarak sağlamlığı üzerinde testler yapılmaktadır. Betonun kırılma enerjisi saptanarak dayanıklılığı ölçülür.

Yapılan çalışmada betonun kırılma enerjisinin dataları alınarak bunun üzerinde gerekli çalışmaları yapıp grafiklerle araştırmacıları bilgilendiren bir yazılım tasarlanması amaçlanmıştır.

Bu çalışma beton yapısı üzerinde çalışan inşaat mühendisi ve araştırmacıların uzun sürede tamamladıkları işlemleri kolaylaştırması amacı ile tasarlanmıştır.

Yazılım Visual Studio platformunda C# dili kullanılarak tasarlanmıştır. Microsoft Office Excel programından beton kırılma enerjisi dataları alınarak bu datalar üzerinde çalışma yapılmıştır.

2. KAYNAK ÖZETLERİ

Çomak (2012) farklı beton numunelerinin kesit görüntülerinden, görüntü işleme tekniği kullanılarak beton basınç dayanımları tahmin edilmeye çalışmıştır. Çalışmasında, 7 farklı cu/çimento oranına sahip hazırlanan beton numuneler üzerinde sertleşmiş beton deneyleri yapmış ve benzer karışıma sahip beton numuneler üzerinde görüntü işleme yöntemi kullanmıştır. Görüntü işleme yöntemi kullanarak elde edilen sonuçlar ile beton basınç dayanımları arasındaki ilişkileri irdeleyebilmek için C# programlama dilinde bir yazılım geliştirmiştir. Böylece, tahribatlı ve tahribatsız deney yöntemlerine yardımcı bir yöntem olarak görüntü işlemenin kullanılıp kullanılmayacağı tespit edilmeye çalışılmıştır.

Adekunle vd. (1983) çelik şaftların kırılma mekaniğinin tam olarak anlaşılamadığını ve kırılma esnasında yaşam kayıplarının bile yaşandığını vurgulamış. Çelik şaft tasarımının mazleme, üretim maliyeti ve güvenlik açılarından önemli olduğunu belirtmişlerdir. Bu nedenle yaptıkları çalışmada çelik şaft üretimin tasarımında kullanılmak üzere geliştirdikleri program C# tabanlı programda statik yükleme, eğilme yükleri gibi değişen çoklu yükleme koşulları ve değişik çapları da dikkate alarak tasarımlar yapabilmıştır.

Zheng ve ark. (2012) karakter tanımlama amaçlı C# tabanlı bir program geliştirmişlerdir. Çalışmalarında resim üzerindeki yazı karakterlerini yazdıkları program tanımlayarak metinlere başarılı bir şekilde dönüştürmüştür.

Tanyer ve Pekeriçli (2008) inşaat sektörü için bilgi teknolojilerindeki son gelişmeleri incelemişler ve ülkemizde bu teknolojilerden çok küçük bir bölümünün günlük inşaat süreçlerine entegre olduğunu belirtmişlerdir.

Biczo ve Pocza (2009) iş-akış tabanlı programlama günceçtikçe pratik problemlerin çözümünde popülerite kazanmakta olduğunu belirtmişlerdir. Fakat iş-akış tabanlı programlamanın ek komponentlere ve bilgiye ihtiyaç

duyduklarını belirtmişlerdir. Ve otomatlar için fonksiyonel bir uygulama üretmek için C# tabanlı basit bir yapı geliştirmişlerdir.

Kozick ve Donze (2008) C++ kullanarak geliştirdikleri program ile malzemelerin mekanik davranışını incelediler.

Mosa vd. (2013) inşaat mühendisliğinde karayolu tasarımında sahada pek çok problemle karşılaştıklarını vurgulamışlardır. Özellikle yeni mezun mühendislerin bilgi ve deneyim yetersizliğinden dolayı sıkıntı yaşamaktadır. Bu nedenle bu problemlere karşı tanımlamaları, neden ve çözüm önerilerini içeren eğitim sisteminin geliştirilmesi inşaat mühendisliği mezunları için etkili bir yol olacaktır. Mosa ve arkadaşları literatürden bilgileri ve verileri elde ettikten sonra bu verileri sorgulayıp işleyen, analiz eden ve öneriler getiren bir programı Visual Basic program dilinde geliştirdiler.

Demis vd. (2014) betonarme yapıların durabilite tasarımında tahmin tabanlı bir yazılım geliştirilmesi için önemli bir çalışma yaptılar. Bu çalışmalarında betonarme yapıların beton dayanımı ve servis ömrü tahmin etmek için çimento ve beton için Avrupa standartları ile uyumlu çalışan bir model olmasının esas olduğunu vurgulamışlardır. Çalışmalarında hazırladıkları yazılım, beton karışım dizaynını tanımlamak için, geliştirdikleri yazılım betonun basınç dayanımı ve betonun ana kimyasal ve hacimsel karakteristikleri hesaplar. Yazılım hesaplamayı yaparken betonarme yapının maruz kalacağı çevresel etki sınıflarını, betonarme betonun içine gömülü bulunan donatıların korozyon mekanizmalarını ve betonun karbonatlaşmasını dikkate alan matematiksel modelleri göz önünde bulundurmaktadır.

Ca vd. (2004) hazır beton tesislerindeki beton üretimi opere eden yeni bir yaklaşım geliştirdiler. Çalışmalarında genetik algoritmaya dayalı bir simülasyon geliştirdiler. Geliştirdikleri algoritma, talep edilen betonun üretiminin planlanması, ne kadar transmiklere ihtiyaç olduğunu ve daha az maliyetli bir üretim gerçekleştirilmesini başarılı bir şekilde optimize etmektedir.

Zhang (2009) inşaat mühendisliği öğrencilerinin bina ve köprü gibi yapıları modelleyip simule edebileceği bir yazılım uygulaması üzerine çalıştı. İnşaat mühendisliği öğrencilerinin eğitiminde etkin bir öğrenme gerçekleştirmek için öğrencilerin tasarımlarını kolaylıkla modelleyebilmesine imkan sağlayabilmektedir.

Yudong ve Xiaochun (2009) bir sonlu eleman yazılımı olan abaques kullanarak üç noktadan yüklemeli kirişin yük ve deformasyon davranışını sayısal yöntemle simule etti.

Gencel vd. (2013) betonun aşınma dayanımı minimize etmek için istatistiksel bir yöntem olan tepki yüzeyi metodu (respond surface method) kullanarak etkin parametreleri tespit etti. Yaklaşım zaman ve maliyet açısından önemli katkı sağlamaktadır.

Sütçü vd. (2014) Ansys programını kullanarak yapılarda kullanılan ısı yalıtım briketlerinin geometrilerinin ısı performans etkilerini inceledi. Ansys programına uygun veriler doğru bir şekilde tanımlandığında ısı davranış başarı ile simule edilebilmektedir. Ayrıca Tuğlaların ısı davranışına etki eden parametreler yine Ansys ortamında deneysel tasarım parametreleri ile belirlenmişlerdir.

Yukarıdaki literatür özetlerinden de anlaşılacağı üzere C#'ın beton teknolojisinde kullanımı oldukça kısıtlı olmakla beraber, betonun kırılma enerjisini hesaplayan bir çalışma bulunmamaktadır. Bu nedenle yapılan bu tez çalışması bu boşluğu dolduracaktır.

3. MATERYAL ve YÖNTEM

3.1. Visual Studio

Yazılım kirilis.exe adı ile Visual Studio yazılım geliştirme programında C# dili kullanılarak geliştirilmiştir.

Bilindiği gibi bilgisayarları kullanabilmek için önce onları programlamak gerekir. Bilgisayarlar ilk çıktıkları günden bu yana değişik düzeyde birçok programlama dili kullanılarak programlanıp insanlara faydalı araçlar haline getirilmiştir. Bilgisayar dilleri düzeylerine göre şu şekilde sınıflandırılabilir:

Script dilleri: Javascript, Perl Script, VBScript...

Yüksek düzeyli diller: VBasic ve Delphi

Orta düzeyli diller: C# ve Java

Düşük düzeyli diller: C/C++

Assembly: Assembly

C# orta düzeyli diller grubuna girer. Özellikle hem alt düzey hem de üst düzey programlar yazılabilir. C# ile programlar geliştirmek için güvensiz kod bloklarında işaretçiler kullanılır. Diğer taraftan büyük çaplı kurumsal projeler geliştirirken kullandığımız dilin çok zor olmaması ve verimli olması her zaman tercih sebebidir. Bu durumda C++ kullanmak gerçekten etkili olmayabilir.

C#'ın söz dizimi C/C++ dillerinden gelmektedir. Uzun seneler yazılım geliştirme alanında en popüler dil olarak C kullanılmıştır. Sonra bu dilin Smalltalk gibi nesne yönelimli diller karşısında zayıfladığı fark edildiğinde C++ geliştirilmiştir. C++; güçlü C diline nesne yönelimli özelliklerin katıldığı 80'li yıllardan bu yana profesyonel programcıların tercihi olan programlama aracı olagelmıştır.

Geliştirilecek olan yazılımımızda C# dilinin kullanılmasının birden çok nedeni vardır. Bunlar;

- C# dili öğrenilmesi kolay bir dildir: C# dili diğer dillere nazaran daha kolay öğrenilebilecek bir dildir. Az sayıda anahtar kelime (keyword) ve basit kontrol yapıları ile çalışılabilir.
- Nesne Yönelimli Programlamaya tam destek verir: C# gerçek manada nesne yönelimli (object oriented) bir dildir. Bu sayede büyük ölçekli projeleri hızlı bir şekilde geliştirilebilir. Ayrıca, program hatalarının giderilmesi veya yeni modüller ekleme safhalarında tam donanımlı nesne yönelimli bir dille çalışmak büyük faydalar sağlar.
- Yüksek verim verir: C/C++ ile yazılan programlar diğer dillere göre daha uzun zamanda ortaya çıkarlar. Sebebi ise bu dillerin zor olması ve hata yapmaya çok açık olmasıdır. Fakat Microsoft, C#'ı tasarlarken çok sık yapılan hataları önlemek için önlemler almıştır. Bu önlemlerin en basiti if() deyiminde parantezin içindeki değerin bool türünden olması gerekliliğidir, yani C#'taki if() içindeki ifadenin değeri int veya türevi bir değer olursa program derlenmez. Bu sayede bir çok hatanın önüne geçilmiş olur.
- Güç ve Kolaylık arasındaki denge: VB kolay kullanılır bir dil olmasına rağmen C++'a göre hem yavaştır hem de yetenekleri azdır. C# dili VB kadar hızlı, C++'taki kadar güçlü programlar geliştirmenin en uygun yoludur.
- XML desteği: Günümüzde yazılan programların sadece bir makinede çalışması yetersiz hale gelmiştir. Artık yazılımların intranet ve internete açılan yönlerinin olması ve başka sistemler ile kolayca etkileşime geçebilme yeteneği olmalıdır. Bu durumda farklı platformların birbiri arasında veri alışverişlerinde ortak bir dil ile konuşmaları gerekir. Microsoft, .NET ve C#'ın içine XML desteğini eklemiştir. Bu sayede XML verileriyle çalışmada C# dili bize büyük kolaylıklar sağlar.
- Yönetilmeyen Kod (Unmanaged Code): C# ile yazılım geliştirirken pek ihtiyacımız olmasa da yine de işaretçileri kullanırız. Bu tür program kodlarına unmanaged code (yönetilmeyen kod) denir. Bu şekilde gerçekleştirilen kodları genelde C# projemizin eski COM ve DLL ile ilişkiye geçtikleri yerlerde kullanırız. Bu sayede .NET ve C#'ta geriye dönük uyumluluk da vardır.

- VB'deki Olay (event) ve Form mantığını destekler: C# ile Windows veya ASP.NET programı geliştirmek Form mantığı sayesinde hem daha hızlıdır hem de geliştirilen uygulamalar daha güçlüdür.
- C#: Modern bir dildir: Aslında bu sonuca yukarıdaki maddeleri birleştirerek ulaşılabilir. İnternet çağının gerektirdiği tüm özellikleri desteklemesinin yanında hızlı yazılım geliştirilmesine olanak tanıyarak verimliliğimizi ve rekabet gücümüzü artırır. C/C++ ve Java'nın güzel yönleri alınıp, haliyle nesne yönelimli, mükemmel bir dil olarak karşımıza çıkar C#. Dahası eskiden yazdığımız bileşenler (COM ve DLL) ile anlaşır, gerekirse onlarla aynı dili konuşmak için işaretçileri kullanır (Microsoft, 2015).

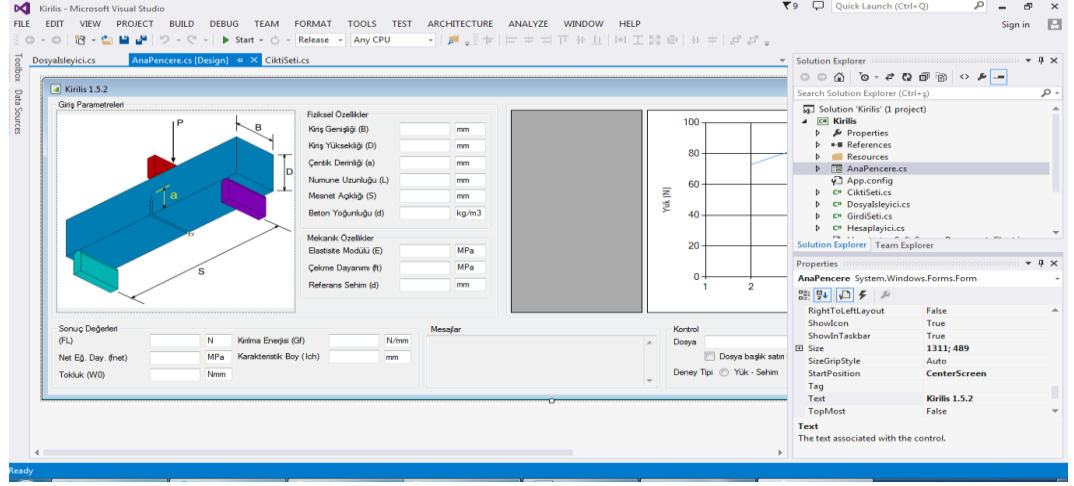
C#'ın Kullanım Alanları;

- Konsol uygulaması geliştirme aracı
- Windows için program yazma
- ASP.NET için en uygun dil
- Web servisleri:
- Mobil uygulama geliştirme
- C# ile DLL yazabiliriz.

Yazılımı geliştirirken Visual Studio programını kullanmamızın nedeni;

- İhtiyacımız olan bilgiye bağlam içinde hızla erişilebilir,
- Üretim ortamında bile, geçmişe dönük hata ayıklama için IntelliTrace'i kullanılabilir,
- Üretim ortamında ölçeklenebilirlik ve performanstan emin olmak için testler uygulanabilir,
- Bir uygulamanın yapısını görselleştirilebilir,
- Mevcut koddaki ilişkileri anlamak için araçlar kullanılabilir, olması olarak gösterilebilir (Microsoft, 2015).

Şekil 3.1'de görüldüğü gibi C# çalışma ortamı kullanıcıya rahat çalışma ortamı ve gerekli kolaylığı sağlamaktadır.



Şekil 3.1. Visual studio çalışma alanı

Bu tez çalışmasının amacını deformasyon kontrollü test cihazından gerekli dataları aldıktan sonra bu dataları istenilen parametrelere göre işleyen ve sonuçları grafik ve metin olarak veren bir yazılım hazırlamaktır. Bu yazılım sayesinde zaman tasarruf, daha sağlıklı sonuçlar sağlanacaktır.

3.2. Kullanılan Malzemeler

3.2.1. Çimento

Çalışmada Yozgat Yibitaş-Lafarge Çimento Fabrikasından temin edilen CEM I 42.5 R (PÇ 42.5) tipi Portland Çimentosu kullanıldı. Çimentonun fiziksel ve kimyasal özellikleri Çizelge 3.1’ de verilmiştir.

Çizelge 3.1 Portland çimentosunun (CEM I 42.5 R) özellikleri

Kimyasal Bileşim (%)							
MgO	SiO ₂	Al ₂ O ₃	Fe ₂ O ₃	Na ₂ O	CaO	SO ₃	K ₂ O
2.75	19.12	5.63	2.39	-	63.17	2.74	1.00
Kızdırma Kaybı				Çözünmeyen Kalıntı			
2.33				0.49			
Fiziksel Özellikler							
Özgül Ağırlık g/cm ³				Özgül Yüzey cm ² /g			
3.09				3114			

3.2.2. Agregalar

İnce agregalar olarak Kırma Kum (0-4 mm), kaba agregalar olarak Kırmataş I (4-12 mm) ve Kırmataş II (12-19 mm) kullanılmıştır. Çalışmada en büyük agregalar boyutu 19 mm olarak seçilmiştir. Agregalara ait fiziksel özellikler Çizelge 3.2’de verilmiştir.

Çizelge 3.2. Agregaların fiziksel özellikleri

Agrega	Su emme % (ağırlıkça)	Birim Ağırlık kg/m ³	Tane Birim Ağırlığı, kg/m ³
Kırma Kum	1.56	1675	2704
Kırmataş I	0.32	1635	2709
Kırmataş II	0.30	1615	2738

3.2.3. Silis Dumanı

Silis dumanı, (SD) Antalya Ferrokrom Tesislerinden temin edildi. Silis dumanının kimyasal ve fiziksel özellikleri ile tane boyutu dağılımı sırasıyla Çizelge 3.3 ve Çizelge 3.4’de verilmektedir.

Çizelge 3.3. Silis dumanının kimyasal ve fiziksel özellikleri

Kimyasal Bileşim (%)							
SiO ₂	Al ₂ O ₃	Fe ₂ O ₃	CaO	MgO	SO ₃	Na ₂ O	K ₂ O
81.35	4.48	1.42	0.80	1.47	1.34	-	-
Kızdırma Kaybı				Çözünmeyen kalıntı			
3.4				-			
Fiziksel Özellikler							
Özgül Ağırlık (g/cm ³)				Özgül Yüzey (cm ² /g)			
2.23				-			

Çizelge 3.4. Silis dumanı elek analiz sonuçları

Elek Boyutu (mm)	% Elek Üstü
>0.250	0.2-0.5
0.250-0.125	1.5
0.125-0.074	0.5-2
0.074-0.044	2.5-7.5
0.044-0.038	3-7
< 0.038	80-92

3.2.4. Süper Akışkanlaştırıcı

Özellikle düşük su/çimento oranlarında beton karışımlarının işlenebilirliğinin sağlanması ve karışım sırasında topaklanmanın önlenmesi için Yapı Kimyasalları Sanayi (YKS) firmasından temin edilen polikarboksilik eter zincirleri esaslı Glenium 51 tipi süper akışkanlaştırıcı kimyasal katkı kullanıldı. Glenium 51, erken ve nihai yüksek mukavemete gereksinim duyulan mükemmel yüzey görünümü elde etmek için, hazır beton ve prekast endüstrisi için geliştirilmiş, yüksek oranda su ihtiyacı azaltan, yeni nesil süper akışkanlaştırıcı katkı malzemesidir.

3.2.5. Çelik Tel

Çalışmada Beksa Çelik Tel ve Kord. Sanayi ve Ticaret A.Ş. firmasından temin edilen aynı performans sınıfında, ancak farklı çekme dayanımına sahip RC 80/60 BN ve RC 80/60 BP tipi iki ucu kancalı çelik teller kullanıldı. Kullanılan çelik tellerin özellikleri Çizelge 3.5'te verilmektedir.

Çizelge 3.5. Çelik tellerin özellikleri

Çelik Tel Tipi	Boy, l (mm)	Çap, d (mm)	Narinlik (l/d) (uzunluk/çap)	Yoğunluk (g/cm^3)	Çekme Dayanım, f_{su} (N/mm^2)
Dramix RC 80/60 BN	60	0.75	80	7.85	1050
Dramix RC 80/60 BP	60	0.71	85	7.85	2000

3.3. Beton Karışımları

Üretilen betonlarda kullanılan agrega oranları hacimce Çizelge 3.6'da ve karışım agregasına ait tane dağılımı eğrisi Şekil 3.1.'de, taze beton özellikleri ve gerçek karışım miktarları ise Çizelge 3.7'de verilmektedir.

Çizelge 3.6. Agrega karışım oranları (hacimce)

Agrega	%
Kırmakum	48
Kırmataş I	27
Kırmataş II	25

Şekil 3.2'de çatlak genişliğini ölçmek için kullanılan düzenek görülmektedir. Bu düzenek sayesinde çatlak genişliği ölçülmektedir.



Şekil 3.2. Çatlak Genişliğini ölçmek için kullanılan düzenek

Şekil 3.3'de kiriş numunelerinin kırıldığı eğilme çerçevesi görülmektedir. Bu çerçeve düzeneği ile kiriş numuneleri kırılmaktadır



Şekil 3.3. Kiriş numunelerin kırıldığı eğilme çerçevesi

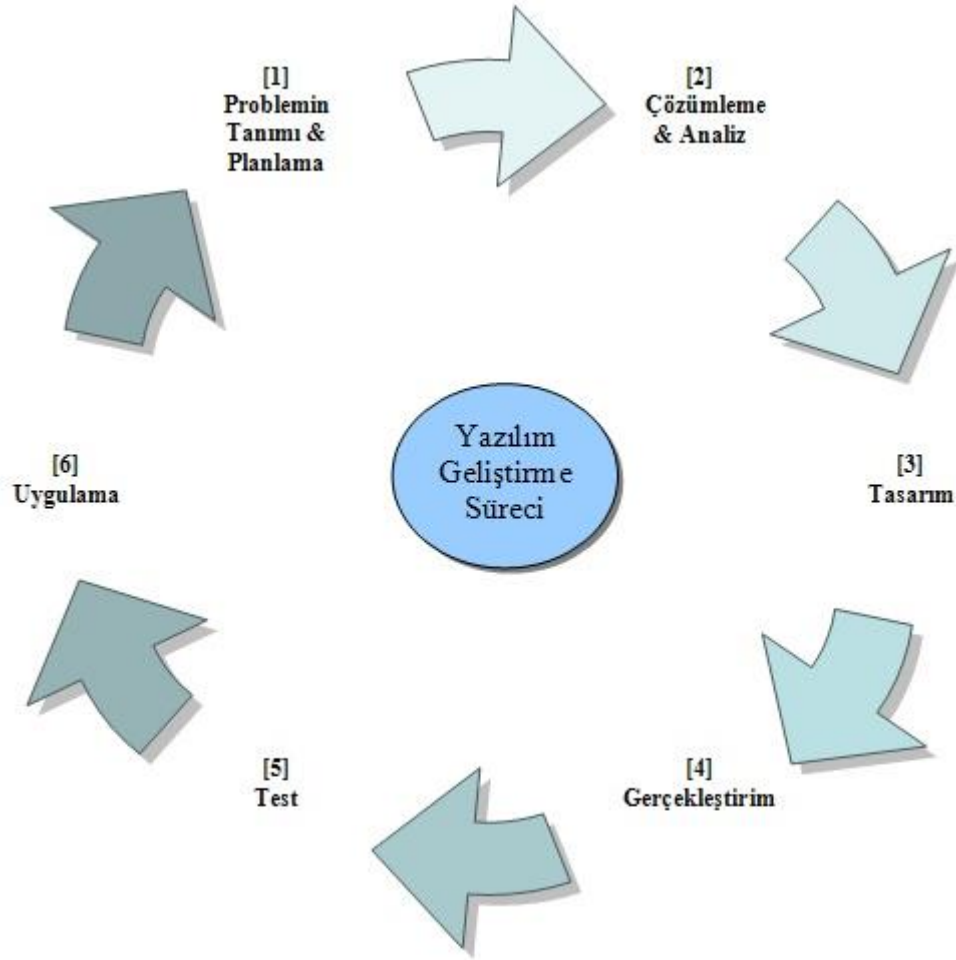
3.4. Yazılımın Tasarımı

Yazılım hazırlanırken Şekil 3.4. deki görülen yazılım tasarım aşamalarından geçilerek yazılımımız tasarlanmıştır.

- **Problemin Tanımı & Planlama:** Son zamanlarda tanıklık ettiğimiz önemli programlardan biri de kentsel dönüşüm programıdır. Bu program çerçevesinde depreme dayanıksız yapıların yıkılıp yeniden inşa edilmektedir. Kentsel dönüşüm programına ek olarak insanların değişen yaşam tarzı ve bu dorultudaki talepleri ile insanlar daha farklı ve modern konutlarda yaşamak arzu etmektedir.

Ayrıca son zamamlarda ülkemizde ciddi can kayıplarının yaşandığı depremler meydana gelmiştir. Yaşanan can kayıpları depremde dolayı değil depremde yıkılan yapılardan kaynaklanmıştır. Aynı depreme maruz kalıp yıkılmayan yapılarda can kayıpları yaşanmamıştır. Buradan da anlaşılacağı üzere konutların depreme dayanıklı inşa edilmesi esastır. Yapıları tasarlarken ticari olarak sektöre sunulan SAP2000, STA4CAD gibi bazı paket programlar

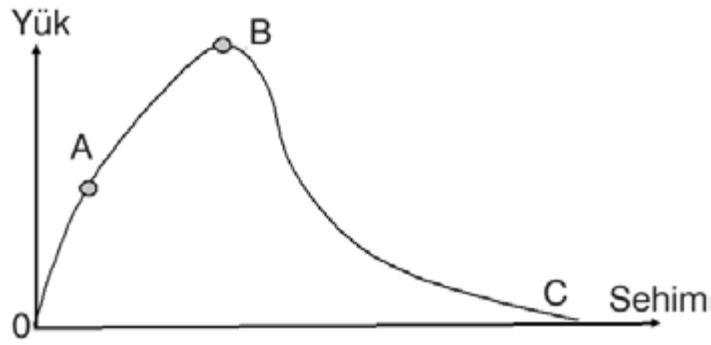
kullanılmaktadır. Bu programlar yapısal analiz yapmaktadır. Betonarmeyi oluşturan kısımlardan olan betonun özelliklerini tespit etmek için ise bir program mevcut değildir. Özellikle betonun kırılma mekaniğini anlamak ve kırılma enerjilerini hesaplamak önemlidir.



Şekil 3.4. Yazılım geliştirme süreci

Betonun kırılma mekaniği depreme dayanıklı binalar, savunma amaçlı yapılar, nükleer santraller, iklim ve çevre koşullarının etkisinde yıpranmaya maruz binalar gibi ekonomik açıdan maliyeti yüksek yapılar için oldukça büyük bir öneme sahiptir. Beton, yük altında olmasa bile içinde boşluklar ve mikroçatlaklar bulunan heterojen ve yarı gevrek bir malzemedir ve düşük yükler altında lineer elastik bir davranış gösterir (O-A arası, Şekil 1.1). Bu bölgede gerilmeler ve şekil değiştirmeler orantılı olarak artar ve yük kalktığında şekil değiştirmeler de kalkar. Yük arttırıldığında, elastik sınır geçilir (A noktası),

betonun içindeki mikroçatlaklar ve boşluklar aktif hale gelir ve yük-sehim eğrisi lineerliğini kaybeder. Özellikle agrega-çimento hamuru arayüzeyindeki boşluk ve mikroçatlaklar büyür ve betonda kalıcı şekil değiştirmeler meydana gelir. Oluşan bu çatlaklar enerji harcadığı için yük - sehim eğrisi nonlineer bir şekilde artış gösterir (A-B arası). Bu bölgede mevcut çatlakların büyümesinin yanı sıra yeni çatlaklar da oluşmaya başlar. Meydana gelen şekil değiştirmeler, tepe yükü civarında (B noktası) kırılmanın gerçekleşeceği düzlemde birikmeye başlar (şekil değiştirme yerelleşmesi). Çatlaklar ilerlemeye devam ettikçe şekil değiştirmeler artar ve yük taşıma kapasitesi azalır. Böylece çatlaklar adım adım ilerleyerek betonun yük taşıma kapasitesinin aniden sıfıra düşmesini engeller, ve tepe yükü sonrasında betona tokluk kazandırır. B-C noktaları arasında kalan bu bölgede şekil değiştirme yumuşaması görülür.



Şekil 3.5. Betonun eğilme yüklemesi altındaki davranışı (0 -A noktaları arasında lineer davranış, A-B noktaları arasında nonlineer bölge, B-C noktaları arasında şekil değiştirme yumuşaması)

Çatlak ucundaki kırılma süreci bölgesi, bir tarafından diğer tarafına yük aktarabilen bir çatlak uzantısı olarak düşünülür. Çentikli kiriş numuneler üzerinde üç noktalı eğilme deneyi yapılarak yük – toplam şekil değiştirme eğrisi elde edilir. Bu eğriden elastik şekil değiştirmeler çıkarıldığında, yük - çatlak genişliği eğrisi bulunur. Bu eğri altında kalan alandan da kırılma enerjisi hesaplanır. Bulunan parametre sayısal analizde kullanılabilen, uygulaması kolay bir yöntemdir. Ancak kiriş ortasındaki sehimin ölçülmesi gerektiğinden ve yük sıfır olana kadar hassas biçimde ölçmek güç olduğundan deneysel zorluklar içerir. Kırılma enerjisi gibi bir tek parametre, hesaplanması da bir sakıncadır.

Bulunan kırılma enerjisi yardımıyla hesaplanan karakteristik boy (l_{ch}) yeni çimento esaslı kompozitlerinin mikro çatlaklara karşı tasarımında başarılı biçimde kullanılmaktadır. Karakteristik boy nominal dayanımı, kırılma modunu ve çatlak büyümesini (çatlak modelini) kontrol ettiğinden önemlidir.

Betonun kırılma enerjisi, beton yapıların göçmesinin anlaşılması için gerekli önemli bir kırılma karakteristiğidir. Ancak betonun kırılma enerjisini hesaplamak kolay değildir. Bunun için deformasyon kontrollü kapalı döngü test cihazı ve donatılarına gerek vardır. Bu cihazla elde edilen değerlerin hesaplanarak kırılma enerjisinin bulunması ise ayrıca zordur bir iştir. Ayrıca bu cihazları her yerde bulmak mümkün değildir. Bir deneysel çalışmada kullanılan parametreler ve numune sayıları gözönünde tutulduğunda betonun kırılma enerjisinin hesabı uzun süre alabilmektedir. Ayrıca çok tekrar nedeni ile sık sık gözden kaçan hatalar yapılmaktadır. Bu nedenle tüm bu işleri kısaltmak ve bu işleri yaparak hata yapma ihtimalini ortadan kaldırmak için bir yazılıma ihtiyaç vardır.

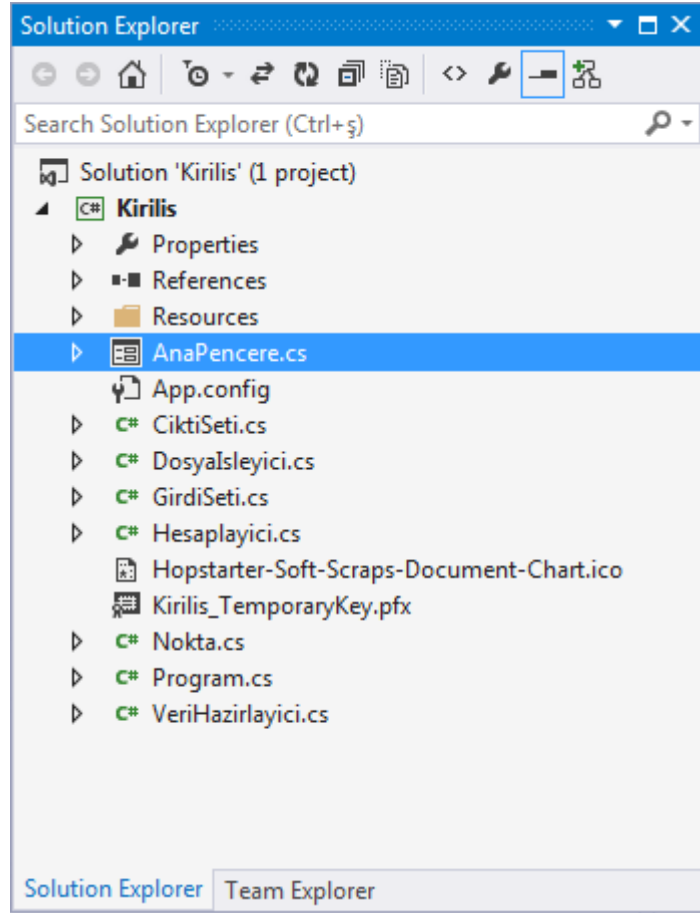
- **Çözümleme & Analiz:** Bu tez çalışmasının amacını deformasyon kontrollü test cihazından gerekli dataları aldıktan sonra bu dataları istenilen parametrelere göre işleyen ve sonuçları grafik ve metin olarak veren bir yazılım geliştirmektir. Bu yazılım sayesinde zaman ve işçilikten tasarruf, daha sağlıklı sonuçlar (hatasız) sağlanacaktır.

Bu çalışmada betonun kırılma enerjisini hesaplamak için dataları istenilen parametrelere göre işleyen ve sonuçları grafik ve metin olarak veren Visual C# programlama dili kullanılarak bir yazılım geliştirilmesi amaçlanmıştır.

- **Tasarım:** Yazılımımız tasarlanırken Visual Studio programından faydalanılmıştır. Program görsel programlamanın en rahat yapılabileceği görsel yazılım programlarından biridir.

İlk etapta bir form üzerine kullanıcının yazılımı kullanırken daha rahat kullanmasını sağlayacak bileşenler (Buton, Label, PictureBox, TextBox vb.) eklenmiştir. Daha sonra bunlar üzerinden kod sayfaları oluşturulmuştur.

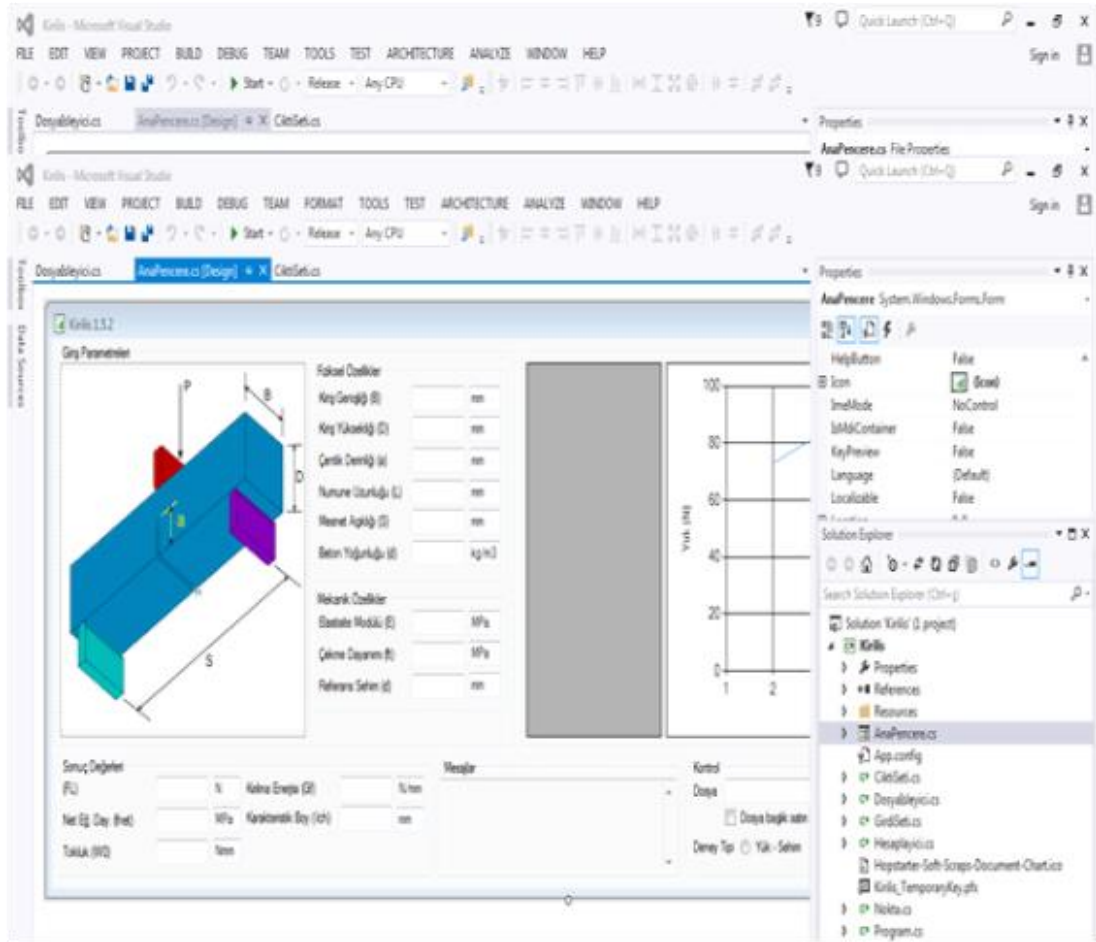
Şekil 3.6'da Kirilis yazılımında kullanılan anasayfa ve kod sayfaları görülmektedir.



Şekil 3.6. Kirilis yazılımında kullanılan anasayfa ve kod sayfaları

- Gerçekleştirim: Yazılım tasarımı gerçekleştirilirken beton kırılma enerjisi ve bu alandaki gereksinimler gözönünde bulundurularak rahat ve hızlı bir şekilde istenileni verebilecek bir yazılım olması sağlanmaya çalışılmıştır. Kod yazımı aşamasında bu konuda inşaat mühendisleri ile görüşülmüştür.

Şekil 3.7'de kirilis yazılımı geliştirme ortamı görülmektedir.



Şekil 3.7. Kirilis yazılımı geliştirme ortamı

- Test: Programın yazılımı tamamlandıktan sonra bir çok kez test edilmiş ve bulunan hatalar düzeltilmiştir. Yazılımımız Microsoft Office Excel programından verileri almaktadır ve bu veriler üzerinde çalışmaktadır. Ayrıca form üzerindeki grafik kutusuna veri değerlerine göre grafikler çizmektedir. Kullanıcıların istekleri üzerine yazılımımızın çeşitli noktaları test aşamasında geliştirilmiştir.
- Uygulama: Kirilis yazılımı test aşaması tamamlandıktan sonra inşaat laboratuvarlarında kullanılmaya başlanmış ve istenilen sonuçlar alınmıştır. Elle yapılarak uzun süren işlemler Kirilis yazılımı sayesinde dakikalar içinde gerçekleştirilmiştir.

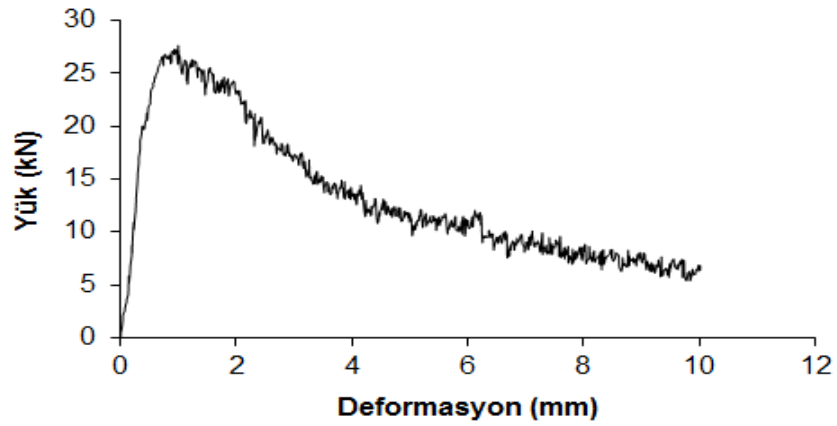
4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yazılan yazılımda girdi olarak kullanılacak beton parametreleri deneyler yapılarak elde edilmiş olup Çizelge 4.1’de belirtilmiştir. Ayrıca üretilen kiriş numune eğilme deneyine tabi tutularak elde edilen gerilme-deformasyon değerleri kullanılarak Şekil 4.1’de gösterilen yük-deformasyon grafiği çizilmiştir.

Çizelge 4.1 Beton deney sonuçları

Özellik	Simge	Değer	Birim
Elastisite modülü	E	39500	MPa
Çekme dayanımı	f_t	8.05	MPa
Kiriş genişliği	B	150	mm
Kiriş yükeksekliği	D	150	mm
Çentik derinliği	A	60	mm
Kiriş boyu	L	750	mm
Mesnetler arası mesafe	S	500	mm
Betonun yoğunluğu	α	2412	kg/m ³

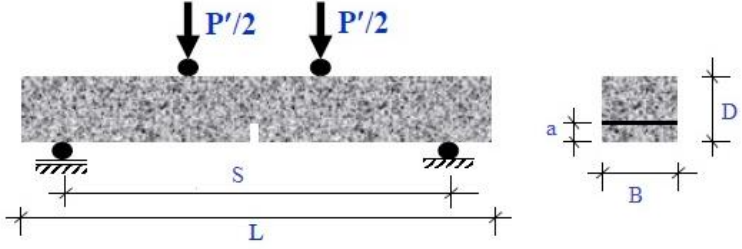
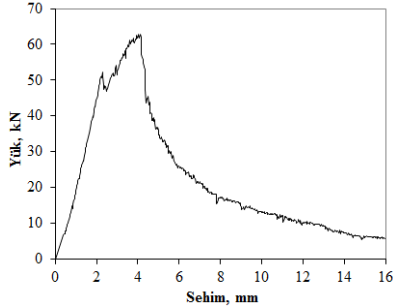
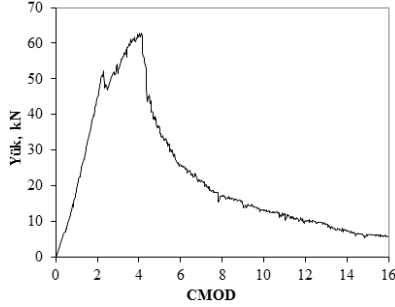
Şekil 4.1’de kırılan malzemenin yük ve deformasyon eğrisi verilmiştir.

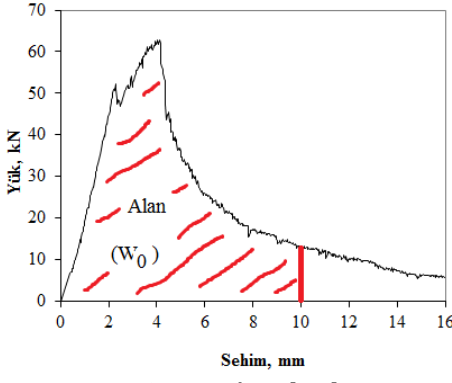
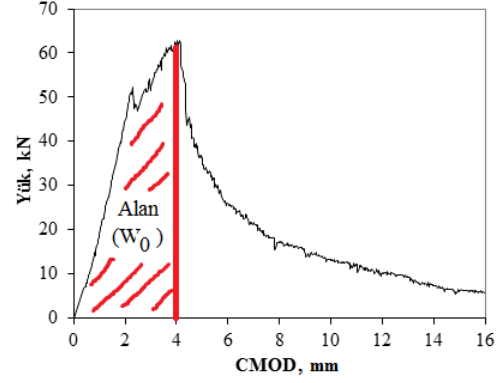


Şekil 4.1. Yük-deformasyon eğrisi

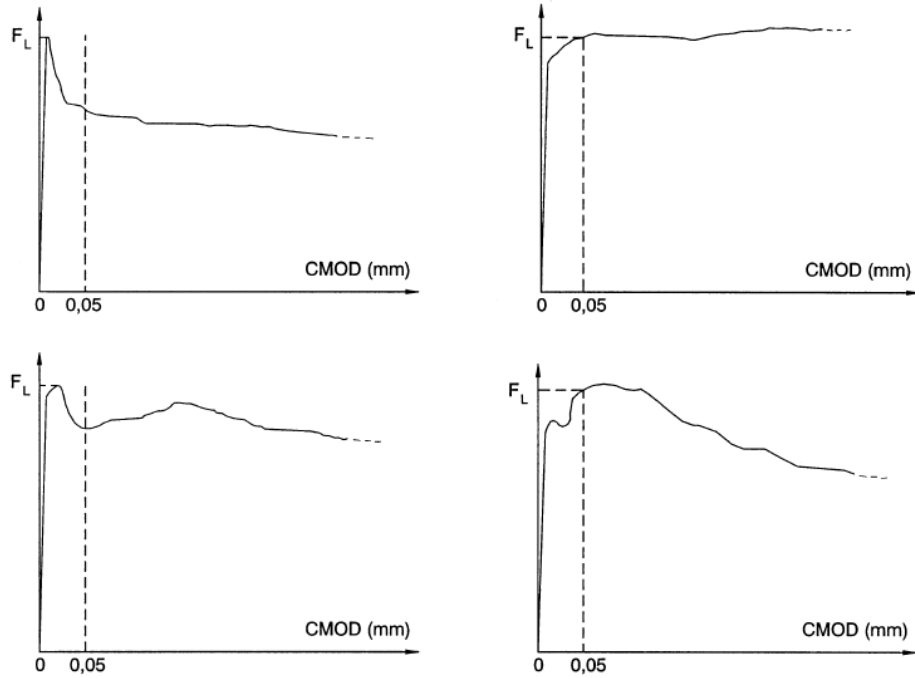
4.1. Yazılım Çalışma İşlem Sırası ve Akış Diagramı

Yazılım aşağıdaki işlem sırasına göre çalışmaktadır.

İŞLEM SIRASI		
Sıra no	İş tanımı	
1	Excel'de A ve B sütünü A = x B = y x : sehim veya CMOD (mm) y : yük (N veya kN) CMOD (Crack Mouth Opening Displacement)	
2	Kiriş numune boyutları  B : Kiriş genişliği, mm a : Çentik derinliği, mm D : Kiriş yüksekliği, mm L : Numune boyu (kiriş boyu), mm α : Betonun yoğunluğu, g/cm³	
3	Diğer veriler: E: Elastisite modülü, N/mm² f_t : Betonun çekme dayanımı, N/mm²	
4	Deney Seç	
5	Yük – Sehim	Yük – CMOD
6	x_i ve y_i oku ve hafızaya al (x_i, y_i) grafiğini çiz Örnek;  Yük-Sehim grafiği	x_i ve y_i oku ve hafızaya al (x_i, y_i) grafiğini çiz Örnek;  Yük-CMOD grafiği
7	F_L'yi bul ve hafızaya al (F_L EN 14654'te Şekil 6'ya göre bulunur. Bu grafikler aşağıda Şekil 4.2'de gösterilmiştir.)	
8	Net eğilme dayanımını hesapla; $f_{net} = \frac{3 \cdot F_L \cdot S}{2 \cdot B \cdot (D - a)^2} (N/mm^2)$	Net eğilme dayanımını hesapla; $f_{net} = \frac{3 \cdot F_L \cdot S}{2 \cdot B \cdot (D - a)^2} (N/mm^2)$

9	Tokluk (W_0) hesapla  W_0, 10 mm'ye kadar	Tokluk (W_0) hesapla  W_0, 4 mm'ye kadar
10	Kırılma enerjisi (G_f) hesapla; $G_f = \frac{W_0 + m \cdot (1 - K^2) \cdot g \cdot \delta_0}{B \cdot (D - a)}$ $g = 9.80665$ $K = (L/S) - 1$ $m = B \times D \times S \times \alpha$ $\delta_0 = 10 \text{ mm}$	Kırılma enerjisi (G_f) hesapla; $G_f = \frac{W_0 + m \cdot (1 - K^2) \cdot g \cdot \delta_0}{B \cdot (D - a)}$ $g = 9.80665$ $K = (L/S) - 1$ $m = B \times D \times S \times \alpha$ $\delta_0 = 4 \text{ mm}$
11	Karakteristik boy (l_{ch}) hesapla $l_{ch} = \frac{E \cdot G_f}{f_t^2}$	Karakteristik boy (l_{ch}) hesapla $l_{ch} = \frac{E \cdot G_f}{f_t^2}$

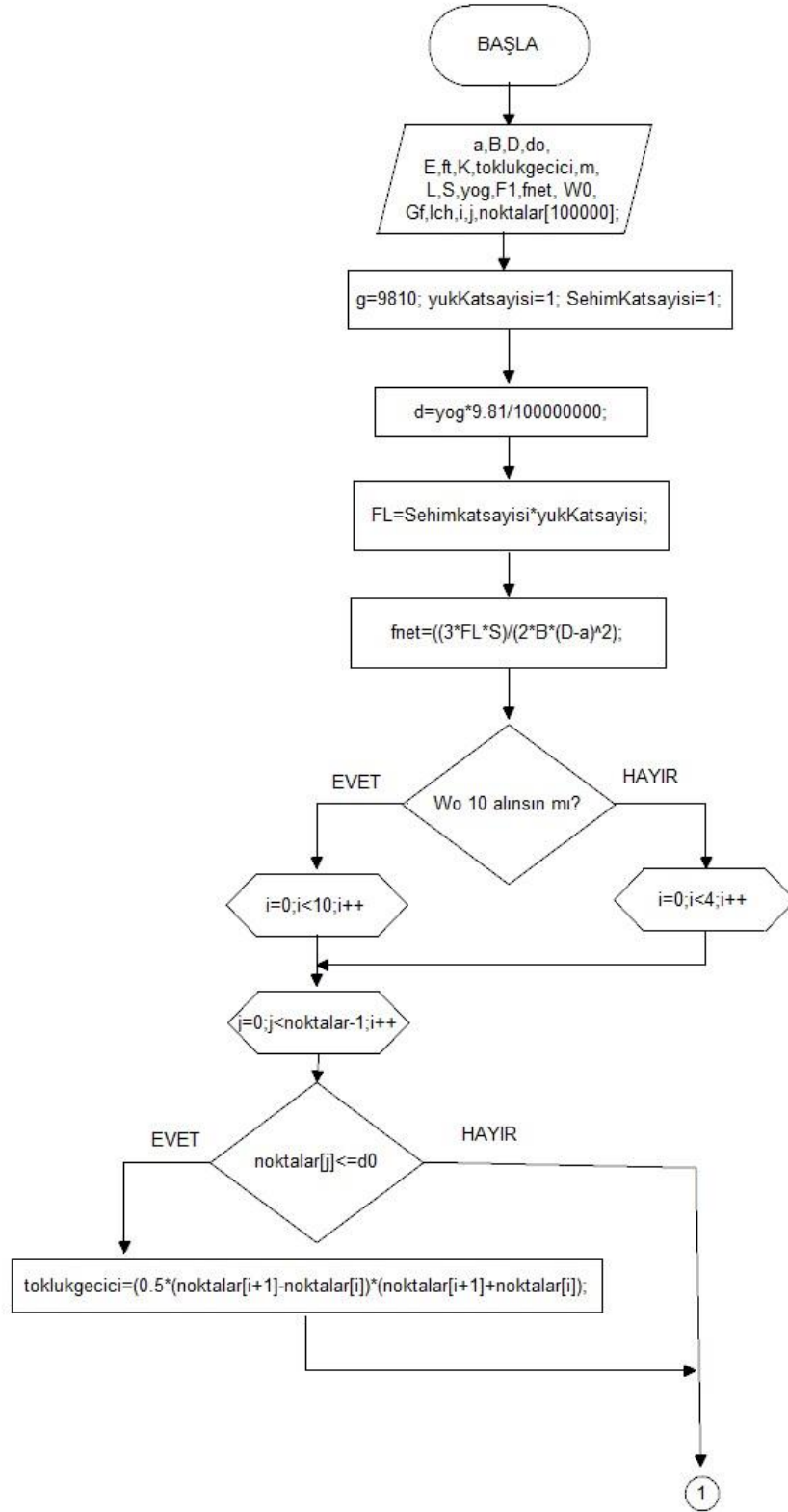
Şekil 4.2'de Yük-CMOD ve F_L diyagramları görülmektedir. Betonun F_L değerini hesaplamada bu grafikler kullanılmaktadır.



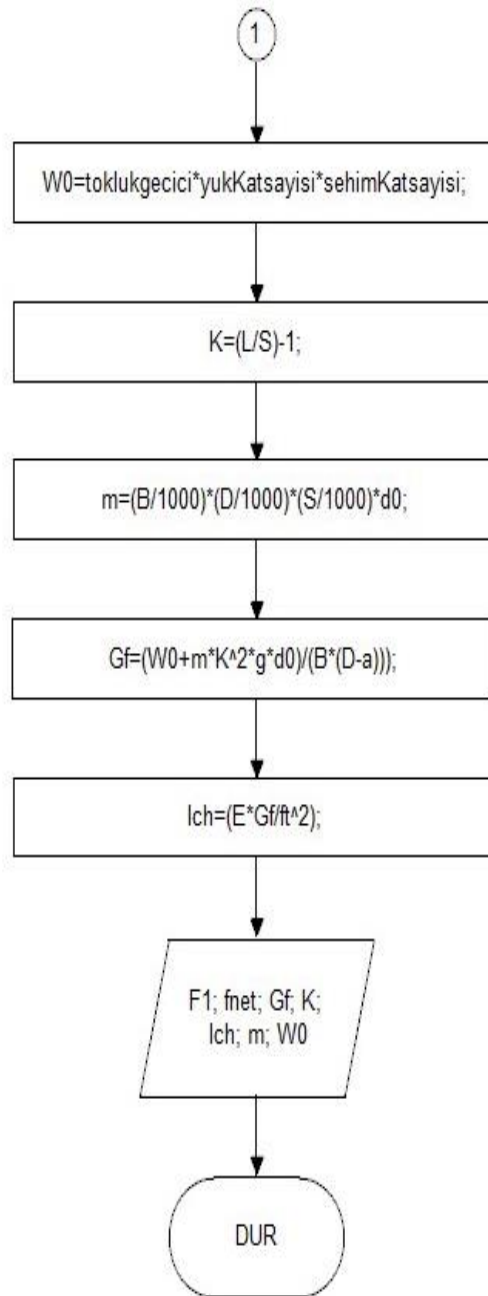
Şekil 4.2. Yük-CMOD ve F_L diyagramları

4.2. Yazılımın Akış Diagramı

Yazılım için akış şeması Şekil 4.3 ve 4.4’de verilmiştir.



Şekil 4.3. Akış diyagramı birinci bölüm



Şekil 4.4. Akış diyagramının ikinci bölüm

Akış diagramında gösterilen ;

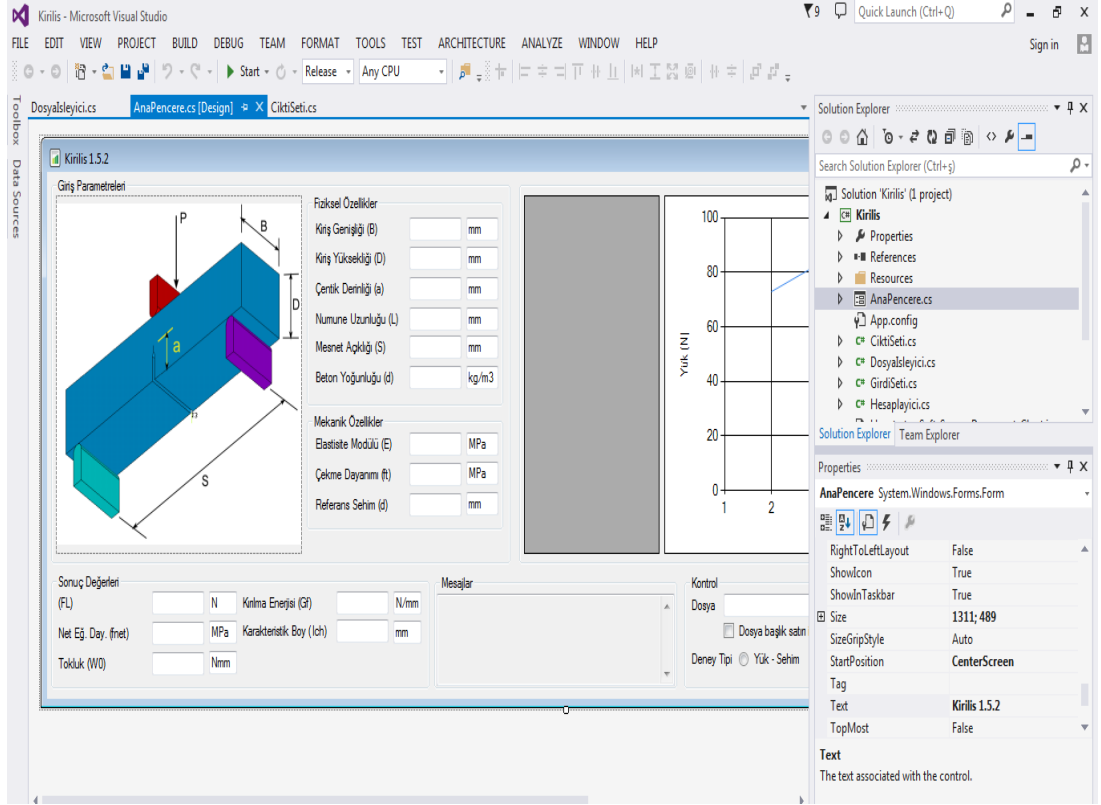
B : Kiriş genişliği, mm
 D : Kiriş yüksekliği, mm
 a : Çentik derinliği, mm
 L : Numune boyu (kiriş boyu), mm
 α : Betonun yoğunluğu, g/cm³
 E: Elastisite modülü, N/mm²

f_t : Betonun çekme dayanımı, N/mm²
 f_{net} = Net eğilme dayanımını
 G_f = Kırılma enerjisi
 l_{ch} = Karakteristik boy
 F_L : Eğilme dayanımı anlamına gelmektedir.

Yazılımın ilk kısmında veriler okutulmaktadır. Daha sonra g değerine sabit olarak belli olan 9810 değeri, yük katsayısı ve sehim katsayısı değişkenlerinede 1 değeri atanmaktadır. Devamında ilk olarak standart olarak belirlenen F_L değeri sonrasında da formüle uygun olarak f_{net} değeri bulunur.

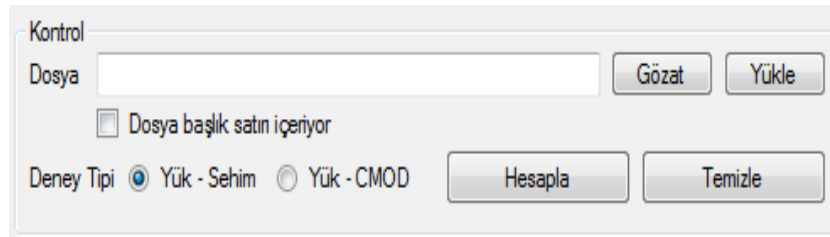
F_{net} değeri bulunduktan sonra W_o yani tokuluk değerine göre program çalışması ikiye ayrılır. Kullanıcı W_o değerini 10 alınsın derse programın değerlere göre çizdiği grafikte 10'a kadar olan değerler alınır hayır derse 4e kadar olan değerler alınır.

Yazılımın sonraki kısmında bulunan tokluk değeri de formüle yerleştirilerek G_f yani kırılma enerjisi hesaplanır. En sonda l_{ch} yani karakteristik boy hesaplanır. Kullanıcı istemesi takdirinde grafikleri ve verileri Microsoft excel programında aktarılmış şekilde görebilmektedir. Bu işlemler sonunda elde edilen kırılma enerjisinin bir göstergesi olan karakteristik boy elde edilerek betonun kırılma davranışı hakkında bilgi sahibi olunur. Yazılım olmadığı takdirde elle uzun zaman alan bu işlemler yazılım sayesinde daha hızlı elde edilmiştir. Şekil 4.5'de yazılımın ana sayfası görülmektedir. Kullanıcı programı açtığında bu sayfa ile karşılaşmaktadır.



Şekil 4.5. Kirilis yazılımı ekran görüntüsü

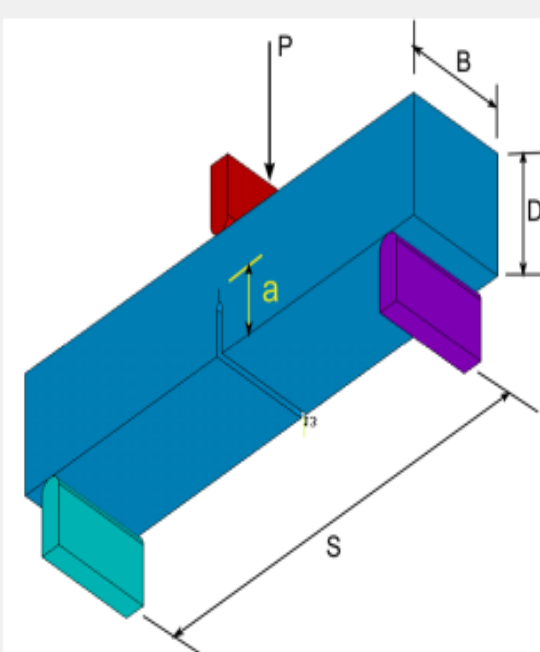
Kirilis yazılımı kontrol bölümünde Şekil 4.6'da görüldüğü gibi verileri yükleyeceğimiz dosyayı seçmek için öncelikle gözet butonuna tıklayarak excel dosyası seçiliyor daha sonrada yüklenerek verilerin yüklenmesi sağlanıyor ayrıca hesaplamayı kullanıcı isteğine göre iki deney tipinde yapmaktadır. Bunlar Yük Sehim veya Yük CMOD dur. İki seçeneğin standartları verilmiştir.



Şekil 4.6. Kirilis yazılımı kontrol bölümü ekran görüntüsü

Dosyadaki veriler yüklendikten sonra giriş parametrelerini yani Şekil 4.7'de görüldüğü gibi kiriş genişliği, kiriş yüksekliği, çentik derinliği, numune uzunluğu, mesnet açıklığı, beton yoğunluğu, elastisite modülü, çekme dayanımı ve referans sehimini kullanıcının numune bilgilerine göre elle girmesi gerekmektedir.

Giriş Parametreleri



Fiziksel Özellikler

Kiriş Geniřlięi (B)	0	mm
Kiriş Yükseklięi (D)	0	mm
Çentik Derinlięi (a)	0	mm
Numune Uzunluęu (L)	0	mm
Mesnet Aıklıęı (S)	0	mm
Beton Yoęunluęu (d)	0	kg/m ³

Mekanik Özellikler

Elastisite Modülü (E)	0	MPa
Çekme Dayanımı (f _t)	0	MPa
Referans Sehim (d)	10	mm

Şekil 4.7. Giriş parametreleri ekran görüntüsü

Eğer hiçbir veri yüklenmeden kullanıcı yanlışlıkla hesaplama bölümüne tıklarsa program mesaj bölümünde Şekil 4.8’de olduğu gibi kullanıcıya uyarı vermektedir.

Mesajlar

- > Form hazırlanıyor.
- > Form hazırlanması tamamlandı.
- > Hesaplama için giriş verisi bulunamadı. Lütfen bir dosyadan giriş değerlerini yükleyin.

Şekil 4.8. Kirilis yazılımı mesajlar bölümü ekran görüntüsü

Veriler yüklenip deney tipi seçilerek hesaplama butonuna tıklandıktan sonra sonuçlar Kirilis yazılımının Sonuç Değerleri kısmında Şekil 4.9’da da görüldüğü gibi çıkmaktadır.

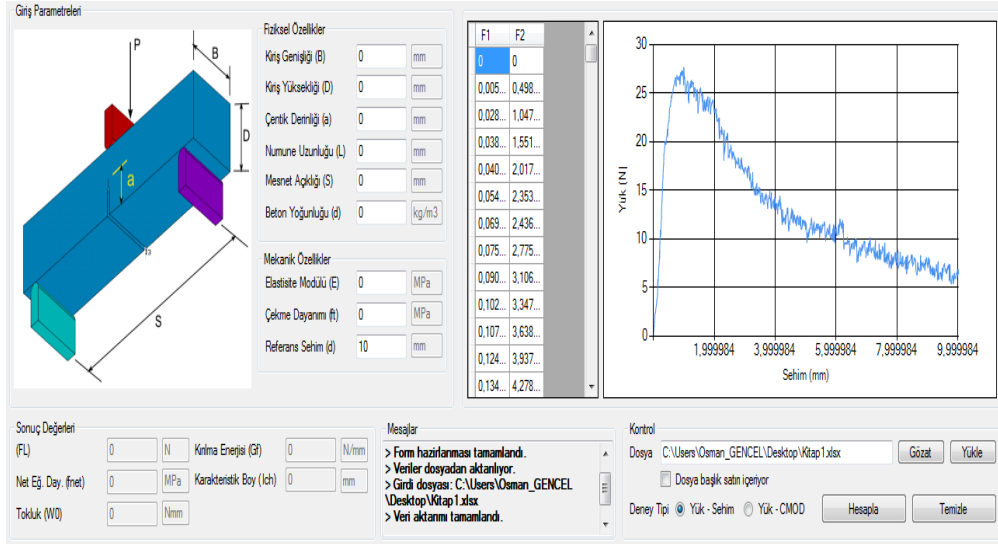
Sonuç Değerleri

(FL)	0	N	Kırılma Enerjisi (Gf)	0	N/mm
Net Eğ. Day. (f _{net})	0	MPa	Karakteristik Boy (l _{ch})	0	mm
Tokluk (W ₀)	0	Nmm			

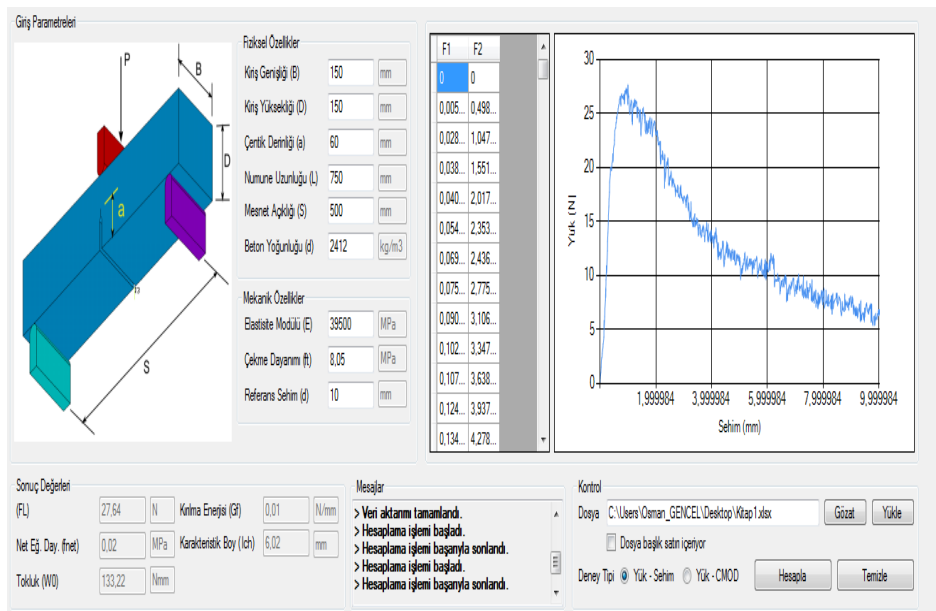
Şekil 4.9. Kirilis yazılımı sonuç değerleri bölümü ekran görüntüsü

4.3. Yazılımın Çalıştırılması

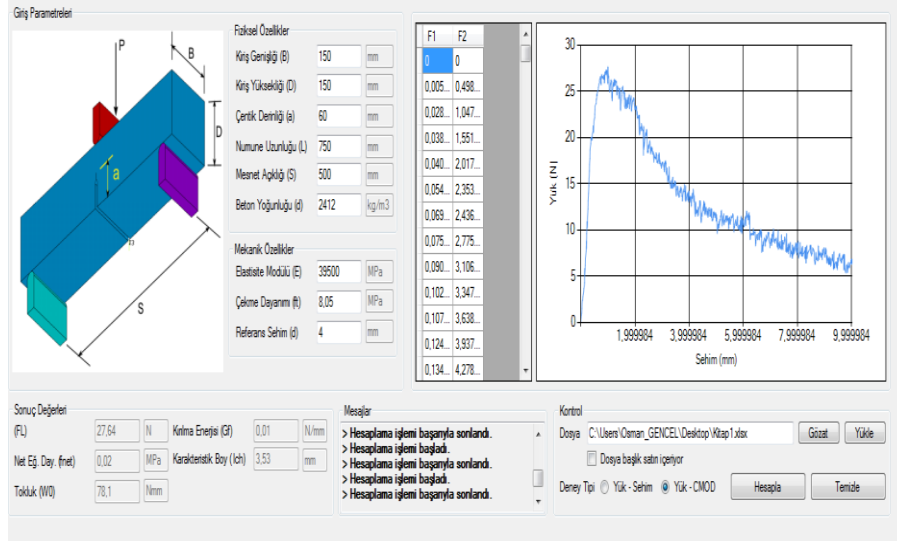
Program Kirilis.exe dosyasına tıklanarak çalışmaktadır. Karşımıza Şekil 4.7'deki Anasayfa gelmektedir. Şekil 4.10'daki Kontrol bölümündeki Gözet bölümüne tıklanır ve excel dosyamız bulunur. Daha sonra Yükle butonuna tıklanır.



Şekil 4.10'daki gibi Excel deki verilerimiz ve yan kısmında bu verilerden yararlanılarak oluşturulan grafiğimiz çizilir. Bu aşamadan sonra beton numunemizin bilgilerini giriş parametrelerine girip hesaplama butonuna tıklarız.



Şekil 4.11'deki gibi sonuç değerleri kısmında değerlerimizi yazılım vermiştir. Bu hesaplama deney tipi Yük-Sehimde olarak gerçekleşmiştir. Deney tipini Yük-CMOD seçersek Şeki 4.12'de görüldüğü gibi sonuç değerleri çıkacaktır.



Şekil 4.12. Kirilis yazılımına sonuç değerleri alma görüntüsü

Kirilis yazılımımız Bozok Üniversitesi İnşaat Mühendislięi laboratuvarında 6 ay süre ile denenmiştir. Alınan sonuçlar elle kontrol edilerek doęruluęu saptanmıştır.

5. SONUÇ VE ÖNERİLER

Bilgisayar ve bilgi teknolojileri birçok alanda hayatımızı değiştirmeye devam etmektedir. İnşaat alanında birçok disiplini destekleyecek çok değişik teknolojiler mevcuttur. Bu teknolojiler, bilgisayar destekli tasarımdan proje yönetimine, görselleştirmeden maliyet tahminine, strüktürel hesaplamalardan bina işletmesine kadar çok değişik alanlarda kullanılmaktadır.

Bilgisayar teknolojisinin olmadığı dönemlerde hesaplamalar ve kayıtların elle yapılır olması insanları zaman ve maliyet yönünden zorluyordu. Bilgisayar teknolojisinin gelişmesiyle işlemler kolaylıkla halledilir olmuştur.

Tasarladığımız yazılımında en büyük avantajı kullanıcının işlemleri daha hızlı yapabilmesini sağlamaktır. Ayrıca bu işlemleri görselleştirilmiş ve bir arayüz oluşturulmuştur.

Yazılımın Microsoft Excel programına bağlanarak oradan verileri alması ve bu verileri görselleştirmesi kullanıcının işini kolaylaştırmıştır.

Yazılımın kullanılmaması takdirinde tüm formül işlemleri ve diyagram işlemleri elle yapılması gerekiyor buda kullanıcıya zaman kaybına neden oluyordu. Bu alanda yapılan başka yazılımın olmamasından kaynaklı zamanla ihtiyaçlara göre yazılım güncellenecektir.

Kırılış yazılımının üzerine aşağıdaki eklemeleri yaparak gelecekte yeni sürümleri kullanılabilir.

- Yazılımın web ortamına uyumlu olarak tasarlanması geniş bir kitle tarafından kullanımını sağlayacaktır.
- Yazılımımız her kullanıcı için aynı arayüz ile açılmaktadır. Ancak gelecekte bir veri tabanı bağlantısı ile her kullanıcıya önceki çalışmalarında gösterilerek çalışmasına olanak sağlayacaktır.
- Beton dayanım makinesi verileri Excel dosyasına kaydetmektedir. Kullanıcı ise Kırılış yazılımına bu Excel dosyasını yüklemektedir.

Yazılımın en büyük dezavantajı budur. Gelecekte makinenin direkt Yazılıma bağlantısı sağlanarak kullanıcıya hatasız ve hızlı sonuçlar verecektir.

- Gelecekte Beton dayanım makinesinin görüntüsünün alınması Kiriliş yazılımına görsellik katacaktır.
- Daha ileri zamanlarda beton pres cihazının tamamen Kirilis yazılımı üzerinden çalıştırılarak kontrol edilebilinmesi kullanıcılara kolaylık sağlayacaktır.
- Bu çalışma, beton silindir numuneler üzerinde gerçekleştirilen elastik çerçeve ile elde edilen verilerin işlenebilmesini kolaylaştırmak üzere modifiye edilebilir. Bu sayede sadece kiriş numuneler değil silindir numuneler içinde uygulanabilir.

KAYNAKLAR

- Adekunle, A.A., Adejuyigbe, S.B., Arulogun, O.T., 2012. Development of CAD Software for Shaft under various Loading Conditions. *Procedia Engineering*, 38, 1962 – 1983.
- Biczo, M., Pocza, K., 2009. Generating Functional Implementations of Finite State Automata in C# 3.0. *Electronic Notes in Theoretical Computer Science*, 238, 3–12.
- Çomak, B., 2012. farklı beton sınıflarının görüntü işleme yöntemi ile mekanik özelliklerinin belirlenmesi. Süleyman Demirel Üniversitesi, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, 132 s, Isparta.
- Demis, S., Efstathiou, M. P., Papadakis, V. G., 2014. Computer-aided modeling of concrete service life. *Cement & Concrete Composites*, 47, 9–18.
- Gencel, O., Koksall, F., Brostow, W., 2013. Wear minimisation in concrete with haematite. *Materials Research Innovations*, 17(2), 92-97.
- Kozicki J., Donze, F.V., 2008. A new open-source software developed for numerical simulations using discrete modeling methods. *Comput. Methods Appl. Mech. Engrg.* 197, 4429–4443.
- Ming C. A., Ming, L., Zhang, J. P., 2004. Concrete plant operations optimization using combined simulation and genetic algorithms. *Proceedings of the Third International Conference on Machine Learning and Cybernetics August 26-29, Shanghai.*
- Microsoft, 2015. Visual Studio Genel Bakış. Erişim Tarihi: 10.01.2015.
<https://www.visualstudio.com/explore/application-development-vs>
- Mosa, A. M., Taha, M. R., Ismail, A., Rahmat, R. A. O. K., 2013. An Educational Knowledge-Based System For Civil Engineering Students In Cement Concrete Construction Problems. *Procedia - Social and Behavioral Sciences*, 102, 311 – 319.
- Sutçü, M., del Coz Diaz, J. J., Alvarez-Rabanal, F. P., Gencel, O., Akkurt, S., 2014. Thermal performance optimization of hollow clay bricks made up of paper waste. *Energy and Buildings*, 75, 96-108.
- Tanyer, A. M., Pekerçli, M. K., 2008. İnşaat Sektörü İçin Bilgi Teknolojilerindeki Son Gelişmeler. *Türkiye Mühendislik Haberleri*, 451, 21-26.
- Yudong, W., Xiaochun, F., 2009. Numerical Simulation of the Load-Deflection Curve of Layered Steel Fiber Reinforced Concrete Beam. *Third International Symposium on Intelligent Information Technology Application.*

- Zheng, Z., Zhao, J., Guo, H., Yang, L., Yu, X., Fang, W., 2012. Character Segmentation System Based on C# Design and Implementation. *Procedia Engineering*, 29, 4073 – 4078.
- Zhang, T., 2009. Teaching Computational Software to Civil Engineering Students. *First International Workshop on Education Technology and Computer Science*.

EKLER

EK A. Yük Deformasyon Verileri

Satır No	Deformasyon (mm)	Yük (kN)	Satır No	Deformasyon (mm)	Yük (kN)	Satır No	Deformasyon (mm)	Yük (kN)
1	0	0	48	0.512752	22.12564	95	1.00539	26.54127
2	0.005535	0.498882	49	0.515554	22.5517	96	1.013815	26.79506
3	0.028474	1.047556	50	0.52414	23.00479	97	1.017784	26.50156
4	0.038266	1.551214	51	0.526785	23.39402	98	1.022804	26.64241
5	0.04028	2.017541	52	0.550572	23.64791	99	1.05275	25.18582
6	0.054004	2.353007	53	0.562623	23.67413	100	1.066991	24.91858
7	0.069326	2.43625	54	0.566185	24.1409	101	1.073322	25.14843
8	0.075917	2.775426	55	0.579803	24.14473	102	1.090927	25.87382
9	0.090716	3.106207	56	0.592284	24.68263	103	1.09168	25.73005
10	0.102657	3.347211	57	0.598483	24.53935	104	1.093991	25.72734
11	0.107981	3.638237	58	0.603084	24.79597	105	1.098643	25.77278
12	0.124113	3.937701	59	0.614486	24.88113	106	1.107909	26.24777
13	0.134142	4.278072	60	0.624597	25.05501	107	1.136389	26.11351
14	0.143227	5.828038	61	0.639285	25.12843	108	1.142276	24.31683
15	0.144073	5.111083	62	0.645458	25.44672	109	1.152881	24.4698
16	0.164566	6.19739	63	0.662284	25.66643	110	1.159407	23.98927
17	0.171486	6.917048	64	0.667114	25.78375	111	1.171509	25.09587
18	0.192941	7.701144	65	0.690726	25.92925	112	1.182673	25.5137
19	0.197927	8.788947	66	0.692243	26.05316	113	1.18626	25.64852
20	0.201022	7.974529	67	0.692259	26.27	114	1.200498	25.92136
21	0.211939	9.626442	68	0.703918	26.3275	115	1.21373	25.56576
22	0.228815	10.9004	69	0.722006	26.58075	116	1.223459	26.1517
23	0.230771	10.23264	70	0.741592	25.97969	117	1.241336	26.29686
24	0.251562	11.51671	71	0.744789	25.74269	118	1.24997	25.64031
25	0.26321	12.30537	72	0.746012	26.58395	119	1.256309	25.72724
26	0.268372	13.20295	73	0.77175	26.81083	120	1.260435	25.72995
27	0.280944	13.97304	74	0.779118	26.60033	121	1.265344	26.17511
28	0.29196	14.77912	75	0.795742	26.39293	122	1.297772	26.12041
29	0.297736	15.5836	76	0.797779	26.49724	123	1.299877	25.96747
30	0.313097	16.38793	77	0.802501	26.55059	124	1.30519	26.09547
31	0.317733	17.08262	78	0.813629	26.60581	125	1.31042	25.93119
32	0.336002	17.71426	79	0.842931	26.34276	126	1.333216	24.61337
33	0.341507	18.66738	80	0.843851	26.69972	127	1.333236	25.24514
34	0.366313	19.32343	81	0.852415	26.77964	128	1.344672	25.18422
35	0.370246	19.93008	82	0.869534	25.90823	129	1.351533	25.58448
36	0.378712	19.8462	83	0.885131	26.41967	130	1.368774	25.36557
37	0.381571	20.02757	84	0.885178	27.07091	131	1.385752	25.35973
38	0.400159	19.71809	85	0.89172	26.958	132	1.391598	25.20733
39	0.415749	19.79284	86	0.913274	27.13969	133	1.404425	24.82931
40	0.417643	20.05144	87	0.927401	26.77958	134	1.405485	24.22225
41	0.423331	20.36142	88	0.930738	27.29645	135	1.4067	24.79993
42	0.44768	20.65175	89	0.931313	27.27397	136	1.437763	25.34817
43	0.451516	21.23637	90	0.94359	26.8955	137	1.443085	24.96868
44	0.46042	21.23987	91	0.959493	27.25234	138	1.460158	23.18555
45	0.461601	20.54775	92	0.970246	26.96849	139	1.464108	22.95493
46	0.47893	21.77861	93	0.994364	25.86969	140	1.468952	23.50493
47	0.494095	21.96309	94	0.998474	27.64273	141	1.493969	24.07275

Satır No	Deformasyon (mm)	Yük (kN)
142	1.510189	24.52528
143	1.512209	24.87052
144	1.52594	25.36046
145	1.528467	25.27708
146	1.531797	25.44838
147	1.550161	25.53456
148	1.552977	25.08599
149	1.573901	25.0697
150	1.586744	24.80035
151	1.587689	24.9743
152	1.601957	24.13588
153	1.609681	24.40319
154	1.612619	24.87922
155	1.616219	23.38825
156	1.643294	23.13897
157	1.664125	24.12135
158	1.667216	23.48414
159	1.667912	24.2779
160	1.695839	23.17568
161	1.708943	23.58596
162	1.717357	24.05437
163	1.727517	23.28333
164	1.743192	24.34327
165	1.745082	24.1865
166	1.752634	23.65659
167	1.763042	24.12799
168	1.779159	24.4667
169	1.789548	24.62556
170	1.789969	23.73521
171	1.798686	23.12052
172	1.81484	23.62429
173	1.834444	24.22321
174	1.83704	24.13737
175	1.854704	24.50484
176	1.857808	24.44203
177	1.86906	23.00147
178	1.87267	22.85841
179	1.884151	23.35205
180	1.897538	23.74057
181	1.909248	23.99881
182	1.923214	23.96321
183	1.933834	24.11532
184	1.94443	24.26006
185	1.95523	24.28102
186	1.961132	24.23986
187	1.967994	24.13193
188	1.972512	23.69267
189	1.99693	23.02703
190	2.009961	23.2783
191	2.020229	23.37352
192	2.030647	23.32175

Satır No	Deformasyon (mm)	Yük (kN)
193	2.052893	22.86517
194	2.05367	22.9769
195	2.05648	23.40011
196	2.071566	22.47584
197	2.072824	22.79192
198	2.082715	22.09673
199	2.090772	22.27363
200	2.118875	22.10264
201	2.124752	22.27035
202	2.125261	22.35644
203	2.140079	22.37578
204	2.160886	20.40242
205	2.164312	22.41384
206	2.168632	20.31035
207	2.186412	20.74206
208	2.193715	21.07387
209	2.223611	21.09611
210	2.224742	21.14876
211	2.23329	21.26177
212	2.236279	20.74929
213	2.238971	21.52235
214	2.265135	20.87026
215	2.285169	20.73122
216	2.288488	20.80425
217	2.301288	20.74731
218	2.311525	18.11612
219	2.319965	21.19743
220	2.322099	18.71684
221	2.336327	19.4676
222	2.341159	19.04814
223	2.362545	19.92179
224	2.373857	20.11486
225	2.386106	20.48191
226	2.391779	20.2409
227	2.404448	20.53569
228	2.417429	20.28004
229	2.420429	20.67608
230	2.42972	20.86856
231	2.435446	20.77549
232	2.458808	18.39653
233	2.464909	18.53037
234	2.485319	18.52332
235	2.486891	18.39656
236	2.496356	19.15905
237	2.506873	18.84479
238	2.512089	18.98644
239	2.520241	19.23067
240	2.52174	19.14646
241	2.542513	19.67205
242	2.551437	19.43404
243	2.580457	18.26019

Satır No	Deformasyon (mm)	Yük (kN)
244	2.599663	18.94185
245	2.600061	18.57542
246	2.602278	18.90397
247	2.604807	18.5125
248	2.613285	18.72033
249	2.628292	18.92876
250	2.642283	18.54696
251	2.654441	18.45565
252	2.655625	18.21585
253	2.678963	18.42283
254	2.699605	17.36977
255	2.705749	18.17271
256	2.70988	17.78825
257	2.716894	17.86264
258	2.723213	17.92192
259	2.744461	17.8793
260	2.75905	17.71712
261	2.773598	17.81989
262	2.773768	17.67482
263	2.779291	18.35664
264	2.801966	18.14504
265	2.811189	17.38374
266	2.814282	17.3294
267	2.819088	17.51357
268	2.832533	17.34601
269	2.84826	17.19725
270	2.869384	17.83969
271	2.869501	18.16783
272	2.884483	17.4917
273	2.889808	18.32833
274	2.904554	17.66425
275	2.911938	17.37645
276	2.913169	17.1724
277	2.916886	17.52252
278	2.956508	16.98288
279	2.960477	17.21357
280	2.976005	16.69145
281	2.986589	16.8915
282	2.994516	17.0349
283	3.004296	17.29902
284	3.008819	17.33892
285	3.023494	17.2872
286	3.070052	16.76623
287	3.072212	16.52944
288	3.083425	16.31413
289	3.083654	16.68168
290	3.085838	16.49168
291	3.091151	17.23418
292	3.094395	16.8395
293	3.105023	17.39265
294	3.111623	17.11057

Satır No	Deformasyon (mm)	Yük (kN)
295	3.128969	17.29658
296	3.129096	17.49696
297	3.144664	17.18948
298	3.14967	16.23009
299	3.159086	16.32993
300	3.163821	16.01077
301	3.180728	16.69845
302	3.206198	15.2652
303	3.210099	15.19758
304	3.234274	15.49679
305	3.237126	15.87254
306	3.241781	15.45
307	3.243763	15.74867
308	3.255733	16.07249
309	3.258874	16.81474
310	3.269125	15.10633
311	3.29172	15.25496
312	3.298872	14.90541
313	3.323687	14.69875
314	3.324809	14.52085
315	3.342693	14.9341
316	3.351263	15.31771
317	3.357828	14.56376
318	3.373354	15.44643
319	3.396827	15.71068
320	3.399554	14.5139
321	3.418175	14.93967
322	3.427769	14.90004
323	3.436212	14.88388
324	3.437472	15.00375
325	3.439592	14.69117
326	3.458189	14.77406
327	3.464823	15.32802
328	3.47507	15.14339
329	3.482911	14.89322
330	3.499317	15.20646
331	3.499706	14.68486
332	3.504588	14.41645
333	3.523952	13.70169
334	3.524231	14.20507
335	3.541493	14.43501
336	3.543771	14.38103
337	3.552166	14.73539
338	3.58269	14.64085
339	3.583784	13.83209
340	3.585366	14.58189
341	3.612697	13.70299
342	3.624773	13.55893
343	3.647677	13.66071
344	3.648257	14.20294
345	3.675366	14.39512

Satır No	Deformasyon (mm)	Yük (kN)
346	3.681646	14.05269
347	3.683138	14.36584
348	3.683581	14.55723
349	3.685752	14.28751
350	3.695909	14.64102
351	3.724958	14.35792
352	3.731573	13.65488
353	3.741792	13.67988
354	3.748775	13.14527
355	3.752868	13.5023
356	3.787196	14.19337
357	3.791075	14.12509
358	3.804399	14.39531
359	3.804803	14.32237
360	3.815552	14.55806
361	3.820804	14.90062
362	3.823773	14.18089
363	3.840592	14.19491
364	3.862472	13.20917
365	3.871463	14.76334
366	3.871603	14.66501
367	3.875677	13.11833
368	3.882553	13.25751
369	3.8927	13.47531
370	3.938264	14.00364
371	3.945421	13.22658
372	3.955831	13.60666
373	3.96972	13.99589
374	3.971069	13.54908
375	3.977598	13.08882
376	3.982846	14.22478
377	3.988327	14.0186
378	3.991782	13.8163
379	4.015887	13.34378
380	4.03131	13.63351
381	4.040701	13.05228
382	4.059743	13.62117
383	4.061123	13.27031
384	4.078181	12.57777
385	4.094012	13.51523
386	4.096725	13.60157
387	4.099087	13.42014
388	4.099492	13.40952
389	4.119464	13.97049
390	4.119645	13.44101
391	4.150171	14.16372
392	4.151668	13.91391
393	4.15219	14.39844
394	4.168868	13.08381
395	4.179323	13.73882
396	4.194047	13.04491

Satır No	Deformasyon (mm)	Yük (kN)
397	4.196924	12.73534
398	4.216579	11.82281
399	4.232595	11.49319
400	4.238153	12.99729
401	4.244703	11.75244
402	4.249394	12.21857
403	4.277269	11.84944
404	4.283301	12.17484
405	4.300106	12.09854
406	4.324257	11.57543
407	4.32441	11.5341
408	4.336438	12.47203
409	4.336548	11.60637
410	4.340831	12.16611
411	4.354085	12.24721
412	4.387132	12.63748
413	4.39522	13.17729
414	4.398531	12.76928
415	4.402996	13.09833
416	4.408663	12.65364
417	4.410395	13.15293
418	4.415104	12.48034
419	4.427467	11.34491
420	4.441663	10.82214
421	4.459384	11.17826
422	4.468554	11.84755
423	4.476875	11.51378
424	4.485461	11.94742
425	4.494294	12.08301
426	4.512863	12.38443
427	4.529905	12.75881
428	4.531308	12.10881
429	4.538873	13.10112
430	4.549973	13.0282
431	4.564923	12.53963
432	4.57705	11.73396
433	4.581896	12.8468
434	4.588824	11.66791
435	4.614949	12.14598
436	4.621403	12.46464
437	4.626628	12.18994
438	4.630579	12.32955
439	4.659536	11.59598
440	4.669261	12.04517
441	4.673421	11.96579
442	4.692994	11.78767
443	4.693932	12.26493
444	4.69411	11.40175
445	4.714789	11.86695
446	4.733117	12.42551
447	4.742975	12.12249

Satır No	Deformasyon (mm)	Yük (kN)
448	4.758038	12.13561
449	4.758296	11.79271
450	4.759393	11.50851
451	4.789927	11.55953
452	4.80183	12.33248
453	4.824265	11.35641
454	4.832451	11.627
455	4.832656	12.25813
456	4.840067	11.45327
457	4.849624	11.00954
458	4.86134	11.15734
459	4.867845	11.16114
460	4.872697	11.05111
461	4.894053	11.61709
462	4.896366	11.94353
463	4.905108	11.40303
464	4.913268	11.95858
465	4.915719	11.92268
466	4.928023	11.63323
467	4.950742	11.84763
468	4.968986	11.34732
469	4.979357	11.84577
470	4.986191	11.6659
471	4.98822	11.6653
472	4.991584	10.96262
473	5.006301	11.64531
474	5.007032	11.38708
475	5.023972	11.32151
476	5.039474	9.654703
477	5.061904	10.60967
478	5.070957	10.16399
479	5.085219	10.63082
480	5.095463	10.22697
481	5.104614	10.88523
482	5.112308	10.94206
483	5.113261	10.88745
484	5.115234	10.57296
485	5.128865	11.10132
486	5.154876	11.06666
487	5.157522	11.51311
488	5.17078	11.43991
489	5.176287	11.70189
490	5.179026	12.0028
491	5.193425	12.12571
492	5.21966	10.44557
493	5.231158	10.77295
494	5.235191	11.20278
495	5.235754	10.90836
496	5.253064	11.45597
497	5.257428	11.41546
498	5.276734	11.74044

Satır No	Deformasyon (mm)	Yük (kN)
499	5.287352	10.56882
500	5.295534	11.05713
501	5.301754	10.77791
502	5.33056	11.09016
503	5.354845	11.47521
504	5.35778	11.17647
505	5.358928	11.15192
506	5.360516	11.53123
507	5.363834	10.88987
508	5.40002	11.06991
509	5.408961	11.47062
510	5.410788	11.53466
511	5.419548	11.05925
512	5.443473	11.28554
513	5.453748	10.14754
514	5.459114	10.23464
515	5.459861	11.08586
516	5.466359	11.2206
517	5.470213	10.5882
518	5.492922	10.79871
519	5.494102	11.40343
520	5.49985	11.27722
521	5.515576	11.0926
522	5.518879	10.65349
523	5.520438	10.88092
524	5.542857	11.03959
525	5.550324	11.51653
526	5.550956	11.36137
527	5.585439	11.34472
528	5.593342	11.39731
529	5.599734	11.58719
530	5.611005	11.75298
531	5.614012	11.49761
532	5.619659	11.61724
533	5.637102	10.61033
534	5.651584	9.807614
535	5.657204	10.38982
536	5.676046	10.24468
537	5.683175	10.60971
538	5.69891	10.84049
539	5.718364	11.09804
540	5.719733	10.72222
541	5.727808	10.49398
542	5.746385	10.48898
543	5.757319	10.89825
544	5.765612	10.5957
545	5.774921	11.37099
546	5.782283	10.95619
547	5.787517	11.47818
548	5.79383	10.35607
549	5.808012	10.11608

Satır No	Deformasyon (mm)	Yük (kN)
550	5.838514	10.27764
551	5.84429	10.54071
552	5.847848	10.38502
553	5.852427	10.01743
554	5.875337	10.30856
555	5.894358	11.04989
556	5.896434	10.87275
557	5.899995	9.81088
558	5.902639	10.458
559	5.911202	11.3845
560	5.922911	11.46469
561	5.947278	9.654338
562	5.955391	10.18001
563	5.960633	9.893615
564	5.962457	11.246
565	5.972775	10.56273
566	5.988372	10.99073
567	6.003178	11.10364
568	6.010864	10.38372
569	6.032085	10.35847
570	6.051284	10.59764
571	6.076311	11.42854
572	6.076374	11.15879
573	6.0783	11.07413
574	6.081769	10.80935
575	6.089968	10.48348
576	6.100363	11.48947
577	6.101566	11.23656
578	6.125084	12.04646
579	6.125744	11.75049
580	6.133526	11.81957
581	6.152432	11.40433
582	6.155092	11.56472
583	6.177043	11.43224
584	6.178992	11.10032
585	6.1864	11.59964
586	6.205066	11.81123
587	6.213864	11.86103
588	6.224429	10.64967
589	6.233437	11.54339
590	6.242576	9.943038
591	6.253548	8.98561
592	6.258448	9.373257
593	6.262363	9.528393
594	6.279479	9.439834
595	6.287285	9.662045
596	6.315065	9.57651
597	6.315627	9.47031
598	6.350345	9.610839
599	6.354437	9.492735
600	6.354995	9.619732

Satır No	Deformasyon (mm)	Yük (kN)
601	6.363463	9.670163
602	6.366156	10.16724
603	6.391623	9.471384
604	6.400571	9.797337
605	6.407807	9.704514
606	6.432763	9.81077
607	6.438008	9.39319
608	6.449673	8.294059
609	6.453821	8.629591
610	6.45764	8.569061
611	6.497398	9.492845
612	6.50475	9.158454
613	6.505141	9.327277
614	6.512748	9.188099
615	6.526876	8.940462
616	6.527917	9.492325
617	6.531607	9.496001
618	6.55174	9.586255
619	6.556943	9.671182
620	6.57354	9.085819
621	6.589328	8.939642
622	6.599254	9.247101
623	6.60449	9.864302
624	6.615915	9.534816
625	6.638981	10.04288
626	6.642592	9.309902
627	6.65139	9.260966
628	6.656953	9.999039
629	6.657135	9.790283
630	6.668531	9.719531
631	6.676621	9.75922
632	6.680739	7.608798
633	6.707297	8.207006
634	6.712275	8.396051
635	6.718474	7.704168
636	6.719884	7.710059
637	6.731774	8.685593
638	6.746918	8.301833
639	6.749979	8.807606
640	6.78357	9.156804
641	6.793155	8.81136
642	6.805824	8.276795
643	6.814824	8.954348
644	6.833232	8.792125
645	6.840723	9.026075
646	6.845463	8.792014
647	6.847269	9.428406
648	6.856317	9.165885
649	6.869998	9.226293
650	6.87488	9.042985
651	6.899848	9.378982

Satır No	Deformasyon (mm)	Yük (kN)
652	6.91752	9.748512
653	6.919568	9.531394
654	6.930313	9.756551
655	6.936107	9.57961
656	6.937392	9.742277
657	6.944164	8.904825
658	6.975126	8.53297
659	6.978298	8.088459
660	6.986219	8.363316
661	6.98847	8.605096
662	6.996698	8.604453
663	7.009367	9.093615
664	7.026504	8.982288
665	7.042652	8.351068
666	7.060197	9.509025
667	7.064843	8.784207
668	7.077988	9.075122
669	7.079201	8.747552
670	7.09208	9.706386
671	7.10036	9.391263
672	7.113885	10.09487
673	7.117852	9.662444
674	7.13603	8.798581
675	7.161661	8.5839
676	7.163904	8.886896
677	7.166832	9.02829
678	7.187024	8.855867
679	7.209755	8.948391
680	7.214649	9.15027
681	7.233434	8.668406
682	7.23354	8.702569
683	7.254662	8.141315
684	7.287002	9.939849
685	7.293465	7.920809
686	7.300726	8.353017
687	7.301982	8.345133
688	7.303352	8.551697
689	7.321597	8.545008
690	7.335539	8.438642
691	7.341003	8.113586
692	7.354342	8.754141
693	7.368687	7.937963
694	7.404791	8.31335
695	7.409267	9.463455
696	7.414336	8.881581
697	7.414408	8.12332
698	7.41925	7.915028
699	7.447491	8.878447
700	7.448607	8.871105
701	7.456478	8.069998
702	7.463476	7.89814

Satır No	Deformasyon (mm)	Yük (kN)
703	7.47283	8.638041
704	7.485704	8.582261
705	7.504299	8.518707
706	7.50849	8.912943
707	7.51514	8.771328
708	7.515376	8.841526
709	7.52013	8.933274
710	7.521283	9.24729
711	7.570664	8.637875
712	7.574465	7.699428
713	7.580465	7.966578
714	7.581074	7.733503
715	7.584272	8.883076
716	7.596611	9.591017
717	7.602572	8.416272
718	7.604828	7.570139
719	7.6313	8.09342
720	7.639291	7.709107
721	7.656452	8.281745
722	7.671563	8.521254
723	7.672869	7.450107
724	7.676194	8.274469
725	7.676274	8.488796
726	7.7007	8.090817
727	7.714112	7.822981
728	7.732943	8.656091
729	7.735735	7.096447
730	7.759993	7.616583
731	7.764565	7.718896
732	7.766062	7.76327
733	7.773455	7.442234
734	7.795762	7.55458
735	7.79691	7.142282
736	7.80824	6.958664
737	7.811936	8.384966
738	7.823458	7.139436
739	7.845196	8.013045
740	7.850689	7.377229
741	7.866799	8.172832
742	7.871286	7.875029
743	7.877456	7.483174
744	7.894669	7.859525
745	7.939203	8.59545
746	7.942098	7.860389
747	7.951025	8.389639
748	7.953543	8.4312
749	7.957886	8.510446
750	7.959216	7.693448
751	7.979612	7.323055
752	7.980127	7.916678
753	7.988748	7.59572

Satır No	Deformasyon (mm)	Yük (kN)
754	7.99062	8.561453
755	7.992893	8.266485
756	8.006333	8.125446
757	8.006522	8.419683
758	8.029273	8.788625
759	8.043555	7.004887
760	8.054196	7.169901
761	8.085666	7.522709
762	8.094381	7.594956
763	8.094805	7.500239
764	8.09807	8.639525
765	8.114797	7.471491
766	8.127521	7.637901
767	8.149412	7.192503
768	8.164929	8.066444
769	8.171124	7.773857
770	8.18243	7.429908
771	8.18385	7.714522
772	8.184551	8.23014
773	8.191185	7.166723
774	8.214753	7.429332
775	8.224076	7.496984
776	8.244734	7.592453
777	8.26707	7.779615
778	8.272658	6.962562
779	8.289356	8.105911
780	8.292337	6.888144
781	8.308712	7.123246
782	8.308932	8.140673
783	8.315401	7.71605
784	8.315663	6.431075
785	8.316952	7.052073
786	8.321475	8.046986
787	8.344014	8.303915
788	8.34757	6.923803
789	8.34772	7.022317
790	8.361387	7.054244
791	8.37781	7.631987
792	8.380048	7.448369
793	8.388106	7.514093
794	8.408019	7.748552
795	8.410009	7.672961
796	8.426136	7.788297
797	8.456919	7.980288
798	8.460689	8.278312
799	8.46679	8.300128
800	8.479958	8.292686
801	8.486515	7.530582
802	8.493515	8.355431
803	8.500679	7.525521
804	8.513229	7.229402

Satır No	Deformasyon (mm)	Yük (kN)
805	8.542144	7.503074
806	8.544276	7.487748
807	8.555029	7.882205
808	8.559965	8.062667
809	8.573996	7.969867
810	8.590673	7.910399
811	8.597838	7.752594
812	8.609569	7.776271
813	8.618612	6.734116
814	8.651632	6.286536
815	8.663949	6.473765
816	8.672927	7.004433
817	8.679448	7.563538
818	8.681203	7.352069
819	8.682857	7.179945
820	8.68525	6.835509
821	8.696869	6.839352
822	8.707524	6.882054
823	8.711228	7.024056
824	8.72986	6.761457
825	8.754154	6.675589
826	8.772061	8.277736
827	8.777682	6.957567
828	8.792244	6.805765
829	8.796428	7.69419
830	8.79877	7.22701
831	8.813506	7.380097
832	8.817585	7.389964
833	8.824955	7.423297
834	8.825858	7.414238
835	8.839107	7.266777
836	8.856438	7.559718
837	8.864481	7.075373
838	8.890699	7.315691
839	8.893993	7.433086
840	8.912831	6.786374
841	8.916554	7.206323
842	8.932101	7.871374
843	8.934646	7.319035
844	8.939594	7.42663
845	8.958072	7.843346
846	8.960276	7.971307
847	8.972936	8.066444
848	8.984661	7.547559
849	8.995627	7.398812
850	9.0098	7.08297
851	9.021008	7.526274
852	9.032413	8.027496
853	9.036331	8.041483
854	9.044641	7.427062
855	9.052001	7.938262

Satır No	Deformasyon (mm)	Yük (kN)
856	9.059817	7.755573
857	9.066404	7.126037
858	9.091374	6.413899
859	9.098366	7.260055
860	9.146856	6.298995
861	9.150895	6.935741
862	9.161226	6.385693
863	9.167911	6.274211
864	9.176175	7.108628
865	9.177107	6.317023
866	9.181416	6.859108
867	9.196436	6.480985
868	9.198761	6.325905
869	9.212871	6.915719
870	9.2139	6.932739
871	9.227782	6.111633
872	9.235194	6.168587
873	9.267567	6.903759
874	9.281268	6.764625
875	9.285274	6.96894
876	9.293012	7.355513
877	9.305012	6.64796
878	9.324532	6.642766
879	9.324873	5.760963
880	9.330313	5.958191
881	9.345792	7.651001
882	9.346903	5.865557
883	9.364825	6.339304
884	9.372577	5.890917
885	9.382986	6.385228
886	9.405966	6.436423
887	9.419441	6.561748
888	9.442393	6.127303
889	9.446074	6.203415
890	9.447767	5.945401
891	9.44814	6.5033
892	9.467157	6.298065
893	9.475375	5.73413
894	9.483962	6.968232
895	9.48694	6.162374
896	9.495388	6.990878
897	9.507155	6.591803
898	9.514175	6.856362
899	9.515599	6.97007
900	9.534531	7.325281
901	9.548573	7.220299
902	9.556481	7.264673
903	9.556945	7.470417
904	9.562022	7.22093
905	9.575682	6.741768
906	9.612191	6.509457

Satır No	Deformasyon (mm)	Yük (kN)
907	9.619612	6.93264
908	9.626621	6.69457
909	9.628883	6.452758
910	9.640775	6.803937
911	9.655975	6.788323
912	9.657117	7.069637
913	9.677125	7.277673
914	9.68502	6.979594
915	9.707883	7.381249
916	9.711573	5.708051
917	9.714362	7.346875
918	9.717452	5.744618
919	9.729355	6.18213

Satır No	Deformasyon (mm)	Yük (kN)
920	9.737695	7.325845
921	9.745924	5.837086
922	9.777895	5.382718
923	9.80559	6.15109
924	9.832923	6.259383
925	9.83372	5.906487
926	9.840186	5.388078
927	9.851262	5.492772
928	9.865008	6.059618
929	9.876614	5.872899
930	9.877773	6.253779
931	9.880492	6.303701
932	9.882259	6.494939

Satır No	Deformasyon (mm)	Yük (kN)
933	9.891545	6.084978
934	9.899628	6.134789
935	9.926495	5.961192
936	9.940602	6.412825
937	9.959596	6.420964
938	9.977909	6.69467
939	9.979794	6.7287
940	9.983786	6.845432
941	9.991562	6.284078
942	9.992532	6.779641
943	10.014254	6.393301
944	10.016159	6.853195

EK B. Yazılım Kodları

ANA PENCERE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
using System.Windows.Forms;
using Series = System.Windows.Forms.DataVisualization.Charting.Series;
namespace Kirilis
{
    public partial class AnaPencere : Form
    {
        List<Nokta> noktalar = null;
        Hesaplayici hesaplayici = new Hesaplayici();
        GirdiSeti girdiSeti = new GirdiSeti();
        CiktiSeti ciktiSeti = new CiktiSeti();
        string dosyaYolu = "";
        public AnaPencere()
        {
            InitializeComponent();
        }
        private void AnaPencere_Load(object sender, EventArgs e)
        {
            FormHazirla();
        }
        public void GrafikHazirla()
        {
            grafik.Series[0].Points.Clear();
            grafik.DataSource = noktalar;
            grafik.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
            grafik.Series[0].XValueType =
System.Windows.Forms.DataVisualization.Charting.ChartValueType.Double;
            grafik.Series[0].YValueType =
System.Windows.Forms.DataVisualization.Charting.ChartValueType.Double;
            grafik.Series[0].YValueMembers = "Y";
            grafik.Series[0].XValueMember = "X";
            grafik.Series[0].XAxisType = System.Windows.Forms.DataVisualization.Charting.AxisType.Primary;
            grafik.Series[0].YAxisType = System.Windows.Forms.DataVisualization.Charting.AxisType.Primary;
            grafik.DataBind();
            grafik.Update();
            Refresh();
        }
        public void FormHazirla()
        {
            txtDurum.AppendText("> Form hazırlanıyor.\n");
            // Giriş değerleri için kontrollere bağlanıyor
            txta.DataBindings.Add("Text", girdiSeti, "a");
            txtB.DataBindings.Add("Text", girdiSeti, "B");
            txtD.DataBindings.Add("Text", girdiSeti, "D");
            txtD0.DataBindings.Add("Text", girdiSeti, "d0");
            txtE.DataBindings.Add("Text", girdiSeti, "E");
            txtft.DataBindings.Add("Text", girdiSeti, "ft");
            txtL.DataBindings.Add("Text", girdiSeti, "L");
            txtS.DataBindings.Add("Text", girdiSeti, "S");
            txtYog.DataBindings.Add("Text", girdiSeti, "yog");
            // Çıkış değerleri için kontrollere bağlanıyor
            txtFL.DataBindings.Add("Text", ciktiSeti, "Fl");
            txtfnet.DataBindings.Add("Text", ciktiSeti, "fnet");
            txtW0.DataBindings.Add("Text", ciktiSeti, "W0");
            txtGf.DataBindings.Add("Text", ciktiSeti, "Gf");
            txtlch.DataBindings.Add("Text", ciktiSeti, "lch");
            chkSehim.Checked = true;
            txtDurum.AppendText("> Form hazırlanması tamamlandı.\n");
        }
    }
}
```

```

// Hesaplama sonuçlarını temizler
private void temizle()
{
    ciktiSeti.ilkle();
    Refresh();
}
private void chkSehim_CheckedChanged(object sender, EventArgs e)
{
    if (chkSehim.Checked == true)
    {
        girdiSeti.d0 = 10.0;
    }
    else
    {
        girdiSeti.d0 = 4.0;
    }
}
private void btnGozat_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "Excel Dosyaları (*.xls;*.xlsx)|*.xls;*.xlsx";
    DialogResult sonuc = ofd.ShowDialog();
    if (sonuc == System.Windows.Forms.DialogResult.OK)
    {
        txtDosyaYolu.Text = ofd.FileName;
        dosyaYolu = ofd.FileName;
    }
}
private void btnHesapla_Click(object sender, EventArgs e)
{
    grbKontrol.Enabled = false;
    hesapla();
    grbKontrol.Enabled = true;
}
private void veriYukle()
{
    if (dosyaYolu != "")
    {
        txtDurum.AppendText("> Veriler dosyadan aktarılıyor.\n");
        txtDurum.AppendText("> Girdi dosyası: " + dosyaYolu + "\n");
        //noktalar = DosyaIsleyici.IceAktor(dosyaYolu);
        DataTable dt = DosyaIsleyici.oleddbIceAktor(dosyaYolu, chkBaslik.Checked);
        dgwNoktalar.DataSource = dt;
        noktalar = VeriHazirlayici.NoktaListesiOlustur(dt);
        if (noktalar == null)
        {
            txtDurum.AppendText("> Dosyadan okuma sırasında sorun oluştu.\n");
            txtDurum.AppendText("> Veri aktarımı başarısız oldu.\n");
            return;
        }
        else
        {
            GrafikHazirla();
            girdiSeti.noktalar = noktalar;
            txtDurum.AppendText("> Veri aktarımı tamamlandı.\n");
        }
    }
}

private void hesapla()
{
    if (noktalar != null)
    {
        txtDurum.AppendText("> Hesaplama işlemi başladı.\n");
        int sonuc = hesaplayici.TumunuHesapla(ref girdiSeti, ref ciktiSeti);
        if (sonuc == 0)
        {
            txtDurum.AppendText("> Hesaplama işlemi başarıyla sonlandı.\n");
        }
        else
        {
            txtDurum.AppendText("> Hesaplama işlemi sırasında hata oluştu. Lütfen giriş verilerinizi kontrol edin.\n");
        }
    }
    else
    {
        txtDurum.AppendText("> Hesaplama için giriş verisi bulunamadı. Lütfen bir dosyadan giriş değerlerini yükleyin.\n");
    }
}
private void btnYukle_Click(object sender, EventArgs e)
{
    grbKontrol.Enabled = false;
    veriYukle();
    grbKontrol.Enabled = true;
}

```

```

private void btnTemizle_Click(object sender, EventArgs e)
{
    grbKontrol.Enabled = false;
    temizle();
    grbKontrol.Enabled = true;
}
private void btnKapat_Click(object sender, EventArgs e)
{
    this.Close();
}
private void txtfnet_TextChanged(object sender, EventArgs e)
{
}
}
}

```

ÇIKTI SETİ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;
// CiktiSeti sınıfı
// Bu sınıf hesaplama çıktılarının tutulması için kullanılır.
namespace Kirilis
{
    class CiktiSeti : INotifyPropertyChanged
    {
        // Değişken isimleri olarak gerçek sembollere yakın ifadeler seçtim.
        // Bunları "refactoring" ile kolayca değiştirmek mümkündür.
        private double _Fl;
        private double _fnet;
        private double _W0;
        private double _Gf;
        private double _m;
        private double _lch;
        private double _K;
        public double Fl
        {
            get
            {
                return _Fl;
            }
            set
            {
                _Fl = value;
                InvokePropertyChanged(new PropertyChangedEventArgs("Fl"));
            }
        }
        public double fnet
        {
            get
            {
                return _fnet;
            }
            set
            {
                _fnet = value;
                InvokePropertyChanged(new PropertyChangedEventArgs("fnet"));
            }
        }

        public double W0
        {

```

```

        get
        {
            return _W0;
        }
        set
        {
            _W0 = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("W0"));
        }
    }
    public double Gf
    {
        get
        {
            return _Gf;
        }
        set
        {
            _Gf = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("Gf"));
        }
    }
}

public double m
{
    get
    {
        return _m;
    }
    set
    {
        _m = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("m"));
    }
}

public double lch
{
    get
    {
        return _lch;
    }
    set
    {
        _lch = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("lch"));
    }
}

public double K
{
    get
    {
        return _K;
    }
    set
    {
        _K = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("K"));
    }
}

public CiktiSeti()
{
    ilkle();
}

public void ilkle()
{
    F1 = 0.0;
    fnet = 0.0;
    W0 = 0.0;
    Gf = 0.0;
}

```

```

        m = 0.0;
        lch = 0.0;
        K = 0.0;
    }
    #region INotifyPropertyChanged gerekleřtirimi
    public event PropertyChangedEventHandler PropertyChanged;
    public void InvokePropertyChanged(PropertyChangedEventArgs e)
    {
        PropertyChangedEventHandler handler = PropertyChanged;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    #endregion
}

```

DOSYA İřLEYİCİ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//using Excel = Microsoft.Office.Interop.Excel;
using System.Data.OleDb;
using System.Data;

// DosyaIsleyici sınıfı
// Dosyadan veri aktarma (import) iřlevini saęlar.

namespace Kirilis
{
    class DosyaIsleyici
    {

        //public static List<Nokta> IceAktar(string dosyaAdi)
        //{

        //    Excel.Application ExcelUygulamasi = new Excel.Application();

        //    try
        //    {

        //        List<Nokta> degerler = new List<Nokta>();
        //        int i;

        //        ExcelUygulamasi.Workbooks.Open(dosyaAdi);
        //        for (i = 2; i < ExcelUygulamasi.Workbooks[1].Worksheets[1].Rows.Count; ++i)
        //        {
        //            // rastlanacak ilk boř satıra kadar okuma yapılacak
        //            if (ExcelUygulamasi.Workbooks[1].Worksheets[1].Cells[i, 1].Value == null)
        //            {

        //                break;

        //            }
        //            else
        //            {
        //                try
        //                {
        //                    double x = (double)ExcelUygulamasi.Workbooks[1].Worksheets[1].Cells[i,1].Value;
        //                    double y = (double)ExcelUygulamasi.Workbooks[1].Worksheets[1].Cells[i,2].Value;
        //                    degerler.Add(new Nokta(x, y));
        //                }
        //            }
        //        }
        //    }
        //}
    }
}

```

```

//      }
//      catch
//      {
//          continue; // okunamayan değerler atlanıyor
//      }
//  }

//  }

//  ExcelUygulamasi.Workbooks.Close();
//  return degerler;

//  }
//  catch (Exception e)
//  {
//      Console.WriteLine("İstisna oluştu. {0}\nBilgi: {1}", e.StackTrace, e.Message);
//      return null; // dosyadan okuma başarısız olduğunda boş veri kümesi döndürülür.
//  }
//}
// Excel dosyasındaki verileri OleDb kullanarak içe aktaran metot
public static DataTable oledbIceAktar(string dosyaAdi, bool baslikSatiriVar)
{
    DataTable dt = new DataTable();
    try
    {
        string cstr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + @dosyaAdi + ";Extended
Properties=Excel 12.0;HDR=";
        if (baslikSatiriVar)
        {
            cstr += "Yes;";
        }
        else
        {
            cstr += "No;";
        }
        OleDbConnection con = new OleDbConnection(cstr);
        con.Open();
        string ilkSayfa = con.GetOleDbSchemaTable(OleDbSchemaGuid.Tables,
null).Rows[0]["TABLE_NAME"].ToString();
        //değişken içindeki tırnaklar çıkarılıyor
        ilkSayfa = ilkSayfa.Replace("'", "");
        OleDbDataAdapter da = new OleDbDataAdapter("select * from [" + ilkSayfa + "A:B]", con);
        da.Fill(dt);
    }
    catch (Exception e)
    {
        Console.WriteLine("İstisna oluştu. {0}\nBilgi: {1}", e.StackTrace, e.Message);
        System.Windows.Forms.MessageBox.Show("İstisna oluştu. " + e.StackTrace + "\nBilgi: " +
e.Message);
    }
    return dt;
}
}
}

```

GİRDİ SETİ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;
// GirdiSeti sınıfı
// Bu sınıf girdi verilerini tutar.
namespace Kirilis
{
    class GirdiSeti : INotifyPropertyChanged

```

```

{
    private double _B;
    private double _D;
    private double _a;
    private double _L;
    private double _S;
    private double _d;
    private double _d0;
    private double _E;
    private double _ft;
    private double _g;
    public List<Nokta> noktalar { get; set; }
    public double B
    {
        get
        {
            return _B;
        }
        set
        {
            _B = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("B"));
        }
    }
    public double D
    {
        get
        {
            return _D;
        }
        set
        {
            _D = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("D"));
        }
    }
    public double a
    {
        get
        {
            return _a;
        }
        set
        {
            _a = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("a"));
        }
    }
    public double L
    {
        get
        {
            return _L;
        }
        set
        {
            _L = value;
            InvokePropertyChanged(new PropertyChangedEventArgs("L"));
        }
    }
}

public double S
{
    get
    {
        return _S;
    }
    set
    {
        _S = value;
    }
}

```

```

        InvokePropertyChanged(new PropertyChangedEventArgs("S"));
    }
}
public double yog
{
    get
    {
        return _d;
    }
    set
    {
        _d = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("yog"));
    }
}
public double d0
{
    get
    {
        return _d0;
    }
    set
    {
        _d0 = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("d0"));
    }
}
public double E
{
    get
    {
        return _E;
    }
    set
    {
        _E = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("E"));
    }
}
public double ft
{
    get
    {
        return _ft;
    }
    set
    {
        _ft = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("ft"));
    }
}

public double g
{
    get
    {
        return _g;
    }
    set
    {
        _g = value;
        InvokePropertyChanged(new PropertyChangedEventArgs("g"));
    }
}
public GirdiSeti()
{
    ilkle();
}

```

```

public double yukKatsayisi { get; set; }
public double sehimKatsayisi { get; set; }

public void ilkle()
{
    B = 0.0;
    D = 0.0;
    a = 0.0;
    L = 0.0;
    S = 0.0;
    yog = 0.0;
    d0 = 0.0;
    E = 0.0;
    ft = 0.0;
    //g = 9.80665;
    g = 9810; // mm/s^2
    yukKatsayisi = 1;
    sehimKatsayisi = 1;
}
#region INotifyPropertyChanged gerekleřtirimi
public event PropertyChangedEventHandler PropertyChanged;
public void InvokePropertyChanged(PropertyChangedEventArgs e)
{
    PropertyChangedEventHandler handler = PropertyChanged;
    if (handler != null)
    {
        handler(this, e);
    }
}
#endregion
}
}

```

HESAPLAYICI

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Kirilis
{
    class Hesaplayici
    {
        public Hesaplayici()
        {
        }

        public int TumunuHesapla(ref GirdiSeti girdiSeti, ref CiktiSeti ciktiSeti)
        {
            // Bařarı durumu parametresi
            int sonuc = 0;
            // Giriř parametreleri
            double B = girdiSeti.B;
            double D = girdiSeti.D;
            double a = girdiSeti.a;
            double L = girdiSeti.L;
            double d = girdiSeti.yog*9.81/1000000000;
            double E = girdiSeti.E;
            double ft = girdiSeti.ft;
            double d0 = girdiSeti.d0;
            double S = girdiSeti.S;
            double g = girdiSeti.g;
            // Sonu deęerleri
            double Fl = 0.0;
            double fnet = 0.0;
            double Gf = 0.0;

```

```

double W0 = 0.0;
double m = 0.0;
double K = 0.0;
double lch = 0.0;
double tokluk_gecici = 0.0;
int i;
try
{
    // Fl hesaplama
    Fl = flBul(girdiSeti.noktalar)*girdiSeti.sehimKatsayisi*girdiSeti.yukKatsayisi;
    // net eğilme dayanımı
    fnet = (double)(3*Fl*S)/(2*B*Math.Pow((D-a), 2));
    // 0/0 durumunda sıfıra bölme istisnası fırlatılmaz. Bu durumda değeri elle değiştirmenin daha
    makul olduğunu düşündüm.
    if (double.IsNaN(fnet) || double.IsInfinity(fnet))
    {
        fnet = 0.0;
        sonuc = -1;
    }
    // tokluk hesaplama
    for (i = 0; i < girdiSeti.noktalar.Count - 1; i++)
    {
        if (girdiSeti.noktalar[i].X <= d0)
        {
            tokluk_gecici += Math.Abs(0.5*(girdiSeti.noktalar[i + 1].X -
girdiSeti.noktalar[i].X)*(girdiSeti.noktalar[i + 1].Y + girdiSeti.noktalar[i].Y));
        }
        else
        {
            break;
        }
    }
    W0 = tokluk_gecici*girdiSeti.yukKatsayisi*girdiSeti.sehimKatsayisi;
    K = (double)(L / S) - 1;
    if (double.IsNaN(K) || double.IsInfinity(K))
    {
        K = 0.0;
        sonuc = -1;
    }
    m = (B/1000) * (D/1000) * (S/1000) * d;
    // kırılma enerjisi
    Gf = (double)((W0 + m * (1 - Math.Pow(K, 2)) * g * d0) / (B * (D - a)));
    if (double.IsNaN(Gf) || double.IsInfinity(Gf))
    {
        Gf = 0.0;
        sonuc = -1;
    }
    // karakteristik boy
    lch = (double)(E * Gf / Math.Pow(ft, 2));
    if (double.IsNaN(lch) || double.IsInfinity(lch))
    {
        lch = 0.0;
        sonuc = -1;
    }
}
catch (Exception e)
{
    Console.WriteLine("İstisna oluştu. {0}\nBilgi: {1}", e.StackTrace, e.Message);
    sonuc = -1;
    return sonuc; // başarısız sonlandırma
}
// sonuç değerleri sonuç setine atanıyor
ciktiSeti.Fl = Math.Round(Fl, 2);
ciktiSeti.fnet = Math.Round(fnet, 2);
ciktiSeti.Gf = Math.Round(Gf, 2);
ciktiSeti.K = Math.Round(K, 2);
ciktiSeti.lch = Math.Round(lch, 2);
ciktiSeti.m = Math.Round(m, 2);
ciktiSeti.W0 = Math.Round(W0, 2);

```

```

        return sonuc; // başarılı sonlandırma
    }

    /*
    // Fl hesaplanması; doğru çalıştığından emin olmak gerekli.
    private double flBul(List<Nokta> noktalar)
    {
        double maksimumYuk = 0.0;
        int i;
        for (i = 0; i < noktalar.Count - 1; i++)
        {
            if (noktalar[i].X <= 0.5)
            {
                if (noktalar[i].Y >= maksimumYuk)
                {
                    maksimumYuk = noktalar[i].Y;
                }
            }
            else
            {
                break;
            }
        }
        return maksimumYuk;
    }
    */
    // Alternatif Fl bulma fonksiyonu
    private double flBul(List<Nokta> noktalar)
    {
        double maksimumYuk = 0.0;
        int i;

        for (i = 0; i < noktalar.Count - 1; i++)
        {
            if (noktalar[i].Y >= maksimumYuk)
            {
                maksimumYuk = noktalar[i].Y;
            }
        }
        return maksimumYuk;
    }
}

```

NOKTA

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Kirilis
{
    class Nokta
    {
        private double _x;
        private double _y;
        public double X
        {

```

```

        get
        {
            return _x;
        }
        set
        {
            _x = value;
        }
    }
    public double Y
    {
        get
        {
            return _y;
        }
        set
        {
            _y = value;
        }
    }
    public Nokta()
    {
        _x = 0.0;
        _y = 0.0;
    }
    public Nokta(double x, double y)
    {
        _x = x;
        _y = y;
    }
}
}

```

PROGRAM

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Kirilis
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new AnaPencere());
        }
    }
}

```

VERİ HAZIRLAYICI

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;

```

```

namespace Kirilis
{
    class VeriHazirlayici
    {
        public static List<Nokta> NoktaListesiOlustur(DataTable dt)
        {
            List<Nokta> noktalar = new List<Nokta>();
            foreach(DataRow dr in dt.Rows)
            {
                try
                {
                    noktalar.Add(new Nokta((double)dr[dt.Columns[0].ColumnName],
(double)dr[dt.Columns[1].ColumnName]));
                }
                catch (Exception e)
                {
                    // Excel dosyalarında çok sayıda boş satır oluşmasına neden olan bir biçimlendirme hatası
olabilmektedir.
                    // Bu hatanın çözümü için http://support.microsoft.com/kb/244435 adresine bakılabilir.
                    // Bu hatanın programın işleyişini etkilememesi için boş gelen satırlar gözardı edilmektedir.
                    Console.WriteLine("İstisna oluştu. {0}\nBilgi: {1}", e.StackTrace, e.Message);
                    break;
                }
            }
            return noktalar;
        }
    }
}

```

--- ANA PENCERE DİZAYN

```

namespace Kirilis
{
    partial class AnaPencere
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea1 = new
System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Series series1 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(AnaPencere));

```

```

this.grafik = new System.Windows.Forms.DataVisualization.Charting.Chart();
this.txtDurum = new System.Windows.Forms.TextBox();
this.grbGirisSeti = new System.Windows.Forms.GroupBox();
this.grbMekanik = new System.Windows.Forms.GroupBox();
this.textBox7 = new System.Windows.Forms.TextBox();
this.textBox8 = new System.Windows.Forms.TextBox();
this.textBox9 = new System.Windows.Forms.TextBox();
this.lblE = new System.Windows.Forms.Label();
this.lblft = new System.Windows.Forms.Label();
this.txtD0 = new System.Windows.Forms.TextBox();
this.lblD0 = new System.Windows.Forms.Label();
this.txtE = new System.Windows.Forms.TextBox();
this.txtft = new System.Windows.Forms.TextBox();
this.pbModel = new System.Windows.Forms.PictureBox();
this.grbFiziksel = new System.Windows.Forms.GroupBox();
this.textBox6 = new System.Windows.Forms.TextBox();
this.textBox5 = new System.Windows.Forms.TextBox();
this.textBox4 = new System.Windows.Forms.TextBox();
this.textBox3 = new System.Windows.Forms.TextBox();
this.textBox2 = new System.Windows.Forms.TextBox();
this.textBox1 = new System.Windows.Forms.TextBox();
this.lblB = new System.Windows.Forms.Label();
this.lblS = new System.Windows.Forms.Label();
this.lblYog = new System.Windows.Forms.Label();
this.lblD = new System.Windows.Forms.Label();
this.lbla = new System.Windows.Forms.Label();
this.txtYog = new System.Windows.Forms.TextBox();
this.lblL = new System.Windows.Forms.Label();
this.txtS = new System.Windows.Forms.TextBox();
this.txtB = new System.Windows.Forms.TextBox();
this.txtL = new System.Windows.Forms.TextBox();
this.txtD = new System.Windows.Forms.TextBox();
this.txta = new System.Windows.Forms.TextBox();
this.grbCikisSeti = new System.Windows.Forms.GroupBox();
this.textBox13 = new System.Windows.Forms.TextBox();
this.textBox15 = new System.Windows.Forms.TextBox();
this.textBox10 = new System.Windows.Forms.TextBox();
this.textBox11 = new System.Windows.Forms.TextBox();
this.textBox12 = new System.Windows.Forms.TextBox();
this.txtlch = new System.Windows.Forms.TextBox();
this.txtGf = new System.Windows.Forms.TextBox();
this.txtW0 = new System.Windows.Forms.TextBox();
this.txtfnet = new System.Windows.Forms.TextBox();
this.txtFL = new System.Windows.Forms.TextBox();
this.lblGf = new System.Windows.Forms.Label();
this.lblch = new System.Windows.Forms.Label();
this.lblW0 = new System.Windows.Forms.Label();
this.lblfnet = new System.Windows.Forms.Label();
this.lblFL = new System.Windows.Forms.Label();
this.grbMesajlar = new System.Windows.Forms.GroupBox();
this.grbKontrol = new System.Windows.Forms.GroupBox();
this.lblDeneyTipi = new System.Windows.Forms.Label();
this.chkSehim = new System.Windows.Forms.RadioButton();
this.chkCMOD = new System.Windows.Forms.RadioButton();
this.btnTemizle = new System.Windows.Forms.Button();
this.btnHesapla = new System.Windows.Forms.Button();
this.btnYukle = new System.Windows.Forms.Button();
this.btnGozat = new System.Windows.Forms.Button();
this.txtDosyaYolu = new System.Windows.Forms.TextBox();
this.lblDosya = new System.Windows.Forms.Label();
this.grbGrafik = new System.Windows.Forms.GroupBox();
this.dgwNoktalar = new System.Windows.Forms.DataGridView();
this.chkBaslik = new System.Windows.Forms.CheckBox();
((System.ComponentModel.ISupportInitialize)(this.grafik)).BeginInit();
this.grbGirisSeti.SuspendLayout();
this.grbMekanik.SuspendLayout();

```

```

((System.ComponentModel.ISupportInitialize)(this.pbModel)).BeginInit();
this.grbFiziksel.SuspendLayout();
this.grbCikisSeti.SuspendLayout();
this.grbMesajlar.SuspendLayout();
this.grbKontrol.SuspendLayout();
this.grbGrafik.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.dgwNoktalar)).BeginInit();
this.SuspendLayout();
// grafik
this.grafik.BorderColor = System.Drawing.Color.Black;
this.grafik.BorderlineDashStyle =
System.Windows.Forms.DataVisualization.Charting.ChartDashStyle.Solid;
chartArea1.AxisX.Title = "Sehim (mm)";
chartArea1.AxisY.Title = "Yük (N)";
chartArea1.Name = "ChartArea1";
this.grafik.ChartAreas.Add(chartArea1);
this.grafik.Location = new System.Drawing.Point(181, 15);
this.grafik.Name = "grafik";
series1.ChartArea = "ChartArea1";
series1.ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
series1.Legend = "Legend1";
series1.Name = "Series1";
this.grafik.Series.Add(series1);
this.grafik.Size = new System.Drawing.Size(507, 307);
this.grafik.TabIndex = 0;
this.grafik.Text = "chart1";
// txtDurum
this.txtDurum.Dock = System.Windows.Forms.DockStyle.Fill;
this.txtDurum.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(162)));
this.txtDurum.Location = new System.Drawing.Point(3, 16);
this.txtDurum.Multiline = true;
this.txtDurum.Name = "txtDurum";
this.txtDurum.ReadOnly = true;
this.txtDurum.ScrollBars = System.Windows.Forms.ScrollBars.Both;
this.txtDurum.Size = new System.Drawing.Size(297, 79);
this.txtDurum.TabIndex = 4;
// grbGirisSeti
this.grbGirisSeti.Controls.Add(this.grbMekanik);
this.grbGirisSeti.Controls.Add(this.pbModel);
this.grbGirisSeti.Controls.Add(this.grbFiziksel);
this.grbGirisSeti.Location = new System.Drawing.Point(5, 5);
this.grbGirisSeti.Name = "grbGirisSeti";
this.grbGirisSeti.Size = new System.Drawing.Size(573, 331);
this.grbGirisSeti.TabIndex = 5;
this.grbGirisSeti.TabStop = false;
this.grbGirisSeti.Text = "Giriş Parametreleri";
// grbMekanik
this.grbMekanik.Controls.Add(this.textBox7);
this.grbMekanik.Controls.Add(this.textBox8);
this.grbMekanik.Controls.Add(this.textBox9);
this.grbMekanik.Controls.Add(this.lblE);
this.grbMekanik.Controls.Add(this.lblft);
this.grbMekanik.Controls.Add(this.txtd0);
this.grbMekanik.Controls.Add(this.lblD0);
this.grbMekanik.Controls.Add(this.txte);
this.grbMekanik.Controls.Add(this.txtft);
this.grbMekanik.Location = new System.Drawing.Point(319, 202);
this.grbMekanik.Name = "grbMekanik";
this.grbMekanik.Size = new System.Drawing.Size(245, 100);
this.grbMekanik.TabIndex = 31;
this.grbMekanik.TabStop = false;
this.grbMekanik.Text = "Mekanik Özellikler";
// textBox7
this.textBox7.Enabled = false;
this.textBox7.Location = new System.Drawing.Point(199, 16);

```

```

this.textBox7.Name = "textBox7";
this.textBox7.Size = new System.Drawing.Size(40, 20);
this.textBox7.TabIndex = 32;
this.textBox7.Text = "MPa";
// textBox8
this.textBox8.Enabled = false;
this.textBox8.Location = new System.Drawing.Point(199, 67);
this.textBox8.Name = "textBox8";
this.textBox8.Size = new System.Drawing.Size(40, 20);
this.textBox8.TabIndex = 31;
this.textBox8.Text = "mm";
// textBox9
this.textBox9.Enabled = false;
this.textBox9.Location = new System.Drawing.Point(199, 41);
this.textBox9.Name = "textBox9";
this.textBox9.Size = new System.Drawing.Size(40, 20);
this.textBox9.TabIndex = 30;
this.textBox9.Text = "MPa";
// lblE
this.lblE.AutoSize = true;
this.lblE.Location = new System.Drawing.Point(8, 19);
this.lblE.Name = "lblE";
this.lblE.Size = new System.Drawing.Size(102, 13);
this.lblE.TabIndex = 19;
this.lblE.Text = "Elastisite Modülü (E)";
// lblft
this.lblft.AutoSize = true;
this.lblft.Location = new System.Drawing.Point(8, 45);
this.lblft.Name = "lblft";
this.lblft.Size = new System.Drawing.Size(101, 13);
this.lblft.TabIndex = 13;
this.lblft.Text = "Çekme Dayanımı (ft)";
// txtD0
this.txtD0.Location = new System.Drawing.Point(128, 68);
this.txtD0.Name = "txtD0";
this.txtD0.Size = new System.Drawing.Size(65, 20);
this.txtD0.TabIndex = 29;
// lblD0
this.lblD0.AutoSize = true;
this.lblD0.Location = new System.Drawing.Point(8, 71);
this.lblD0.Name = "lblD0";
this.lblD0.Size = new System.Drawing.Size(97, 13);
this.lblD0.TabIndex = 17;
this.lblD0.Text = "Referans Sehim (d)";
// txtE
this.txtE.Location = new System.Drawing.Point(128, 16);
this.txtE.Name = "txtE";
this.txtE.Size = new System.Drawing.Size(65, 20);
this.txtE.TabIndex = 26;
// txtft
this.txtft.Location = new System.Drawing.Point(128, 42);
this.txtft.Name = "txtft";
this.txtft.Size = new System.Drawing.Size(65, 20);
this.txtft.TabIndex = 27;
// pbModel
this.pbModel.Image = global::Kirilis.Properties.Resources.SingleEdgeNotchBending_edited2;
this.pbModel.Location = new System.Drawing.Point(6, 15);
this.pbModel.Name = "pbModel";
this.pbModel.Size = new System.Drawing.Size(307, 307);
this.pbModel.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
this.pbModel.TabIndex = 9;
this.pbModel.TabStop = false;
// grbFiziksel
this.grbFiziksel.Controls.Add(this.textBox6);
this.grbFiziksel.Controls.Add(this.textBox5);
this.grbFiziksel.Controls.Add(this.textBox4);

```

```

this.grbFiziksel.Controls.Add(this.textBox3);
this.grbFiziksel.Controls.Add(this.textBox2);
this.grbFiziksel.Controls.Add(this.textBox1);
this.grbFiziksel.Controls.Add(this.lblB);
this.grbFiziksel.Controls.Add(this.lblS);
this.grbFiziksel.Controls.Add(this.lblYog);
this.grbFiziksel.Controls.Add(this.lblD);
this.grbFiziksel.Controls.Add(this.lbla);
this.grbFiziksel.Controls.Add(this.txtYog);
this.grbFiziksel.Controls.Add(this.lblL);
this.grbFiziksel.Controls.Add(this.txtS);
this.grbFiziksel.Controls.Add(this.txtB);
this.grbFiziksel.Controls.Add(this.txtL);
this.grbFiziksel.Controls.Add(this.txtD);
this.grbFiziksel.Controls.Add(this.txta);
this.grbFiziksel.Location = new System.Drawing.Point(319, 15);
this.grbFiziksel.Name = "grbFiziksel";
this.grbFiziksel.Size = new System.Drawing.Size(245, 181);
this.grbFiziksel.TabIndex = 30;
this.grbFiziksel.TabStop = false;
this.grbFiziksel.Text = "Fiziksel Özellikler";
// textBox6
this.textBox6.Enabled = false;
this.textBox6.Location = new System.Drawing.Point(199, 19);
this.textBox6.Name = "textBox6";
this.textBox6.Size = new System.Drawing.Size(40, 20);
this.textBox6.TabIndex = 28;
this.textBox6.Text = "mm";
// textBox5
this.textBox5.Enabled = false;
this.textBox5.Location = new System.Drawing.Point(199, 146);
this.textBox5.Name = "textBox5";
this.textBox5.Size = new System.Drawing.Size(40, 20);
this.textBox5.TabIndex = 31;
this.textBox5.Text = "kg/m3";
// textBox4
this.textBox4.Enabled = false;
this.textBox4.Location = new System.Drawing.Point(199, 120);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(40, 20);
this.textBox4.TabIndex = 30;
this.textBox4.Text = "mm";
// textBox3
this.textBox3.Enabled = false;
this.textBox3.Location = new System.Drawing.Point(199, 96);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(40, 20);
this.textBox3.TabIndex = 29;
this.textBox3.Text = "mm";
// textBox2
this.textBox2.Enabled = false;
this.textBox2.Location = new System.Drawing.Point(199, 70);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(40, 20);
this.textBox2.TabIndex = 28;
this.textBox2.Text = "mm";
// textBox1
this.textBox1.Enabled = false;
this.textBox1.Location = new System.Drawing.Point(199, 44);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(40, 20);
this.textBox1.TabIndex = 27;
this.textBox1.Text = "mm";
// lblB
this.lblB.AutoSize = true;
this.lblB.Location = new System.Drawing.Point(8, 22);

```

```

this.lblB.Name = "lblB";
this.lblB.Size = new System.Drawing.Size(84, 13);
this.lblB.TabIndex = 1;
this.lblB.Text = "Kiriş Geniřlięi (B)";
// lblS
this.lblS.AutoSize = true;
this.lblS.Location = new System.Drawing.Point(8, 123);
this.lblS.Name = "lblS";
this.lblS.Size = new System.Drawing.Size(94, 13);
this.lblS.TabIndex = 3;
this.lblS.Text = "Mesnet Açıklığı (S)";
// lblYog
this.lblYog.AutoSize = true;
this.lblYog.Location = new System.Drawing.Point(8, 149);
this.lblYog.Name = "lblYog";
this.lblYog.Size = new System.Drawing.Size(104, 13);
this.lblYog.TabIndex = 5;
this.lblYog.Text = "Beton Yoęunluęu (d)";
// lblD
this.lblD.AutoSize = true;
this.lblD.Location = new System.Drawing.Point(8, 47);
this.lblD.Name = "lblD";
this.lblD.Size = new System.Drawing.Size(94, 13);
this.lblD.TabIndex = 7;
this.lblD.Text = "Kiriş Yükseklięi (D)";
// lbla
this.lbla.AutoSize = true;
this.lbla.Location = new System.Drawing.Point(8, 73);
this.lbla.Name = "lbla";
this.lbla.Size = new System.Drawing.Size(92, 13);
this.lbla.TabIndex = 9;
this.lbla.Text = "Çentik Derinlięi (a)";
// txtYog
this.txtYog.Location = new System.Drawing.Point(128, 146);
this.txtYog.Name = "txtYog";
this.txtYog.Size = new System.Drawing.Size(65, 20);
this.txtYog.TabIndex = 25;
// lblL
this.lblL.AutoSize = true;
this.lblL.Location = new System.Drawing.Point(8, 99);
this.lblL.Name = "lblL";
this.lblL.Size = new System.Drawing.Size(110, 13);
this.lblL.TabIndex = 11;
// txtS
this.txtS.Location = new System.Drawing.Point(128, 120);
this.txtS.Name = "txtS";
this.txtS.Size = new System.Drawing.Size(65, 20);
this.txtS.TabIndex = 24;
// txtB
this.txtB.Location = new System.Drawing.Point(128, 19);
this.txtB.Name = "txtB";
this.txtB.Size = new System.Drawing.Size(65, 20);
this.txtB.TabIndex = 10;
// txtL
this.txtL.Location = new System.Drawing.Point(128, 96);
this.txtL.Name = "txtL";
this.txtL.Size = new System.Drawing.Size(65, 20);
this.txtL.TabIndex = 23;
// txtD
this.txtD.Location = new System.Drawing.Point(128, 44);
this.txtD.Name = "txtD";
this.txtD.Size = new System.Drawing.Size(65, 20);
// txta
this.txta.Location = new System.Drawing.Point(128, 70);
this.txta.Name = "txta";
this.txta.Size = new System.Drawing.Size(65, 20);

```

```

this.txta.TabIndex = 22;
// grbCikisSeti
this.grbCikisSeti.Controls.Add(this.textBox13);
this.grbCikisSeti.Controls.Add(this.textBox15);
this.grbCikisSeti.Controls.Add(this.textBox10);
this.grbCikisSeti.Controls.Add(this.textBox11);
this.grbCikisSeti.Controls.Add(this.textBox12);
this.grbCikisSeti.Controls.Add(this.txtlch);
this.grbCikisSeti.Controls.Add(this.txtGf);
this.grbCikisSeti.Controls.Add(this.txtW0);
this.grbCikisSeti.Controls.Add(this.txtfnet);
this.grbCikisSeti.Controls.Add(this.txtFL);
this.grbCikisSeti.Controls.Add(this.lblGf);
this.grbCikisSeti.Controls.Add(this.lblch);
this.grbCikisSeti.Controls.Add(this.lblW0);
this.grbCikisSeti.Controls.Add(this.lblfnet);
this.grbCikisSeti.Controls.Add(this.lblFL);
this.grbCikisSeti.Location = new System.Drawing.Point(5, 344);
this.grbCikisSeti.Name = "grbCikisSeti";
this.grbCikisSeti.Size = new System.Drawing.Size(472, 99);
this.grbCikisSeti.TabIndex = 6;
this.grbCikisSeti.TabStop = false;
this.grbCikisSeti.Text = "Sonuç Değerleri";
this.grbCikisSeti.Enter += new System.EventHandler(this.grbCikisSeti_Enter);
// textBox13
this.textBox13.Enabled = false;
this.textBox13.Location = new System.Drawing.Point(427, 15);
this.textBox13.Name = "textBox13";
this.textBox13.Size = new System.Drawing.Size(35, 20);
this.textBox13.TabIndex = 38;
this.textBox13.Text = "N/mm";
// textBox15
this.textBox15.Enabled = false;
this.textBox15.Location = new System.Drawing.Point(427, 40);
this.textBox15.Name = "textBox15";
this.textBox15.Size = new System.Drawing.Size(35, 20);
this.textBox15.TabIndex = 36;
this.textBox15.Text = "mm";
// textBox10
this.textBox10.Enabled = false;
this.textBox10.Location = new System.Drawing.Point(197, 15);
this.textBox10.Name = "textBox10";
this.textBox10.Size = new System.Drawing.Size(35, 20);
this.textBox10.TabIndex = 35;
this.textBox10.Text = "N";
// textBox11
this.textBox11.Enabled = false;
this.textBox11.Location = new System.Drawing.Point(197, 66);
this.textBox11.Name = "textBox11";
this.textBox11.Size = new System.Drawing.Size(35, 20);
this.textBox11.TabIndex = 34;
this.textBox11.Text = "Nmm";
// textBox12
this.textBox12.Enabled = false;
this.textBox12.Location = new System.Drawing.Point(197, 40);
this.textBox12.Name = "textBox12";
this.textBox12.Size = new System.Drawing.Size(35, 20);
this.textBox12.TabIndex = 33;
this.textBox12.Text = "MPa";
// txtlch
this.txtlch.Enabled = false;
this.txtlch.Location = new System.Drawing.Point(356, 39);
this.txtlch.Name = "txtlch";
this.txtlch.Size = new System.Drawing.Size(65, 20);
this.txtlch.TabIndex = 31;
// txtGf

```

```

this.txtGf.Enabled = false;
this.txtGf.Location = new System.Drawing.Point(356, 15);
this.txtGf.Name = "txtGf";
this.txtGf.Size = new System.Drawing.Size(65, 20);
this.txtGf.TabIndex = 30;
// txtW0
this.txtW0.Enabled = false;
this.txtW0.Location = new System.Drawing.Point(126, 67);
this.txtW0.Name = "txtW0";
this.txtW0.Size = new System.Drawing.Size(65, 20);
this.txtW0.TabIndex = 28;
// txtfnet
this.txtfnet.Enabled = false;
this.txtfnet.Location = new System.Drawing.Point(126, 41);
this.txtfnet.Name = "txtfnet";
this.txtfnet.Size = new System.Drawing.Size(65, 20);
this.txtfnet.TabIndex = 27;
this.txtfnet.TextChanged += new System.EventHandler(this.txtfnet_TextChanged);
// txtFL
this.txtFL.Enabled = false;
this.txtFL.Location = new System.Drawing.Point(126, 15);
this.txtFL.Name = "txtFL";
this.txtFL.Size = new System.Drawing.Size(65, 20);
this.txtFL.TabIndex = 26;
// lblGf
this.lblGf.AutoSize = true;
this.lblGf.Location = new System.Drawing.Point(236, 18);
this.lblGf.Name = "lblGf";
this.lblGf.Size = new System.Drawing.Size(93, 13);
this.lblGf.TabIndex = 25;
this.lblGf.Text = "Kırılma Enerjisi (Gf)";
// lblch
this.lblch.AutoSize = true;
this.lblch.Location = new System.Drawing.Point(236, 42);
this.lblch.Name = "lblch";
this.lblch.Size = new System.Drawing.Size(112, 13);
this.lblch.TabIndex = 23;
this.lblch.Text = "Karakteristik Boy ( lch)";
// lblW0
this.lblW0.AutoSize = true;
this.lblW0.Location = new System.Drawing.Point(6, 70);
this.lblW0.Name = "lblW0";
this.lblW0.Size = new System.Drawing.Size(66, 13);
this.lblW0.TabIndex = 19;
this.lblW0.Text = "Tokluk (W0)";
// lblfnet
this.lblfnet.AutoSize = true;
this.lblfnet.Location = new System.Drawing.Point(6, 44);
this.lblfnet.Name = "lblfnet";
this.lblfnet.Size = new System.Drawing.Size(95, 13);
this.lblfnet.TabIndex = 17;
this.lblfnet.Text = "Net Eğ. Day. (fnet)";
// lblFL
// this.lblFL.AutoSize = true;
this.lblFL.Location = new System.Drawing.Point(6, 18);
this.lblFL.Name = "lblFL";
this.lblFL.Size = new System.Drawing.Size(25, 13);
this.lblFL.TabIndex = 13;
this.lblFL.Text = "(FL)";
// grbMesajlar
this.grbMesajlar.Controls.Add(this.txtDurum);
this.grbMesajlar.Location = new System.Drawing.Point(483, 345);
this.grbMesajlar.Name = "grbMesajlar";
this.grbMesajlar.Size = new System.Drawing.Size(303, 98);
this.grbMesajlar.TabIndex = 7;
this.grbMesajlar.TabStop = false;

```

```

this.grbMesajlar.Text = "Mesajlar";
// grbKontrol
this.grbKontrol.Controls.Add(this.chkBaslik);
this.grbKontrol.Controls.Add(this.lblDeneyTipi);
this.grbKontrol.Controls.Add(this.chkSehim);
this.grbKontrol.Controls.Add(this.chkCMOD);
this.grbKontrol.Controls.Add(this.btnTemizle);
this.grbKontrol.Controls.Add(this.btnHesapla);
this.grbKontrol.Controls.Add(this.btnYukle);
this.grbKontrol.Controls.Add(this.btnGozat);
this.grbKontrol.Controls.Add(this.txtDosyaYolu);
this.grbKontrol.Controls.Add(this.lblDosya);
this.grbKontrol.Location = new System.Drawing.Point(795, 345);
this.grbKontrol.Name = "grbKontrol";
this.grbKontrol.Size = new System.Drawing.Size(488, 98);
this.grbKontrol.TabIndex = 8;
this.grbKontrol.TabStop = false;
this.grbKontrol.Text = "Kontrol";
// lblDeneyTipi
this.lblDeneyTipi.AutoSize = true;
this.lblDeneyTipi.Location = new System.Drawing.Point(6, 65);
this.lblDeneyTipi.Name = "lblDeneyTipi";
this.lblDeneyTipi.Size = new System.Drawing.Size(58, 13);
this.lblDeneyTipi.TabIndex = 8;
this.lblDeneyTipi.Text = "Deney Tipi";
// chkSehim
this.chkSehim.AutoSize = true;
this.chkSehim.Location = new System.Drawing.Point(68, 63);
this.chkSehim.Name = "chkSehim";
this.chkSehim.Size = new System.Drawing.Size(82, 17);
this.chkSehim.TabIndex = 7;
this.chkSehim.Text = "Yük - Sehim";
this.chkSehim.UseVisualStyleBackColor = true;
this.chkSehim.CheckedChanged += new System.EventHandler(this.chkSehim_CheckedChanged);
// chkCMOD
this.chkCMOD.AutoSize = true;
this.chkCMOD.Location = new System.Drawing.Point(156, 63);
this.chkCMOD.Name = "chkCMOD";
this.chkCMOD.Size = new System.Drawing.Size(85, 17);
this.chkCMOD.TabIndex = 6;
this.chkCMOD.Text = "Yük - CMOD";
this.chkCMOD.
// btnTemizle
this.btnTemizle.Location = new System.Drawing.Point(372, 59);
this.btnTemizle.Name = "btnTemizle";
this.btnTemizle.Size = new System.Drawing.Size(110, 25);
this.btnTemizle.TabIndex = 5;
this.btnTemizle.Text = "Temizle";
this.btnTemizle.UseVisualStyleBackColor = true;
this.btnTemizle.Click += new System.EventHandler(this.btnTemizle_Click);
// btnHesapla
this.btnHesapla.Location = new System.Drawing.Point(256, 59);
this.btnHesapla.Name = "btnHesapla";
this.btnHesapla.Size = new System.Drawing.Size(110, 25);
this.btnHesapla.TabIndex = 4;
this.btnHesapla.Text = "Hesapla";
this.btnHesapla.UseVisualStyleBackColor = true;
this.btnHesapla.Click += new System.EventHandler(this.btnHesapla_Click);
// btnYukle
this.btnYukle.Location = new System.Drawing.Point(421, 15);
this.btnYukle.Name = "btnYukle";
this.btnYukle.Size = new System.Drawing.Size(61, 21);
this.btnYukle.TabIndex = 3;
this.btnYukle.Text = "Yükle";
this.btnYukle.UseVisualStyleBackColor = true;
this.btnYukle.Click += new System.EventHandler(this.btnYukle_Click);

```

```

// btnGozat
this.btnGozat.Location = new System.Drawing.Point(354, 15);
this.btnGozat.Name = "btnGozat";
this.btnGozat.Size = new System.Drawing.Size(61, 21);
this.btnGozat.TabIndex = 2;
this.btnGozat.Text = "Gözet";
this.btnGozat.UseVisualStyleBackColor = true;
this.btnGozat.Click += new System.EventHandler(this.btnGozat_Click);
// txtDosyaYolu
this.txtDosyaYolu.Location = new System.Drawing.Point(49, 16);
this.txtDosyaYolu.Name = "txtDosyaYolu";
this.txtDosyaYolu.Size = new System.Drawing.Size(299, 20);
this.txtDosyaYolu.TabIndex = 1;
// lblDosya
this.lblDosya.AutoSize = true;
this.lblDosya.Location = new System.Drawing.Point(6, 19);
this.lblDosya.Name = "lblDosya";
this.lblDosya.Size = new System.Drawing.Size(37, 13);
this.lblDosya.TabIndex = 0;
this.lblDosya.Text = "Dosya";
// grbGrafik
this.grbGrafik.Controls.Add(this.dgwNoktalar);
this.grbGrafik.Controls.Add(this.grafik);
this.grbGrafik.Location = new System.Drawing.Point(589, 5);
this.grbGrafik.Name = "grbGrafik";
this.grbGrafik.Size = new System.Drawing.Size(694, 331);
this.grbGrafik.TabIndex = 9;
this.grbGrafik.TabStop = false;
// dgwNoktalar
this.dgwNoktalar.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.ColumnHeader;
this.dgwNoktalar.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgwNoktalar.Location = new System.Drawing.Point(6, 15);
this.dgwNoktalar.Name = "dgwNoktalar";
this.dgwNoktalar.RowHeadersWidth = 10;
this.dgwNoktalar.Size = new System.Drawing.Size(169, 307);
this.dgwNoktalar.TabIndex = 22;
// chkBaslik
this.chkBaslik.AutoSize = true;
this.chkBaslik.Location = new System.Drawing.Point(49, 40);
this.chkBaslik.Name = "chkBaslik";
this.chkBaslik.Size = new System.Drawing.Size(146, 17);
this.chkBaslik.TabIndex = 10;
this.chkBaslik.Text = "Dosya başlık satırı içeriyor";
this.chkBaslik.UseVisualStyleBackColor = true;
// AnaPencere
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleModeMode.Font;
this.ClientSize = new System.Drawing.Size(1295, 451);
this.Controls.Add(this.grbGrafik);
this.Controls.Add(this.grbCikisSeti);
this.Controls.Add(this.grbKontrol);
this.Controls.Add(this.grbMesajlar);
this.Controls.Add(this.grbGirisSeti);
this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.Name = "AnaPencere";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Kirilis 1.5.2";
this.Load += new System.EventHandler(this.AnaPencere_Load);
((System.ComponentModel.ISupportInitialize)(this.grafik)).EndInit();
this.grbGirisSeti.ResumeLayout(false);
this.grbMekanik.ResumeLayout(false);
this.grbMekanik.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.pbModel)).EndInit();
this.grbFiziksel.ResumeLayout(false);

```

```

        this.grbFiziksel.PerformLayout();
        this.grbCikisSeti.ResumeLayout(false);
        this.grbCikisSeti.PerformLayout();
        this.grbMesajlar.ResumeLayout(false);
        this.grbMesajlar.PerformLayout();
        this.grbKontrol.ResumeLayout(false);
        this.grbKontrol.PerformLayout();
        this.grbGrafik.ResumeLayout(false);
        ((System.ComponentModel.ISupportInitialize)(this.dgwNoktalar)).EndInit();
        this.ResumeLayout(false);
    }
    #endregion
    private System.Windows.Forms.DataVisualization.Charting.Chart grafik;
    private System.Windows.Forms.TextBox txtDurum;
    private System.Windows.Forms.GroupBox grbGirisSeti;
    private System.Windows.Forms.Label lblB;
    private System.Windows.Forms.Label lblE;
    private System.Windows.Forms.Label lblD0;
    private System.Windows.Forms.Label lblft;
    private System.Windows.Forms.Label lblL;
    private System.Windows.Forms.Label lbla;
    private System.Windows.Forms.Label lblD;
    private System.Windows.Forms.Label lblYog;
    private System.Windows.Forms.Label lblS;
    private System.Windows.Forms.GroupBox grbCikisSeti;
    private System.Windows.Forms.Label lblGf;
    private System.Windows.Forms.Label lblch;
    private System.Windows.Forms.Label lblW0;
    private System.Windows.Forms.Label lblfnet;
    private System.Windows.Forms.Label lblFL;
    private System.Windows.Forms.GroupBox grbMesajlar;
    private System.Windows.Forms.GroupBox grbKontrol;
    private System.Windows.Forms.Button btnYukle;
    private System.Windows.Forms.Button btnGozat;
    private System.Windows.Forms.TextBox txtDosyaYolu;
    private System.Windows.Forms.Label lblDosya;
    private System.Windows.Forms.Label lblDeneyTipi;
    private System.Windows.Forms.RadioButton chkSehim;
    private System.Windows.Forms.RadioButton chkCMOD;
    private System.Windows.Forms.Button btnTemizle;
    private System.Windows.Forms.Button btnHesapla;
    private System.Windows.Forms.PictureBox pbModel;
    private System.Windows.Forms.TextBox txtB;
    private System.Windows.Forms.TextBox txtlch;
    private System.Windows.Forms.TextBox txtGf;
    private System.Windows.Forms.TextBox txtW0;
    private System.Windows.Forms.TextBox txtfnet;
    private System.Windows.Forms.TextBox txtFL;
    private System.Windows.Forms.TextBox txtD0;
    private System.Windows.Forms.TextBox txtft;
    private System.Windows.Forms.TextBox txtE;
    private System.Windows.Forms.TextBox txtYog;
    private System.Windows.Forms.TextBox txtS;
    private System.Windows.Forms.TextBox txtL;
    private System.Windows.Forms.TextBox txta;
    private System.Windows.Forms.TextBox txtD;
    private System.Windows.Forms.GroupBox grbFiziksel;
    private System.Windows.Forms.GroupBox grbGrafik;
    private System.Windows.Forms.GroupBox grbMekanik;
    private System.Windows.Forms.DataGridView dgwNoktalar;
    private System.Windows.Forms.TextBox textBox7;
    private System.Windows.Forms.TextBox textBox8;
    private System.Windows.Forms.TextBox textBox9;
    private System.Windows.Forms.TextBox textBox6;
    private System.Windows.Forms.TextBox textBox5;
    private System.Windows.Forms.TextBox textBox4;

```

```
private System.Windows.Forms.TextBox textBox3;  
private System.Windows.Forms.TextBox textBox2;  
private System.Windows.Forms.TextBox textBox1;  
private System.Windows.Forms.TextBox textBox13;  
private System.Windows.Forms.TextBox textBox15;  
private System.Windows.Forms.TextBox textBox10;  
private System.Windows.Forms.TextBox textBox11;  
private System.Windows.Forms.TextBox textBox12;  
private System.Windows.Forms.CheckBox chkBaslik;  
}  
}
```

ÖZGEÇMİŞ

Adı Soyadı : Nurgün GENÇEL

Doğum Yeri ve Yılı : Isparta, 1985

Medeni Hali : Evli

Yabancı Dili : İngilizce

E-posta : eposta@sdu.edu.tr

Eğitim Durumu

Lise : Ankara Üreğil Anadolu Kız Meslek Lisesi, 2002

Lisans : Gazi, Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Öğretmenliği

Mesleki Deneyim

Isparta Şehit Erhan Çakmak İlköğretim Okulu Bilgisayar Öğretmenliği 2008-2010

Bartın Kemal Sabriye Ocakçı İlköğretim Okulu Bilgisayar Öğretmenliği 2010-2011

Bartın Ahmetler Ortaokulu 2011-..... (halen)