



DERS ADI Görsel Programlama I
ÜNİTE ADI Görsel Programlamaya Giriş
ÜNİTE NO 1
YAZAR Öğr. Gör. GÖKHAN TUTAR

GÖRSEL PROGRAMLAMA NEDİR?

Görsel programlama genel itibarıyla grafiksel kullanıcı arayüzlü bir yapıya sahiptir. İngilizcede “Graphical User Interface” olarak geçmekte ve kısaltması GUI olarak yazılmaktadır. GUI yani grafiksel kullanıcı arayüzü, kullanıcının bilgisayar ekranındaki küçük resimlerle bir işaretleme aygıtı (fare veya kalem gibi) kullanarak etkileşime girmesine olanak tanımaktadır. Tanımda belirtilen küçük resimlere “simge” veya “widget” denmektedir. Dolayısıyla grafik arayüzü kullanan programlara da görsel programlar denmektedir. Görsel programlar komut satırından kod girilen console programlarından oldukça farklıdır. Console programları daha önceden geliştirilen kodları sırasıyla gerçekleştirerek programı sonlandırırlar. Ancak görsel programlarda kullanıcılar simgeler kullanılarak oluşturulan bir arayüz üzerinde çalışırlar. Arayüzde bulunan simgelere tıklayarak kullanıcılar yapmak istedikleri işlemleri gerçekleştirirler. Bir başka ifadeyle programdaki kodların çalıştırılma sırası yukardan aşağı değil kullanıcının bu simgeler ile etkileşimine göre yapılmaktadır.

Görsel Programlamanın özellikleri

Görsel programlamanın en dikkat çeken özelliği olaylardır. GUI’de yazılımın sürekli olarak kullanıcıyı takip etmesi ve kullanıcının yaptığı işleme göre gerekli kodları çalıştırması gerekir. Kullanıcının yaptığı bu işlemin adına “olay” denir. Kullanıcının ekrandaki bir butona tıklaması örnek bir “olay” dır. Bu programlama tarzına ise olay güdümlü programlama denmektedir.

Görsel Programlamanın Avantaj ve Dezavantajları

Grafiksel arayüzün en büyük avantajı, son kullanıcılar (end user) için yazılımların kullanımını oldukça kolaylaştırmasıdır. Örneğin dosyaları aktarmak için sürükleyip bırak gibi bilindik olaylardan yararlanır. Silinen dosyalar için çöp kutusu gibi günlük hayattan tanıdık simgeleri kullanarak ve bilgisayar işlemlerini sezgisel hale getirerek herhangi bir işlem yapmadan kolayca yönetildiği bir ortam oluşturur. Kısacası grafiksel arayüzler sayesinde uygulamalar kendi kendini açıklayıcı hale gelirler. Geri bildirim genellikle anındadır ve görsel ipuçları sayesinde kolay keşfedilebilir.

GÖRSEL PROGRAMLAMANIN TARİHSEL GELİŞİMİ

Görsel programlama 1964 yılında Douglas Engelbart tarafından farenin icadı ile başlamıştır. 1960’lı yıllarda bilgisayarlar kişisel kullanımdan çok hükümetler tarafından yüksek maliyetli projelerde kullanılmaktayken, 1980’li yılların ortalarına gelindiğinde bilgisayarların küçülmesi ve maliyetlerin azalması ile birlikte bu durum yavaş yavaş değişmeye başlamıştır. 1984 yılında Apple, kişisel bilgisayarların kullanımı arttırmak için Macintosh adını verdiği ürünü piyasaya sürmüştü ve Macintosh’lar günümüz kişisel bilgisayarlarında kullanılan birçok özelliğin öncüsü olmuştur. Öte yandan görsel programlamanın gelişimini tek bir kişi veya firma ile ilişkilendirmek çok doğru bir yaklaşım değildir. Grafiksel arayüzlerin gelişiminde birçok kişi ve firma farklı zamanlara değerli katkılarda bulunarak gelecekteki gelişmelere ön ayak olmuştur. Görsel programlamaya değerli katkılar sunan ürünler ve geliştirmeler şu şekildedir:

Xerox PARC

Kişisel bilgisayarların kullanımının artması bir fotokopi makinesi üreticisi olan Xerox firmasını rahatsız etmiştir. Xerox, birçok işlemin dijital olarak gerçekleştirilmesinin kâğıda ve fotokopiye olan ihtiyacı azaltacağını ön görmüştür. Teknoloji dünyasında yaşanacak büyük dönüşümün farkına varan Xerox firması yeniliklere ayak uydurabilmek ve rekabet avantajını kaybetmemek amacıyla 1970 yılında Palo Alto Araştırma Merkezi (Palo Alto Research Center) diğer adıyla PARC’ı kurmuştur. Xerox, burada insanlara gerekli imkânları sunarak araştırma yapmalarına olanak sağlamıştır. Yapılan yatırımlar kısa sürede meyvesini vermiş ve kullanımı günümüze uzanan birçok icat ortaya çıkmıştır. Ortaya çıkan bu icatlardan birisi de lazer yazıcıdır. Lazer yazıcının icadı ile kullanıcılar bir grafik ekrana ve bu ekranı yönetebilecek yazılıma ihtiyaç duymuşlardır. Bu ihtiyaçlar doğrultusunda ise Xerox Alto cihazı üretilmiştir. Xerox Alto hem bilgisayar hem de işletim sistemi barındırmaktaydı. Bu aygıt, bir sayfayı düzenlenip yazıcıya göndererek işlemler yapabilmekteydi.

Tasarımı modern kişisel bilgisayarlara benzemesine karşın Xerox Alto ticari başarı yakalayamamıştır. Üretilen bilgisayarların bir kısmı Xerox firmasında kullanılmış bir kısmı da üniversitelere dağıtılmıştır. Alto için geliştirilen yazılım günümüzdeki yazılımlar kadar görsel özelliklere sahip olmamasına rağmen grafiksel bir ekrana sahipti. Bunun yanında Alto'da farklı işlemleri aynı anda yapılmasını sağlayan pencereler bulunmamaktaydı. Her işlemin tek tek yapılması gerekmektedir. Öte yandan Xerox'un Alto projesinde çalışan mühendislerin bir kısmı, ilerleyen zamanda Microsoft veya Apple gibi firmalara geçiş yaparak Xerox PARC projesinde edindikleri tecrübeleri sayesinde görsel programlamanın gelişimine katkı sağlamışlardır.

Smalltalk

Smalltalk, hem bir yazılım dili hem de yazılım geliştirme ortamıdır. Smalltalk, önceden oluşturulan nesnelerin tekrar tekrar kodlanmaya gerek kalmadan kullanıldığı ilk görsel programlama dillerinden birisidir. Smalltalk dilinde bellek yönetimi gibi zor işlemleri otomatik olarak gerçekleştirmektedir. Smalltalk kullanım kolaylıklarından dolayı zamanının popüler programlama dilleri arasındaydı. Smalltalk'ın ilk sürümü 1974 yılında piyasaya sürülmüştür ve sürekli güncellenerek gelişmiştir.

Apple

GUI'nun öncülerinden hatta en önemlilerinden biriside Steve Jobs ve Steve Wozniak tarafından 1976 yılında Apple ismiyle kurulan şirkettir. Apple firmasının ilk bilgisayarı, hem metin hem de grafik görüntüleyen ancak geleneksel komut satırı arayüzüne sahipti. Xerox PARC'dan ayrılan mühendislerin Apple'da kendilerine iş bulmasıyla bu bilgisayar, grafiksel kullanıcı arayüzü (GUI) dönüştürülmüş ve bu bilgisayara "Lisa" adı verilmiştir.

Lisa bilgisayarının geliştirilmesine 1978 yılında başlanmasına rağmen ancak 1983 yılında piyasaya sürülebilmektedir. Lisa bilgisayarı dönemine göre oldukça gelişmiş özelliklere sahip olmasına rağmen fiyatının pahalı olmasından dolayı satışı sınırlı sayıda kalmıştır. Bunun üzerine Steve Jobs, daha düşük maliyetli bir bilgisayar olan Macintosh projesini başlatmıştır. Macintosh'ta hem bilgisayarın maliyeti düşürülmüş hem de Lisa'nın grafiksel özelliklerinin çoğu korunmuştur. Macintosh projesi Lisa'ya göre başarılı olmuş ve önemli satış rakamlarına ulaşmıştır.

VisiOn

Apple grafiksel kullanıcı arayüzü alanında çalışırken diğer firmalarda bu alana çoktan yönelmişlerdi. Grafiksel arayüz alanında çalışma yapan firmalardan birisi de VisiCorp'dur. VisiCorp firması IBM'in ürettiği bilgisayarlar için grafik kullanıcı arayüzlü VisiCalc adında grafik arabirimleri üretmekteydi. VisiCorp firması 1983 yılında VisiOn'u piyasaya sürmüştür. VisiOn yazılımı "Word" veya "Excel" gibi birçok programı içerinden barındırmaktaydı. VisiOn programının rakiplerinden farklı hem fare kullanılabilir olması hem de işlemlerin grafik arayüzleri sayesinde yapılmasıydı. Ancak VisiOn'un ömrü kısa sürmüştür. VisiOn'un ömrü kısa olsa da gelecekte dünyadaki en büyük teknolojik firmalarından olacak olan Microsoft'un kurulmasında büyük etkisi vardır.

Windows

VisiOn her ne kadar geniş kullanıcı kitlesine ulaşamasa da Microsoft'un kurucusu Bill Gates'e ilham kaynağı olmuş Windows'un geliştirilme sürecinde önemli bir yer edinmiştir. Windows tıpkı Lisa gibi grafiksel kullanıcı arayüzüne sahipti. Tek bir menü çubuğu yerine her uygulamanın kendine ait menü çubuğu bulunmaktaydı. Pencere diğer bir adıyla form kavramı hayatımıza Windows ile girmiştir.

GÖRSEL PROGRAMLAMA DİLLERİ

Bilgisayarların herkes tarafından kullanılması, programların görsel arayüzlerini daha önemli hale getirmiştir. Çünkü artık bilgisayarlar sadece teknik bilgiye sahip olan kişiler tarafından değil toplumun bütün kesimleri tarafından kullanılmaktadır. Bundan dolayı günümüzde çok sayıda görsel programlama dili ve bu dilleri geliştirebilecek çok daha fazla ortam bulunmaktadır.

C/C++

C++ dili Bjarne Stroustrup tarafından 1979 yılından itibaren geliştirilmeye başlanmıştır. C++ dili aslında C dilinin geliştirilmiş halidir. C dili nesne yönelimli bir dil değildir. C++ dili C dilinin nesne yönelimli şekle dönüştürülmüş halidir. İsmi de bulunan "++" ifadesi programlama dillerinde arttırma operatörü olarak kullanılmaktadır. Bundan dolayı C++'ın C dilinin bir üst versiyonu olduğu anlamı çıkarılabilir.

Delphi

Delphi günümüzde kullanılmaya devam edilen görsel programlama dillerinden birisidir. Köken olarak Pascal diline dayanmaktadır. 1980'lerin sonlarında Borland firması, Mac için Turbo Pascal'ı ve DOS için Turbo Pascal 5.5'i tanıttı. Doksanların ortalarında Borland, DOS'tan Windows'a geçiş yaparken Turbo Pascal yerine Delphi adını kullanmaya başladı.

Java

İnteraktif televizyonlar için yazılım üretmek amacı ile Java ortaya çıkmıştır. Zamanının çok ilerisinde olmasına rağmen etkileşimli televizyon alanında büyük başarı elde edememiştir. Java programlama dilinin temel ilkeleri arasında platformdan bağımsız, basit, performanslı ve nesne tabanlı bir programlama dili olmak vardır. Bu temel ilkeler arasında en öne çıkan platformdan bağımsız yapıya sahip olmasıdır. Java günümüzde de bu ilkeyi devam ettirmektedir. Java'nın elektronik makineleri

programlamadan mobil programlamaya kadar çok sayıda kullanım alanı bulunmaktadır.

Objective-C

Objective-C dilinin ilk sürümü 1980'lerin başında geliştirildi. Bu dil geçmişte NextStep işletim sistemi için ana programlama dili olarak seçilmiştir. Objective-C daha sonra MacOS'de kullanılmıştır. 2014 yılında Apple firması Swift dilini tanıtana kadar kullanılmaya devam etmekteydi.

Visual Basic

Visual Basic olaylara dayalı üçüncü nesil görsel programlama dillerinden bir tanesidir. Visual Basic, Basic dilinin geliştirilmiş halidir. İlk olarak 1991 yılında Microsoft tarafından piyasaya sürüldü. Microsoft, 2008 yılında Visual Basic'in kullanımı kolaylaştırılarak 9.0 versiyonunu piyasaya sürdü. Kullanımı kolay bir dildir. Günümüzde geliştirme ortamı olarak Visual Studio IDE'sini kullanmaktadır.

C#

C# günümüzde en çok kullanılan programlama dilleri arasındadır. Microsoft tarafından 2002 yılında Visual Studio.Net çatısı altında piyasaya sürülmüştür. C# ile görsel, console veya web programlama yapılabilmektedir. IDE olarak Microsoft'un Visual Studio programını kullanmaktadır. Visual Studio çok güçlü bir IDE'dir. Bu IDE sayesinde formlar ve forma elementler (textbox veya button gibi) tut sürükleyip bırak yöntemi ile kolaylıkla oluşturulabilir.

Python

Guido Van Rossum tarafından geliştirilmiştir. Çeşitli programlama stillerini desteklemek ve kullanımı kolaylaştırmak için oluşturulmuş genel amaçlı, üst düzey bir programlama dilidir. Python kendini her ne kadar kullanımı kolay bir dil olarak tanımlasa da bu dilde görsel arayüzler hazırlamak karmaşıktır. Python dili yapay zeka, makine öğrenmesi veya bulanık mantık gibi alanlarda çok sayıda açık kaynak kodlu kütüphaneye sahiptir. Bundan dolayı Python dili bilimsel projelerde sıklıkla kullanılmaktadır.

GÖRSEL PROGRAMLAMA GELİŞTİRME ORTAMLARI

Görsel programlama dillerinde (C#, Delphi veya Python gibi) yazılım geliştirmek için bilgisayarınıza bir geliştirme ortamı yani IDE yüklemeniz gerekmektedir. Geliştirme ortamları, yazılım geliştirme sürecini hızlandırmak ve kolaylaştırmak için bir ihtiyaçtır. 1980'li yıllarda bir yazılım dili için sadece bir geliştirme ortamı veya editörü kullanırdı. Ancak günümüzdeki IDE'ler birden fazla yazılım dilini destekleyebilmektedir. IDE tercihi yazılımcılar için önemlidir çünkü yazılan programın çalıştırılması, hata ayıklama veya kodun otomatik tamamlanması gibi işlemler IDE üzerinden gerçekleştirilmektedir.

NetBeans

NetBeans başta Java olmak üzere C++, PHP gibi farklı yazılım dillerini geliştirmek için kullanılmaktadır. Açık kaynak kodlu ücretsiz bir yazılımdır. Açık kaynak kodlu olduğu için diğer kullanıcılar da Netbeans için eklentiler hazırlayıp kullanıma sunabilmektedir. Kullanımı basit ve kolaydır. NetBeans'de derleme, hata bulma ve otomatik kod tamamlama gibi özellikler bulunmaktadır. Netbeans, Java dili ile geliştirilmiştir. Netbeans Windows, Mac OS veya Linux gibi çok sayıda işletim sisteminde çalışabilmektedir. Netbeans'ın kodları Oracle firması tarafından 2016 yılında Apache Software Foundation'a devredilmiştir.

Xcode

MacOS, iOS ve tvOS gibi Apple ortamlarına yazılım geliştirilmek için kullanılan bir IDE'dir. 2003 yılında Apple tarafından piyasaya sürülmüştür. Objective-C ve Swift gibi Apple firmasının sıklıkla kullandığı dilleri desteklemektedir. Xcode sayesinde tasarım, hata ayıklama ve kod tamamlama gibi özellikler çok kolay ve hızlı bir şekilde gerçekleştirilebilmektedir.

Eclipse

Eclipse tıpkı Netbeans gibi açık kaynak kodlu ve ücretsiz bir yazılım geliştirme platformudur. Eclipse projesi 2001 yılında IBM tarafından başlatılmıştır. Bu IDE başlangıçta Java dilinde programlar geliştirmek için oluşturulmuştur. Daha sonraları PHP, C/C++, Perl veya Python gibi çok kullanılan dilleri ile programlama imkânı sunacak şekilde kapsamı genişletilmiştir. Eclipse kullanılarak mobil programlama bile geliştirilebilmektedir. Tıpkı NetBeans gibi kullanıcıların geliştirdikleri eklentiler Eclipse ile entegre edilebilmektedir. Eclipse ile tasarım, hata ayıklama ve kod tamamlama gibi işlemler gerçekleştirilebilir.

Visual Studio

Visual Studio IDE'si Microsoft firması tarafından geliştirilmiştir. Visual Studio'nun asıl amacı Microsoft yazılım dillerinde (C# veya Visual Basic gibi) geliştirme yapılabilecek ortamlar sağlamaktır. Ancak günümüzde Python veya Ruby gibi farklı dilleri de desteklemektedir. Visual Studio programı ile web, console, görsel ve mobil programlama eş zamanlı olarak yapılabilmektedir. Hata ayıklama, kod tamamlama veya tasarım gibi işlemler hızlı ve kolay şekilde gerçekleştirilebilir. Ücretli ve ücretsiz sürümleri bulunmaktadır. Öğrenciler ve öğretmenler yani ticari amaç gütmeyenler ücretsiz sürümünü kullanabilmektedir.



DERS ADI Görsel Programlama I
ÜNİTE ADI C# ile Görsel Programlama
ÜNİTE NO 2
YAZAR Öğr. Gör. GÖKHAN TUTAR

VISUAL STUDIO KURULUMU

Visual Studio programının birçok sürümü bulunmaktadır. Günümüzdeki son sürümü 2019'dur ve 2022 yılında yeni sürüm çıkarılması planlanmaktadır. Visual Studio eski sürümleri sadece Windows işletim sistemi üzerinde çalışabilmekteydi ancak günümüzde MacOS işletim sisteminde çalışan Visual Studio sürümleri de bulunmaktadır.

Visual Studio bireysel kullanımlarda ücretsizdir ancak ticari amaçlar için kullanımı ücretlidir. Visual Studio'nun ücretsiz sürümünde bazı özellikler kapalı olsa da bireysel ve eğitim amaçlı kullanım için idealdir. Visual Studio IDE'sini ücretsiz indirmek <https://visualstudio.microsoft.com/tr/downloads/> adresinin ziyaret edilmesi gerekmektedir.

Visual Studio IDE'sinin genel olarak üç çeşidi bulunmaktadır. Bunlar; "Topluluk", "Professional" ve "Enterprise" dir. Topluluk tipi IDE ücretsizken professional ve enterprise tipleri ücretlidir. Professional ve enterprise tipleri ücretli olsa da deneme sürümleri ücretsiz olarak indirilebilir. Professional tipi IDE küçük yazılım geliştirme ekipleri için ideal kullanıma sahipken enterprise tipi IDE büyük boyutlardaki yazılım geliştirme ekipleri için ideal kullanıma sahiptir. Bu derste C#'ın bireysel kullanımı anlatılacağından dolayı topluluk tipi Visual Studio IDE'si indirilecektir.

Visual Studio programını indirmek için öncelikle "Topluluk" sekmesinin altında "Ücretsiz indirin" butonuna tıklanması gerekmektedir. Daha sonra açılan sayfada bir exe dosyası bilgisayarınıza otomatik olarak indirilecektir. Bu exe dosyası Visual Studio'nun setup dosyası değildir sadece indirme yöneticisidir. Visual Studio indirme yönetici sayesinde Visual Studio programının istenilen bileşenleri indirilip yüklenebilir.

Visual Studio indirme yöneticisi ilk çalıştırıldığında öncelikle yazılımın lisans koşulları kabul edilmesi gerekmektedir. Bu aşamada istenirse hem lisans koşulları hem de yazılımın gizlilik bildirimine göz atılabilir. Visual Studio ile web, masaüstü ve mobil uygulamalar geliştirilebilmektedir. Bundan dolayı Visual Studio indirme yöneticisinde birçok seçenek bulunmaktadır. Visual Studio indirme yöneticisindeki seçenekleri kullanarak sadece C# dilinde program yazabilir veya hem C# hem de Python dilinde programlar yazılabilir. Bu derste C# dilini kullanılacağı için "Masaüstü ve Mobil" bölümünün altında bulunan ".NET masaüstü geliştirme" seçeneğini tıklanması gerekmektedir.

Visual Studio yükleme yöneticisinin "Bağımsız Bileşenler" sekmesi kullanılarak özel olarak ihtiyaç duyulabilecek bileşenler yüklenebilir. Buradaki alanlar bir önceki sekmede (İş yükleri) yapılan seçime göre otomatik doldurulmaktadır. Özel bir ihtiyaç olmaması halinde bu alanda herhangi bir değişiklik yapılmasına gerek yoktur.

VISUAL STUDIO İLK AÇILIŞI

Visual Studio programının yüklenme işlemi sonlandıktan sonraki ilk açılışta bazı ayarlar yapmanız istenecektir. Bu ayarlardan ilki Visual Studio programı tarafından kullanılacak tema tercihidir. Bu ekranda Visual Studio'nun kullanılacağı tema seçimi yapılmaktadır. Eğer uzun süre bilgisayar başında yazılım geliştiriyorsanız "Koyu" temayı seçebilirsiniz. Bu seçim daha sonra Visual Studio programı açıldıktan sonra da değiştirilebilmektedir.

Visual Studio karşılama ekranının sol tarafında daha önceden oluşturduğunuz veya açtığınız proje, dosya veya klasörler listelenmektedir. Karşılama ekranının sağ tarafında ise; "Depoyu klonla", "Bir projeyi veya çözümü aç", "Yerel bir klasör aç", "Yeni bir proje oluştur" ve "Kodsuz olarak devam et" seçenekleri bulunmaktadır.

Depoyu klonla: Bu seçenek daha önceden internet üzerinde oluşturulmuş bir projenin (GitHub veya Azure projeleri gibi) dosyalarını indirilip çalışmaya devam etmek için kullanılmaktadır.

Bir proje veya çözüm aç: Bu seçenek yerel bilgisayarda daha önceden oluşturulan bir projeyi açmak için kullanılmaktadır.

Yerel klasör aç: Bu seçenek bilgisayarda bulunan bir klasör içerisindeki dosyaları düzenlemek için kullanılmaktadır. Seçilen klasör içerisinde Visual Studio projesi olması zorunlu değildir.

Yeni bir proje oluřtur: Bu seenek yeni bir proje oluřturmak iin kullanılmaktadır. Bu nitenin devamında bu seenek kullanılarak anlatılacaktır.

Kodsuz olarak devam et: Bu seenek proje veya dosya amadan sadece Visual Studio programı amak iin kullanılmaktadır.

Karřılama ekranından “Yeni bir proje oluřtur” seeneęi tıklandıktan sonra karřımıza yeni proje oluřturma sihirbazı gelmektedir. Bu ekranın sol tarafında daha nceden oluřturulan proje řablonları yer almaktadır. řuan daha nceden hibir proje oluřturulmadıęı iin bu alan boş olarak gsterilmektedir. Yeni projeler oluřturduka bu alan dolacaktır. Ekranın saę tarafında ise Visual Studio programı kullanılarak oluřturulabilecek proje trleri listelenmektedir. Bu alanda ok fazla seenek olduęu iin st kısımda bulunan arama kutusu kullanılarak liste daraltılabilir.

Bu nitede C# anlatılacaęı iin ekrandaki arama kutusuna “Windows c#” yazarak listeyi daraltabilirsiniz. Yeni bir C# projesi oluřturmak iin listeden “Windows Forms Uygulaması” seilerek “Sonraki” butonuna tıklanması gerekmektedir.

Yeni proje oluřturma sihirbazındaki ikinci adımda projenin adını, konumu ve zmn adı iin varsayılan deęerler belirlenmektedir. İstenirse bu deęerler deęiřtirilebilmektedir. zellikle proje adı ve zm adının deęiřtirilmesi nerilmektedir. Proje adı yapılacak iřlemler iř ile alakalı olması faydalı olacaktır. nk bu sayede proje daha sonra aılırken projenin ierięini tahmin etmek kolaylařır.

Yeni proje oluřturma sihirbazının son adımda projede kullanılacak hedef ereve yani kullanılacak framework bilgisi seilmektedir. Framework seildikten sonra “Olurur” butonuna tıcklayarak yeni proje oluřturulabilmektedir.

Framework, yazılım ktphanelerinin birleřimi olarak tanımlanabilir. Microsoft tarafında oluřturulan “.NET Framework” ok geliřmiř bir ktphanedir. Bu ktphane sayesinde form nesnelerin kontrollerinden veri tabanı baęlantısına kadar birok iřlem kolayca gerekleřtirilebilmektedir. Microsoft’un “.NET Framework” ktphanesi zaman ierisinde gncellemektedir. Bu ktphanenin her srmnde kendine zg deęiřiklikleri bulunmaktadır. Yani “.NET Framework” 3.1 srm ile alıřan bir program 5.0 srm ile alıřamaz. Bundan dolayı srm seerken dikkat edilmelidir.

VISUAL STUDIO ANA PENCERESİ

zel bir ayar yapılmadıęa ana pencerenin st kısmında men, sol kısmında “Ara Kutusu”, saę kısmında “zm gezini” ve “zellikler” penceresi, orta kısmında ise aılan projenin tasarım grnts bulunmaktadır.

Visual Studio Mens

Visual Studio programın st mens sayesinde projenizi kaydedebilir, projenizi kapatabilir, proje ierisinde arama yapabilir ve projenizi alıřtırmak dahil birok iřlem yapabilirsiniz.

Dosya: Bu mende yeni proje oluřturma, projeyi kapatma, proje veya dosyaları kaydetme seenekleri bulunmaktadır.

Dzen: Bu menden proje ierisinde metin arama ve deęiřtirme iřlemi yapılabilir. Ayrıca seilen metni kesme, kopyalanan metni yapıřtırma veya geri alma gibi iřlemler yapılabilir.

Grnm: Bu mens sayesinde Visual Studio programında tasarımsal deęiřiklikler yapılabilir. Yani “zm gezgeni” veya “zellikler” gibi pencereleri aıp kapatılabilir.

Git: Bu men projenin dosyalarını internet zerinden hizmet veren (GitHub veya Azure) depolama alanlarına aktarım iin kullanılır.

Proje: Bu men sayesinde proje yeni form, bileřen veya sınıf gibi elementler eklenebilir. Ayrıca proje zelliklerine bakılarak gncellenebilir.

Derle: Bu men sayesinde oluřturulan projeler derlenerek alıřtırılabilir. Aynı zamanda derleme ayarları da bu menden yapılabilir.

Hata Ayıkla: Bu men projede oluřan kod hatalarını ortaya ıkarmak iin hata ayıklama iřlemleri yapılabilir.

Dzen: Bu men sayesinde form zerindeki nesnelerin (button veya label gibi) hizalanması, nesneler arası boşlukların dzenlenmesi gibi iřlemler yapılabilir.

Test: Bu men sayesinde projeye testler uygulayarak projedeki sorunlar belirlenebilir.

Analiz: Bu men sayesinde proje kodlarındaki yapısal bozukluklar (kod satırlarındaki tab eksiklikleri veya fazlalıkları gibi) otomatik olarak dzeltilebilir.

Aralar: Bu men Visual Studio’ya yeni zellikler eklemek, veri tabanına veya sunucuya baęlantı yapmak iin kullanılmaktadır.

Uzantılar: Bu men sayesinde Visual Studio mens zelleřtirilebilir, Visual Studio marketten yeni uzantılar edinilebilir veya var olan uzantılar ynetilebilir.

Pencere: Bu men kullanılarak Visual Studio’daki aık pencerelerin hizalanma veya pencereler arası geiř yapılabilir.

Yardım: Bu men kullanarak karřılařılabilecek sorunlar hakkında yardım talebinde bulunulabilir.

Ara Kutusu

Ara kutusunda projedeki formlara eklenebilecek nesneler listelenmektedir. Bu nesneler ara kutusundan tut srkle bırak yntemi ile forma eklenebilir.

Eklenebilecek nesne sayısı çok fazla olduğundan ve nesneler arasından kolay seçim yapabilmek için araç kutusundaki nesneler gruplandırılmıştır. Projeye eklenebilecek bütün nesneler “All Windows Forms” grubunun içerisinde bulunmaktadır. Grupların içerisinde bulunan nesneleri görebilmek için grup adının solunda bulunan üçgen şeklindeki simge tıklanmalıdır.

En çok kullanılan form nesnelerinden bazıları:

Button: Türkçe adıyla düğme olan bu nesne fare ile tıklandığında içe çökmüş bir görüntü vermektedir.

Genellikle kullanıcı verileri girdikten sonra butona tıklayarak gerekli işlemin (kaydetme veya güncelleme gibi) yapılmasını sağlar.

CheckBox: Kullanıcıya seçenekler sunup bu seçeneklerden bir veya birden fazla seçilmesi için kullanılan nesnedir.

RadioButton: Bu nesnede tıpkı CheckBox gibi kullanıcıya seçenekler sunmak için kullanılmaktadır. Ancak bu nesnede kullanıcılar sadece bir tane seçenek seçebilir.

ComboBox: Bu nesne sayesinde kullanıcılar açılır listeden seçim yapabilmektedirler.

DateTimePicker: Bu nesne kullanıcıdan tarih ve saat verisi almak için kullanılmaktadır.

TextBox: Bu nesne kullanılarak kullanıcıdan metin biçiminde tek satırlık veri alınabilmektedir.

RichTextBox: Bu nesnede tıpkı TextBox gibi kullanıcıdan metin biçiminde veri almak için kullanılmaktadır. Bu nesnenin TextBox nesnesinden farklı kullanıcıdan alınan veri birden fazla satır halinde olabilmektedir.

Label: Forma eklenen nesnelere (TextBox veya ComboBox gibi) açıklama eklemek için kullanılır. Bu nesnenin Türkçesi etikettir.

ListBox: Bu nesne bir liste oluşturarak kullanıcının bu listeden seçim yapmasını sağlamaktadır.

PictureBox: Bu nesne forma fotoğraf eklemek için kullanılmaktadır.

Çözüm Gezgini

Çözüm gezgini Visual Studio programında açılan bütün projeleri yönetmek için kullanılmaktadır. Çözüm gezgini kapandığında “Görünüm” menüsü altındaki “Çözüm gezginine” seçeneğine tıklanarak açılabilir.

Çözüm gezgininde projedeki öğeler hiyerarşik şekilde listelenmektedir. Çözüm gezgini kullanılarak projeye yeni öğeler de eklenebilmektedir. Yeni öğe eklemek için çözüm gezgininde farenin sağ tuşuna tıklayıp açılan menüden “Ekle” seçeneği seçilmelidir.

Özellikler Penceresi

Özellikler penceresi sayesinde projede bulunan formun veya forma eklenen nesnelerin özelliklerini görüp değiştirebilmektedir. Bu penceredeki özellikler listesi seçilen nesneye göre değişiklik göstermektedir.

Pencerenin sol tarafında özellik adı, sağ tarafında ise özelliğin değeri bulunmaktadır. Özelliklerin alabileceği değerler farklılıklar göstermektedir. Bu farklılığa göre özelliklerin değerleri metin, sayı veya listeden seçim halinde olabilir.

Özellikler penceresindeki en çok kullanılan özelliklerden bazıları:

Name: Bu özellik nesneye isim vermek için kullanılmaktadır. Nesneye isim verirken Türkçe karakterlerin (ş,ğ,ç veya ü gibi) kullanılmaması tavsiye edilmektedir. Çünkü kod üzerinden nesneye etki etmek için nesnenin “Name” özelliği kullanılmaktadır.

Size (Width, Height): Bu özellik kullanılarak seçilen nesnenin genişlik ve yüksekliğini değiştirilebilmektedir.

Text: Bu özellik kullanılarak nesnenin üzerinde yazılan metin değiştirilebilmektedir.

TextAlign: Bu özellik nesnenin üzerinde bulunan yazının konumunu belirlemek için kullanılmaktadır. Yani bu özellik kullanılarak nesne üzerindeki yazı sağa veya sola yaslanabilir.

Visible: Bu özellik nesnelerin görünürlüğünü belirlemek için kullanılmaktadır. Yani bu özellik nesnenin görünür veya gizli olmasını sağlamaktadır.

Image: Bu özellik sayesinde nesnenin üzerine bilgisayarda yüklü bir fotoğraf eklenebilmektedir.

Location: Bu özellik nesnenin form üzerindeki konumunu belirlemek için kullanılmaktadır.

Font (Name, Size): Bu özellik nesne üzerindeki yazının boyutunu ve fontunu değiştirmek için kullanılmaktadır.

BackColor: Bu özellik nesnenin arka plan rengini değiştirmek için kullanılmaktadır.

Olaylar (Events)

Olay, kullanıcı tarafından veya kodsız olarak tetiklenen eylemleri temsil etmektedir. Örnek olarak forma eklenen butonun kullanıcı tarafından fare ile tıklanması bir olaydır. C#'ın windows uygulamaları görselliğe dayalı olduğu için olaylar çok önemlidir.

En çok kullanılan olaylardan bazıları:

Click: Nesneye fare ile bir defa tıklandığında bu olay tetiklenmektedir. Bu olay genellikle Button gibi tıklanabilir nesnelerde kullanılmaktadır.

DoubleClick: Nesneye fare ile çift tıklandığında bu olay tetiklenmektedir. Bu olay genellikle Button gibi tıklanabilir nesnelerde kullanılmaktadır.

KeyPress: Kullanıcının klavyeden bir tuşa basıldığında bu olay tetiklenmektedir. Bu olay genellikle TextBox veya RichTextBox gibi klavyeden veri alan nesneler ile beraber kullanılmaktadır.

KeyDown: Bu olay da tıpkı KeyPress gibi kullanıcının klavyeden bir tuşa basıldığında tetiklenmektedir. KeyPress ile KeyDown arasındaki fark ise KeyDown olayı klavyeden tuşa basıldığı anda tetiklenirken, KeyDown olayı klavyeden tuşa basılıp kaldırıldığı anda tetiklenmektedir.



DERS ADI Görsel Programlama I
ÜNİTE ADI Windows Formları ve Özellikleri
ÜNİTE NO 3
YAZAR Öğr. Gör. DAHA ORHAN

WINDOWS FORMLAR

C# programlama dili ile Windows tabanlı bir uygulama gerçekleştirmek için Visual Studio 2019 programını başlattıktan sonra sırasıyla Yeni Bir Proje Oluştur seçeneğini işaretleyerek karşımıza gelen ekrandan Windows Forms Uygulaması (.Net Framework)'ü seçmemiz gerekmektedir.

Windows formlar genel olarak iki ana başlık altında incelenmektedir. Bunlar SDI (Single Document Interface) formlar ve MDI (Multiple Document Interface) formlardır. Yani tek formlu arayüzler ve çok formlu arayüzlerdir. Bunlar Ana ve Çocuk Form olarak da adlandırılmaktadırlar.

SDI Formlar

SDI formlar İngilizce "Single Document Interface" kelimelerinin baş harfleri kullanılarak isimlendirilen tek belgeli arayüze sahip formlardır. Başka bir forma ihtiyaç duymaksızın tek başına çalışabilen bir yapıdadırlar. Yeni başlattığımız bir projede ilk olarak ekrana gelen formlar SDI formlardır.

MDI Formlar

Microsoft firması tarafından İngilizce Multiple Document Interface kelimelerinin baş harflerinden oluşturulmuş bir kavramdır. Türkçe karşılığı çoklu doküman arayüzü anlamına denk gelmektedir. Yani MDI formlar bir ana pencere altında bu ana pencereye bağlı olarak bir veya birden fazla çocuk pencere çalıştırabilen yapılardır.

FORM ÖZELLİKLERİ

Visual Studio ortamında ekranın sağ tarafına yerleşen Özellikler paneli üzerinden, çalışma esnasında aktif nesneye ait olan özellikleri ve bu özelliklerin o an itibarıyla değerlerinin görüntülediği bir alan bulunmaktadır. Bu alan üzerinden formumuza ait özellikleri değiştirme imkânına sahibiz. Aynı zamanda forma ait özellikler kod editöründen de değiştirilebilmektedir.

Özellikler paneli iki farklı görünümde nesnelere ait özellikleri listeleyebilir. Bunlardan biri Alfabetik seçeneğidir. Bu seçenek ile aktif olan kontrolün özellikleri panel içinde alfabetik olarak sıralanır. Diğer bir listeleme şekli ise Kategorilere Ayrıldı seçeneğidir.

Text Özelliği

Formun başlık çubuğunda görüntülenen isminin ayarlandığı özellik seçeneğidir. Ancak burada dikkat etmemiz gereken nokta bu isim sadece projemizin tasarım kısmındaki isimdir. Text özelliği haricinde bir değişiklik yapılmadıysa kod editöründe çalışırken formumuzun ismi programın varsayılan olarak atadığı (Form1, Form2 vb.) isimdir.

ControlBox Özelliği

Bu özellik program çalışırken anlaşılabilir bir yapıdadır. Mantıksal olarak değer alabilen iki seçenek söz konusudur. Bunlardan biri True (doğru veya 1) diğeri ise False (yanlış veya 0) değeridir.

Enabled Özelliği

Controlbox özelliğinde olduğu gibi True ve False şeklinde iki farklı değer alabilen mantıksal bir büyüklüktür. Enabled özelliğinin True olması durumunda form aktif konumdadır ve formun üzerindeki nesneler ile işlem yapılması mümkündür. Formun enabled özelliğini false yapacak olursak bu defa form pasif konuma geçecektir. Bununla birlikte forma yerleştirdiğimiz nesneler de aynı şekilde pasif konumda olacaktır.

Font Özelliği

Font özelliği ile form üzerindeki nesnelere ait text özellikleri, form üzerindeki yazı stilleri, yazı font özellikleri ve yazı boyutu istenilen ölçüde ayarlanmaktadır. Böylelikle form ve form üzerindeki nesneler belirlenen parametreler doğrultusunda görüntülenmektedir. Bu değişiklikleri gerçekleştirmek için font özelliğinin sağ tarafında bulunan ... butonuna tıklamak gerekmektedir.

AutoScroll Özelliği

AutoScroll özelliği form üzerinde bulunan nesnelerin çok fazla olması durumunda veya kullanıcının form alanını tutamaçlar yardımıyla daralttığı durumda yatay ve dikey kaydırma çubuklarının aktif veya pasif olmasını sağlamaktadır. True veya false değerlerini alabilir. True olduğu durumda form üzerinde görünmeyen nesnelere kaydırma çubukları yardımıyla ulaşılabilir.

Locked Özelliği

Form üzerindeki nesneler hizalanırken, tasarım yapıldığı süre içinde formun kilitli olması istenebilir. Böyle bir durumda locked özelliğine ait değer true olarak ayarlanmalıdır. Bu durumda formun konumu ve boyutu programcı tarafından değiştirilemez.

StartPosition Özelliği

StartPosition özelliği ile projeye ait başlangıç formunun ilk açılıştaki ekran üzerinde konumlanacağı alana yönelik ayarlamalar yapılmaktadır. Bu işlem startPosition özelliğinin 5 farklı seçenek sunmasıyla gerçekleştirilmektedir. Bu seçenekler sırasıyla: Manual, CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds ve CenterParent seçenekleridir.

MaximizeBox, MinimizeBox ve WindowState Özellikleri

Projelerimizde hazırlanan formların çalışma esnasında tam ekranı kaplaması, simge durumunda küçülmesi veya görev çubuğunda görünmesiyle ilgili özellikler ayarlanabilmektedir. Bu durumda özellikler panelinde bulunan MaximizeBox, MinimizeBox ve WindowState özelliklerini kullanmamız gerekmektedir. Bu üç farklı özelliğin tek başlık altında anlatılmasının sebebi arayüzdeki etkilerinden kaynaklanmaktadır.

Opacity Özelliği

Bu özellik 0-100 arasında değer alabilen bir yapıdadır. Aksi değerler girildiği takdirde hata mesajı vermektedir. İşlerliğinden bahsedecek olursak, formumuzun saydamlığını ayarlamamıza imkân sağlamaktadır. En küçük değer olan sıfır değerini yazdığımızda formun arkasında ne olduğunu tam olarak görebiliriz ancak forma ait işlemlerimizi gerçekleştirme noktasında güçlük çekebiliriz. Yüz değerini yazdığımızda ise formun arkası görülmemektedir.

ShowInTaskbar Özelliği

Bu özellik projemizin simgesinin çalışma esnasında görev çubuğu üzerinde bulunup bulunmayacağını ayarlamamıza imkân sağlar. True veya False olmak üzere iki farklı değer alabilir. ShowInTaskbar özelliği True olarak ayarlandığında proje simgesi görev çubuğu üzerinde görüntülenmektedir. Aksi takdirde proje simgesi görev çubuğunda görünmeyecektir.

AcceptButton ve CancelButton Özellikleri

Projelerimizde bazen ilgili kutucukları klavye ile doldurup onay işlemi vermemiz gerekebilir. Durum tersi şeklinde ilerleyerek iptal etmemiz de söz konusu olabilmektedir. Bu gereklilikler karşısında onay ve iptal işlemleri için formumuza butonlar yerleştiririz. Burada kullanacağımız özellik bu butonları fare yardımı kullanmadan işlevsel yapmamızı yani klavyeden ayrılmadan onaylamamızı veya iptal etmemizi sağlayacaktır.

BackColor Özelliği

Formun arka plan rengini değiştirme gereği duyulduğunda kullanılan bir özelliktir. Bu özelliğin yanındaki oka tıkladığımızda Custom, Web ve System olmak üzere üç sekme altında bize renk alternatifi sunmaktadır.

Cursor Özelliği

Fare işareti, yapacağı göreve veya bulunduğu duruma göre farklı şekillerde olabilmektedir. Projemizin çalışması esnasında fare ile form üzerinde gezinirken farenin görünmesini istediğimiz şekli belirlemeye yarayan özellik cursor'dür.



DERS ADI Görsel Programlama I
ÜNİTE ADI Değişken Tanımlama ve Veri Tipleri
ÜNİTE NO 4
YAZAR EMİR SULTAN GÖKKAYA

DEĞİŞKEN

Program yaşam döngüsü içerisinde belirli bir veri türünün değerini içeren yapılara değişken adı verilir. Değişkenler bellek üzerinde geçici olarak saklanırlar ve program yaşam döngüsünün sona ermesiyle birlikte bellekten düşerler. Her değişkene, bellek üzerinde o değişkenin tutacağı veri tipine yetecek büyüklükte bir yer ayrılır. Bu yere değişkenin adresi adı verilir.

Değişken Tanımlama Kuralları

C# programlama dilinde değişken tanımlanırken uyulması gereken birtakım kurallar bulunmaktadır. Bu kurallar aşağıdaki listede yer almaktadır.

- ☐ Değişken ismi yalnızca harf, rakam ve _ (alt çizgi) içerebilir.
- ☐ Değişken ismi harf ile başlamak zorundadır.
- ☐ Değişken ismi 255 karakterden fazla olamaz.
- ☐ Değişken ismi tanımlanırken karakterler arasında boşluk bırakılamaz.
- ☐ C# büyük küçük harf duyarlı bir programlama dili olduğundan; Değişken hangi isimle tanımlanmış ise o isimle kullanılabilir.
- ☐ Komut ifadeleri değişken adı olarak kullanılamaz. Örneğin; AND, OR ve NOT gibi.

Değişken İsimlendirme Standartları

Kurallara uygun şekilde tanımlanmış bir değişkene verilen isim; Her ne kadar programın çalışması için bir sorun teşkil etmese de tutarlı bir görünüm, kod bütünlüğü ve kodların rahat okunup anlaşılabilmesi açısından önem arz etmektedir. Bu sebeple resmi olarak belirlenmiş 2 (iki) adet isimlendirme standardı bulunmaktadır. Bunlar; Pascal Case ve Camel Case'dir.

Pascal Case

Değişken adını oluşturan her kelimenin baş harfinin büyük yazılarak değişkenin isimlendirildiği standarttır.

Camel Case

Değişken adını oluşturan ilk kelimenin baş harfi küçük, diğer kelimelerin baş harfleri büyük yazılarak değişkenin isimlendirildiği standarttır.

Değişken Tanımlama

Her programlama dilinin bir syntax (sözdizimi) yapısı bulunmaktadır. Yazılan kodlar bu sözdizimi yapısına uygun olarak yazılmak zorundadır. C# dilinde ise değişken tanımlamak için kullanılan syntax (sözdizimi) yapısı;

şeklinde dir. İlk olarak değişken tipini tanımlayan ifade yazılır ardından değişken adı belirlenerek tanımlama işlemi gerçekleştirilir.

Değişkene Değer Atama

Değer atama işlemi, değişkeni tanımlama esnasında yapılabileceği gibi daha sonrada gerçekleştirilebilir. Bir değişkene değer atamak için C# da atama operatörü olan "=" eşittir sembolü kullanılır.

DEĞİŞKEN TİPLERİ

Değişkenler değer tipi ve referans tipi değişkenler olmak üzere 2 (iki) gruba ayrılırlar. Değer tipi değişkenler bellekte stack adı verilen alanlarda saklanırken referans tipi değişkenler heap adı verilen alanlarda saklanır.

Değer Tipleri

Değer tipleri; veriyi taşıyan ve taşıdığı veriye göre bellek üzerinde yer dolduran değişken tipleridir. Değer tipi değişkenler; Tamsayı Değişken Tipleri, Ondalık Sayı Değişken Tipleri, Bool Değişken Tipi ve Char Değişken Tipi'dir.

Tamsayı Değişken Tipleri

Tam sayıları saklamak ve üzerlerinde işlem yapabilmek için kullanılan değişken tipleridir. Bunlar; byte, sbyte, short, ushort, int, uint, long, ulong, nint ve nuint'dir.

Ondalık Sayı Değişken Tipleri

Ondalık sayıları saklamak ve üzerlerinde işlem yapabilmek için kullanılan değişken tipleridir. Bunlar;

float, double ve decimal'dir.

Bool Değişken Tipi

Bool veri tipi yalnızca iki değer alabilir. Bunlar "true" ve "false"dur. Bellekte 1 byte yer kaplar. Bool değişken tipi, programın akışının yönlendirilmesinde ve döngü yapılarında sıkça kullanılmaktadır.

Char Değişken Tipi

C#'da karakterler, "Unicode" adı verilen ve uzunluğu 2 byte olan bir sistemle belirlenir. Bu sistem içerisinde tüm alfabeler, sayılar, özel karakter ve semboller yer almaktadır. "Char" veri tipi de bu "Unicode" karakterler içerisinde sadece tek bir değer alabilen bir değişken tipidir.

Referans Tipleri

Referans tipleri; bellek bölgesinde veri yerine adresi tutarlar ve o adresin gösterdiği yerde de veri tutulur.

Referans tipi değişkenler; string, object ve dynamic'dir.

Değişken Tiplerini Öğrenmek

Değişkenin sahip olduğu veri tipini öğrenmek, doğruluğunu kontrol etmek ve program akışını yönlendirmek gibi nedenlerden dolayı değişkenin tipinin çalışma zamanında öğrenilmesine ihtiyaç duyulur. Bunun için kullanılabilecek 2 (iki) adet yardımcı metod bulunmaktadır. Bunlar; GetType ve typeof metodlarıdır.

GetType Metodu

Bir değişkenin tipinin öğrenilmesinde kullanılır. "Type" türünden sonuç üretmektedir.

typeof Metodu

Bir değişken tipinin başka bir değişken tipi ile karşılaştırılması ya da kontrol edilmesinde kullanılır.

DEĞİŞKENLERİN GEÇERLİLİK BÖLGESİ

Değişkenler tanımlanmış oldukları {...} (küme parantezi) bloğu içerisinde geçerlidirler. Değişkenlere tanımlı oldukları blok içerisinde ve alt bloklardan ulaşılabilirken üst bloklardan alt bloktaki değişkenlere erişilemez.

TİP DÖNÜŞÜMLERİ (CASTING)

Tip dönüşümü, bir veri tipinin bir başka veri tipine dönüştürülmesidir. C# programlama dilinde 4 (dört) adet tip dönüşümü bulunmaktadır. Bunlar; Örtülü/Otomatik Dönüşüm (Implicit Conversions), Açık/Manuel Dönüşüm (Explicit Conversions), Kullanıcı Tanımlı Dönüşüm (User-defined conversions) ve Yardımcı Sınıflarla Dönüşüm (Conversions with helper classes)'dür.

Örtülü/Otomatik Dönüşüm (Implicit Conversions)

Dönüşecek değişkenin bellekte kapladığı alan, dönüştürülmek istenen değişken tipinin kapsadığı alandan daha küçük ise bu dönüşüm işlemi C# tarafından otomatik olarak gerçekleştirilir. Bu dönüşüm işlemine de Örtülü/Otomatik Dönüşüm (Implicit Conversions) denir.

Açık/Manuel Dönüşüm (Explicit Conversions)

Dönüşecek değişkenin bellekte kapladığı alan, dönüştürülmek istenen değişken tipinin kapsadığı alandan daha büyük ise bu dönüşüm işlemi C# tarafından otomatik olarak gerçekleştirilemez. Dönüşüm (casting) ifadesi kullanılarak gerçekleştirilen bu dönüşüm işlemine Açık/Manuel Dönüşüm (Explicit Conversions) adı verilmektedir. Ayrıca Açık/Manuel Dönüşüm (Explicit Conversions) işlemleri için "Parse", "TryParse" ve "Convert" yöntemleri de kullanılmaktadır.

Parse Metodu

"String" veri tipindeki bir değişkenin değerini farklı bir değişken türüne dönüştürmek için kullanılır. "Parse" metodu kullanılırken dönüşecek değişkenin değerinin "null" olmaması ve değerinin hedef değişkenin değer aralığında olması gerekmektedir.

TryParse Metodu

İşlevi "Parse" komutu ile aynıdır. Tek farkı; Dönüşecek değişkenin değerinin "null" veya değer aralığı dışında olması ya da değişkenin farklı bir değişken tipine ait olması durumunda meydana gelecek hatada geriye 0 (sıfır) değeri döndürmesidir. "out" eki ile birlikte kullanılır ve sonuç "out" referanslı değişkene aktarılır.

Convert Sınıfı

Değişken tipleri arasında dönüşüm yapmak için kullanılan bir sınıftır. "Convert" sınıfının metodları kullanılarak yapılabilecek tip dönüşümleri Tablo 4.7.'de listelenmiştir.



DERS ADI Görsel Programlama I
ÜNİTE ADI Karar Yapıları ve Döngüler
ÜNİTE NO 5
YAZAR Öğr. Gör. ÇAĞRI BURAK ARINÇ

KARAR YAPILARI

Karar yapıları bilgisayarın iki ya da daha fazla akış bulunması durumunda seçim yapabilmesini sağlayan mantık yapılarıdır. Karar yapıları olmadan yapılacak işlemler basit hesaplamaların ötesine geçemeyecektir. Karar mekanizması günlük yaşamımızda yaptığımız her seçime benzer özellikler sağladığından anlaşılması kolay mantıksal gruptandır. Günlük yaşantımız ile bu mantıksal yapıyı açıklamak istersek hayatımızda yaptığımız seçimler bu mekanizma için doğru bir tanımlama olacaktır. Örneğin bilgisayar satın almak istediğimizde kendimize özellik listesi yapar ve bu doğrultuda inceleme yapmaya başlarız, karar mekanizması bu aşamada karşımıza çıkar ve şu sorunun cevabını arar eğer ürün listemde ki özelliklere sahipse alabilirim ya da eğer sahip değil ise başka bir ürün incelemeliyim.

C# programlama dilinde karar yapıları incelediği zaman karşımıza iki farklı yapı çıkmaktadır bunlar: “if” (eğer) – “else” (başka) ve “switch – case” dir. Bu yapılar kullanılarak tek koşullu, çok koşullu ve iç içe karar yapıları oluşturulabilir.

If-else Yapısı

Programlama temellerinde yazılan kodların belli bir akış içerisinde ilerlediğinden bahsetmiştik. Bu akışın farklılaştırılabilmesi için gerekli olan yapı “if” (eğer) ve “else” (başka) ile gerçekleşmektedir. “If” ve “else” kod bloğu verinin doğru yada yanlış olmasını baz alarak akışı değiştirir ve bazı kod bloklarının çalıştırılmamasını sağlar.

Switch Case Yapısı

“Switch-case” yapısı, farklı durumların dallanması ile oluşan ifadelerde “if-else” blokları yerine kullanılabilen yapılardandır. “Switch –case” yapısı ile tasarlanan kod blokları “if-else” yapısı ile de yapılabilecek olsa da bu yapı anlaşılabilirlik açısından daha sade ve karmaşıklığı önleyebilecek bir yapıya sahip olması nedeniyle kullanılmaktadır.

DÖNGÜLER

Programlama dillerinde tasarlanan akış birçok kez çalıştırılmak zorunda kalınabilir. Tekrarlanan akışı çalıştırmak için kod bloğu bir kez yazılır ve döngü ifadeleri ile tekrar tekrar çalıştırılır. Programlama dillerinin temel yapılarından olan döngüler neredeyse her dilde aynı yapıya sahiptir. C# programla dili incelendiği zaman karşımıza dört farklı döngü yapısı çıkmaktadır. Döngü yapıları “for”, “while”, “do-while” ve “foreach” olarak bu dil içerisinde tanımlanmıştır.

For Döngüsü

C# dilinde bulunan “for” döngüsü bir ifadenin ya da kod bloğunun birden fazla akış yapabilmesini sağlayan ifadedir. “For” döngüsü sayesinde birden fazla iç içe döngü de kurulabilmektedir. For döngüsü, belirtilen koşul ifadesi “false” (yanlış) değer döndürene kadar art arda ifadeyi yürütür.

While Döngüsü

C#, belirtilen koşul ifadesi false (yanlış) değer döndürdüğü sürece kod bloğunu tekrar tekrar yürütmek için while döngüsünü kullanır. For döngüsünde olduğu gibi while döngüsünde de birden fazla döngü iç içe kullanılabilir.

Do-while Döngüsü

C# programlama dilinde kullanılan do-while döngüsünün yapısı “while” ile benzerlik göstermektedir. “Do-while” döngüsü koşulun doğruluğunu kontrol etmeden en az bir kere çalıştırılması durumunda kullanılan bir yapıdır. Bu durumun oluşmasının sebebi program akışının ilerle biçimine dayanmaktadır. “While” döngüsünde koşul ifadeleri en başta kontrol edilirken “do-while” döngüsünde işlem yapıldıktan sonra döngü şartları kontrol edilir bu nedenle koşul ifadesi ne sonuç verirse versin işlem sonucu çıktı olarak alınır.

Foreach Döngüsü

“Foreach” döngüsü dizi (array) ve koleksiyon (collection) tabanlı nesnelerin elemanlarına erişip bu elemanlar üzerinden işlem yapılabilmesini sağlar..



DERS ADI Görsel Programlama I
ÜNİTE ADI Tip Dönüşümleri
ÜNİTE NO 6
YAZAR SEDA KAYA

Tip Dönüşümü Nedir?

Programlama dillerinde, daha önceki ünitelerde de gördüğünüz üzere birçok veri tipi mevcuttur. Her veri tipinin kendi içinde yapılabilecek birçok işlem vardır. Aynı veri tipleriyle işlem yaptığımız gibi bazı durumlarda farklı veri tipleriyle de işlem yapmak isteyebiliriz. Normalde farklı veri tipleri arasında işlem yapılmazken işlem yapılması gereken durumlarda veri tipinin dönüştürülmesi gerekmektedir. Örneğin; güncel yaşı hesaplamak için int veri tipindeki bir yıl sayısından, textboxtan alacağımız doğum yılını çıkarmak isteyebiliriz. Bu durumda textboxtan string veri tipinde bir veri geleceği için bu değeri önce int veri tipine dönüştürmemiz gerekir. Ya da date tipindeki bir tarihi ekrana string veri tipinde yazdırmak isteyebiliriz.

Herhangi bir veri tipindeki bir veriyi, bir değişkende saklamak istediğimiz zaman verinin veri tipinde bellekte bir yer açılır. Bu yüzden bir değişkenin veri tipini dönüştürmek istediğimiz zaman, verinin bellekteki yerini değiştiririz. Veri küçük bellek adresinden büyük bellek adresine de taşınabilir; büyük bellek adresinden küçük bellek adresine de taşınabilir. Büyük bellekten küçük bellek adresine taşındığında da veri kayıpları meydana gelir.

Tip dönüşümleri, yapıma şekline göre ikiye ayrılır. Bunlar; bilinçli tip dönüşümü ve bilinçsiz tip dönüşümüdür. Bazı kaynaklarda bilinçli dönüşüm için açık dönüşüm, bilinçsiz dönüşüm için de kapalı dönüşüm ifadesini görebilirsiniz. Aynı manaya gelirler. Bu ünite de veri tipi dönüşümlerini detaylı bir şekilde inceleyeceğiz.

Bilinçsiz(Implicit) Tip Dönüşümü

Herhangi bir operatör kullanmadan derleyicinin kendisinin yaptığı dönüşüme bilinçsiz dönüşüm denir. Bilinçsiz dönüşümde küçük bellekli bir veri tipinden büyük bellekli bir veri tipine dönüşüm yapılabilirken büyük bellekli veri tipinden küçük bellekli veri tipine dönüşüm yapılamaz. Bu durumda hata alırız.

Veri tiplerinin hepsinin birbirine dönüşümü mümkün değildir. Büyük veri tipleri küçük veri tiplerine dönüştürülürken veri kaybı olacağından derleyici hata verir. Küçük veri tipleri büyük veri tiplerine dönüştürülürken; büyük tipten dolayı eklenen fazla bitler sıfır ile doldurulur. Bu durum değişken içinde bulunan veriyi etkilemeyeceği için veri kaybı olmaz (Algan, 2013).

Bilinçli(Explicit) Tip Dönüşümü

Bilinçsiz dönüşümde veri tipleri uygun veri tiplerine derleyici tarafından dönüştürülebiliyordu. Ama her veri tipi her veri tipine dönüştürüleliyordu. Büyük veri tiplerini küçük veri tiplerine dönüştürmeye çalıştığımız zaman hata ile karşılaşılıyorduk. Bazı durumlarda büyük veri tiplerini küçük veri tiplerine dönüştürmek isteyebiliriz. O zaman yapmamız gereken bir dönüştürme operatörü kullanmaktır. Bilinçli tip dönüşümünde, kendisine dönüştürülmesi amaçlanan veri tipi parantez içine alınır ve yanına değişken ya da sabit değer yazılır. Bu durumda veri kaybını göze almış ve verimizi istediğimiz türe dönüştürebilmiş olacağız. Tabi bu ancak sayısal değerler içinde yapılabilir.

Küçük veri tipindeki veriyi büyük veri tipine bilinçli dönüştürmenin, bilinçsiz dönüştürmeden bir farkı yoktur. Bir dönüşüm operatörü kullansak da kullanmasak da aynı sonucu verecektir.

Dönüşüm Metotları

- **ToString() Metodu:** ToString metodu ile herhangi bir veri tipi string veri tipine dönüştürülebilir. Ekrana basılacak değer string veri tipinde olması gerektiği için bu metot çokça kullanılır. Yukarıdaki kodları tekrar incelediğinizde label' a yazdırdığımız tüm değerleri önce string' e çevirdiğimizi görebilirsiniz. `degiskenAdi.ToString()` şeklinde yazarak dönüşüm yapabilirsiniz.
- **Parse Metodu:** Parse metodu ile string veri tipi herhangi bir veri tipine dönüştürülebilir. `hedefVeriTipi.Parse(degiskenAdi)` şeklinde yazarak dönüşüm yapabilirsiniz.
- **Convert Metotları:** Önceki konularda anlatılan C# dönüştürme metotlarının haricinde .NET sınıf kütüphanesindeki özel metotlar da tip dönüştürme için kullanılır. Convert isimli bu sınıf, dönüşümün anlamlı olacağı tüm veri tiplerinin dönüştürülmesinde kullanılabilir (Algan, 2013). Aşağıda metotlar ve açıklamaları listelenmiştir.

- Convert.ToBoolean(x): x nesnesini bool tipine dönüştürür.
- Convert.ToByte(x): x nesnesini 8 bitlik byte tipine dönüştürür.
- Convert.ToSbyte(x): x nesnesini 8 bitlik sbyte tipine dönüştürür.
- Convert.ToInt16(x): x nesnesini 16 bitlik short tipine dönüştürür.
- Convert.ToUInt16(x): x nesnesini 16 bitlik ushort tipine dönüştürür.
- Convert.ToInt32(x): x nesnesini 32 bitlik int tipine dönüştürür.
- Convert.ToUInt32(x): x nesnesini 32 bitlik uint tipine dönüştürür.
- Convert.ToInt64(x): x nesnesini 64 bitlik long tipine dönüştürür.
- Convert.ToUInt64(x): x nesnesini 64 bitlik ulong tipine dönüştürür.
- Convert.ToSingle(x): x nesnesini float tipine dönüştürür.
- Convert.ToDouble(x): x nesnesini double tipine dönüştürür.
- Convert.ToDecimal(x): x nesnesini decimal tipine dönüştürür.
- Convert.ToChar(x): x nesnesini char tipine dönüştürür.
- Convert.ToString(x): x nesnesini string tipine dönüştürür.
- Convert.ToDateTime(x): x nesnesini date time tipine dönüştürür.



DERS ADI Görsel Programlama I

ÜNİTE ADI Operatörler

ÜNİTE NO 7

YAZAR Öğr. Gör. ÇAĞRI BURAK ARINÇ

Operatörler

Operatörler, kendi başlarına bir anlam ifade etmeyen ancak başta matematiksel ve mantıksal olmak üzere çeşitli işlemleri gerçekleştirmek için kullanılan özel karakter ya da sembollerdir. Örneğin $a+b$ ifadesinde $+$ işareti bir operatördür. Operatörlerin işlevlerini yerine getirebilmesi için değişkenlere ihtiyaç duymaktadır. Operatörlerin etki ettiği değişkenlere operand denir.

Aritmetik Operatörler

Toplama, çıkarma, çarpma, bölme ve mod gibi temel matematiksel işlemleri gerçekleştirmek için aritmetik operatörler kullanılır. Aritmetik operatörlerden olan $+$ ve $-$ işaret operatörü olarak da isimlendirilmektedir. Sayısal değerlerin sıfırdan küçük olduğu durumlarda başına $-$ operatörü koyulan sabitin veya değişkenin negatif olduğu belirtilir. Pozitif olan sayılar için isteğe bağlı olarak sayının başına $+$ işareti konulmaktadır. Önünde işaret olmayan sayılar varsayılan olarak pozitif kabul edilmektedir. Toplama ve işaret operatörü olarak da kullanılan $+$ işaretine en az iki string veya char operand verilmesi durumunda bu iki değeri birleştirip tek bir değer haline getirmektedir. Ayrıca sayısal değerleri birer artırmak ya da azaltmak amacı ile $++$ ve $--$ operatörleri de kullanılmaktadır.

Atama operatörleri

Atama operatörleri bir değişkene değer atamak amacıyla kullanılır. Atama operatörlerinin temel karakteri $=$ dir. Bazı durumlarda değişkenin varolan değeri üzerinde toplama, çıkarma gibi işlemler yapılabilir. Bu gibi durumlarda eşittir ($=$) operatörünün soluna istenilen aritmetik işlemi temsil eden operatör konur. Bu tip operatörlere işlemlili atama operatörü de denebilir.

Mantıksal Operatörler

Sıklıkla kullanılan operatör gruplarından biri de mantıksal operatörlerdir. Özellikle program akışını yönlendirirken kurulan döngü ve kontrol yapılarında mantıksal operatörlerden yararlanılır. Elde edilmek istenilen verinin mantıksal olarak sınanması bu operatörler sayesinde gerçekleştirilir. Mantıksal sınanma karar verme mekanizması ile doğru orantılıdır. Karar verme durumunda karşımıza iki farklı durum çıkmaktadır. Örneğin a sabiti b sabitinden büyük ise x işlemi değil ise y işlemi gerçekleştirmek gibi. Bu işlemlerin gerçekleşmesi için gerekli döngüler kurulurken döngü içerisinde mantıksal operatörlerden yararlanılması gerekmektedir.

Bitsel Operatörler

Bitsel operatörler $\&$ (bitsel ve), \sim (bitsel değil), $|$ (bitsel veya), \wedge (bitsel özel veya) operatörleridir. Sayıların kendileri yerine bitlerini yani ikilik tabanda ki karşılıklarını kullanan operatörlerdir. Bitsel operatörlerin kullanılabilmesi için değerlerin tam sayı olması gerekmektedir. $\&$ (ve) operatörü incelendiğinde çalışma mantığında karşılıklı sayıların 1 olması durumunda ilgi basamak 1 diğer durumlarda 0 olarak saklanmaktadır. \sim (değil) operatörünün çalışma mantığında ise ikilik sayı sisteminde bulunan her basamağın tam tersi saklanır. $|$ (veya) operatöründe karşılıklı sayıların herhangi birisinin 1 olması durumu aranır ve ilgili basamak 1 olarak saklanır diğer durumlarda 0 olarak saklanır. \wedge (özel veya) operatörü karşılıklı basamakların farklı olması durumunda ilgili basamak 1 diğer durumlarda 0 olarak saklanır.



DERS ADI Görsel Programlama I
ÜNİTE ADI Windows Form Kontrolleri I
ÜNİTE NO 8
YAZAR Öğr. Gör. DAHA ORHAN

STANDART KONTROLLER

Windows form projelerinde gerek programcının gerek kullanıcının ihtiyaç duyduğu bilgilendirme, etkileşim, bilgi girişi gibi birçok işlem bulunmaktadır. Bu işlemleri gerçekleştirmek üzere Visual Studio .Net içerisinde kullanıma hazır bir şekilde geliştirilen pek çok kontrol mevcuttur.

Label Kontrolü

Bazı durumlarda form üzerinde bulunan nesnelerin ne işe yaradıklarını ifade etmek veya belirli konularda kullanıcıya bilgi vermek için etiketleme işleminin yapılmasına ihtiyaç duyulmaktadır. Böyle durumlarda label kontrolünün kullanımı ihtiyaca cevap verebilmektedir. Genellikle içeriği sabit kalıp üzerinde değişiklik yapılmayacak bilgilerin gösteriminde kullanılmaktadır.

Label kontrolü form üzerine eklendikten sonra özellikler panelinde bulunan Text özelliği sayesinde görüntülenecek metin düzenlenebilmektedir. Bu işlem genellikle proje çalıştırılmadan önce sabit bir şekilde ayarlanmaktadır.

LinkLabel Kontrolü

Projelerimizde bazı durumlar için web sitesine bağlantı yapılması gerekebilir. Böyle durumlarda LinkLabel kontrolü kullanılmaktadır. Form üzerinde görüntülenmesi istenilen bağlantı metni, linklabel kontrolünün Text özelliğine yazılmalıdır. Ancak text özelliğine yazılan metin bağlantının yapılmasını sağlamamaktadır. Bu alan sadece kullanıcının görüntülediği isimdir. LinkLabel kontrolüne tıklandığı zaman yönlendirilecek olan bağlantı adresi kod editöründe belirtilmelidir. Özellikler panelinde bağlantı adresini yazacağımız bir alan bulunmamaktadır.

TextBox Kontrolü

Programcı tarafından görüntülenmesi istenen bilgilerin yer alabileceği ve kullanıcı tarafından bu bilgilerin değiştirilerek bilgi girişi yapabileceği metin kutularıdır. Özellikler panelinde bulunan Text özelliği sayesinde 255 karakteri aşmamak üzere görüntülenecek metin girişi yapılabilir.

TextBox kontrolünün özellikleri

Textbox kontrolünün içerisindeki metnin kullanıcı tarafından değiştirilmesini istemediğimiz durumlar olabilir ya da projemizde bulunan şartlara göre bilgi girişinin açık veya kapalı olmasını sağlayabiliriz. Bu durumda kontrolün ReadOnly özelliğini ayarlamamız gerekmektedir.

Her bir karakterin özel bir simge ile gösterilmesi için PasswordChar özelliği kullanılır.

CharacterCasing özelliği ile TextBox içine yazılan yazıların büyük ya da küçük harflerle yazılması mümkün olmaktadır.

TextBox içerisine girilen metinlerin yazı tipini ayarlamak için özellikler panelinde bulunan Font seçeneğini kullanabiliriz.

TextBox kontrolünün çok satırlı olmasını Multiline özelliği ile ayarlayabiliriz.

Lines özelliği sayesinde Textbox kontrolü bir dizi değişkeni gibi işlev görebilmektedir.

Windows form uygulamalarında kullanıcının veri girişi için belirli sınırlamalar getirmek istenebilir. Böyle bir durumda TextBox kontrolü için girebileceğimiz azami karakter sayısı MaxLengt özelliği ile ayarlanabilmektedir.

TextBox kontrolünün metodları

TextBox kontrolü kes, kopyala, yapıştır vb. pano işlemlerini metotlar yardımıyla gerçekleştirebilmektedir.

Sık kullanılan metotlar: SelectAll, cUt, Copy, Paste, Undo, Clear.

TextBox kontrolünün olayları

Çalışma anında TextBox kontrolü üzerinde yapılan değişiklikleri izleme ihtiyacımız olabilir. Bu durumda TextBox kontrolüne ait olaylardan istifade etmemiz gerekmektedir. TextBox kontrolüne ait birçok olay mevcuttur. Bu olaylardan sık kullanılanlar: TextChange, Enter, Leave, Validated.

Button Kontrolü

Windows form uygulamalarının hemen hemen tamamında yer alan Button kontrolü herhangi bir işlemi

veya komutu çalıştırmak amacıyla kullanılmaktadır. Button kontrolünü form üzerine yerleştirdikten sonra yazı tipi, boyutu, rengi gibi tasarım seçeneklerini özellikler paneli üzerinden düzenleyebiliriz. Ancak programın çalışma anında Button kontrolüne tıklanmasıyla birlikte yapılacak işlemleri kod editöründe ayarlamamız gerekmektedir.

RadioButton Kontrolü

RadioButton kontrolü birden fazla seçenek arasından sadece birini seçmeniz gereken bir durum olduğu zaman kullanılmaktadır. Form üzerine beş adet RadioButton eklediğimizde bunlardan biri seçili konumdayken başka birini işaretlediğinizde önceden seçili olan RadioButton kontrolüne ait seçim işleminin kalktığını gözlemleyebilirsiniz. Bu kontrol genellikle anlam yönünden birbiriyle zıt olan ilişkilerde kullanılmaktadır.

CheckBox Kontrolü

CheckBox kontrolü kullanıcıya belirli seçenekler arasından bir veya daha fazla seçim yapabilmesini sağlamaktadır. RadioButton kontrolünde kullanıcının yaptığı bir seçim diğer kontrolü etkilerken, CheckBox kontrolünde yapılan işlemler birbirinden bağımsız bir şekilde çalışmaktadır.

Form üzerine eklenen CheckBox kontrolünün seçili olma durumu Checked özelliği ile kontrol edilir. Bu özellik True değerini aldığı anda kontrol işaretlenmiş, False değerini aldığı anda kontrol işaretlenmemiş anlamı ortaya çıkar.

CheckedListBox Kontrolü

CheckedListBox kontrolü, CheckBox elemanlarını ListBox içerisinde görüntülenmesini sağlamaktadır. Yani her bir öğenin sol kısmında onay kutusunun bulunduğu bir listeleme şeklidir. CheckedListBox kontrolüne eleman eklemek için, kontrolün sağ üst kısmında bulunan oka tıkladıktan sonra Öğeleri Düzenle... seçeneğini işaretlememiz gerekmektedir. Bu işlemin ardından karşımıza gelecek olan Dize Koleksiyonu Düzenleyicisi penceresinde her bir öğeyi satırlar halinde girip tamam butonuna tıklamamız gerekmektedir. Kontrole öğe ekleme işini yukarıda bahsedildiği üzere tasarım ekranından yapabileceğimiz gibi özellikler panelinden Items seçeneğiyle de yapabilmemiz mümkündür.

ListBox Kontrolü

ListBox kontrolü içerisindeki elemanları liste şeklinde görüntülemeyi sağlamaktadır. Bu kontrolde bulunan elemanlara index numaraları ile ulaşılabilir. Kontrole girilen ilk elemanın index numarası sıfır değerini alır. ListBox içerisine değer girebilmek için özellikler panelinde bulunan Items seçeneği kullanılabilir.

Çalışma anında kontrol üzerindeki öğelerin seçiminin farklı şekillerde yapılabilmesini sağlayabilirsiniz. Bu işlemi gerçekleştirmek için özellikler panelinde bulunan SelectionMode seçeneğini kullanmamız gerekmektedir. Bu özellik None, One, MultiSimple ve MultiExtended olmak üzere dört farklı seçenek barındırmaktadır.

ComboBox Kontrolü

ComboBox kontrolü, içinde bulunan elemanları açılabilen liste halinde görüntülemeyi sağlamaktadır. Bu kontrol içerisindeki elemanlara da ListBox ve CheckedListBox kontrollerinde olduğu gibi index numaraları ile ulaşılabilir.

ComboBox içinde bulunan elemanları görüntülemek için, çalışma anında ComboBox üzerine tıklamak gerekmektedir. Bu tıklama işlemi yapılmadan ComboBox içinde herhangi bir metin görüntülemek istiyorsak, özellikler panelinde bulunan Text alanını doldurmamız gerekmektedir.

Uygulama çalışırken kontrol içinde bulunan öğelerin sıralı bir şekilde görüntülenmesini sağlayabiliriz.

Özellikler panelinde bulunan Sorted seçeneğini True olarak ayarlasak bu işlemimiz gerçekleşmiş olacaktır.

KONTROLLERİN ARA YÜZ TASARIMI

Form üzerine yerleştirdiğimiz kontrolleri birbirlerine göre hizalarken ufak farkları gözden kaçırabiliriz. Benzer şekilde kontrollerin aynı boyutta olmasını istiyorsak fark edemeyeceğimiz düzeyde eşitsizlikler olabilir. Bu yüzden Düzen Araç Çubuğunu kullanmak işimizi kolaylaştıracaktır. Düzen araç çubuğu ekranda yoksa Görünüm menüsü altında bulunan Araç Çubukları açılır listesindeki Düzen seçeneğini işaretleyerek ekrana getirebiliriz. Grup halindeki kontrolleri bu araç çubuğunda bulunan butonlar yardımıyla sağa, sola, yukarı veya aşağı hizalayabiliriz. Aynı zamanda kontrollerin boyutlarının da aynı yükseklik veya genişliğe gelmesini sağlayabiliriz.



DERS ADI Görsel Programlama I
ÜNİTE ADI Windows Form Kontrolleri II
ÜNİTE NO 9
YAZAR Öğr. Gör. DAHA ORHAN

GELİŞMİŞ KONTROLLER

Windows tabanlı uygulamalar geliştirilirken programcının ihtiyaç duyduğu resim ekleme alanları, zamanlayıcı, tarih bilgisi gibi standart kontrol nesnelerine göre biraz daha gelişmiş kontroller söz konusu olabilir. Geliştirilen uygulamaların tamamında gerekli olmasa bile projemizin niteliğine bağlı olarak bu kontrollerden bir veya daha fazlası istenildiği zaman formumuza eklenebilmektedir.

PictureBox Kontrolü

Uygulamalarımızın tasarımını gerçekleştirirken resimler ekleyerek daha canlı bir arayüz oluşturmak isteyebiliriz. Böyle bir durumda araç kutusunda bulunan PictureBox kontrolünü kullanmak işimize yarayacaktır. Bu kontrol sayesinde form üzerine resimler ekleyebilir ve bu resimlerin biçim özelliklerini ayarlayabiliriz.

PictureBox kontrolünü formumuz üzerine sürükleyip bırakmak yerine, araç kutusundan seçtikten sonra formda istediğimiz alana fare yardımıyla tarayarak boyutlandırmak resimleri için ayırdığımız bölgeyi ayarlamak için daha elverişli bir yöntemdir. Ayrıca kontrolün form üzerindeki konumunu ayarlamak için özellikler panelinde bulunan location seçeneğini kullanabiliriz. Bu seçenekte bulunan ilk sayı kontrolün formun sağ kısmına göre piksel cinsinden uzaklığını, ikinci sayı ise formun üst kısmına göre piksel cinsinden uzaklığını ifade etmektedir. Bu yöntemlerin haricinde PictureBox kontrolünü form üzerinde konumlandırmak için özellikler panelinde bulunan Dock seçeneğindeki 6 farklı yöntemden birini tercih etmemiz de mümkündür.

WebBrowser Kontrolü

Uygulamalarımızda bazı durumlar için internet sayfalarının görüntülenmesi isteyebiliriz. Böyle bir durumda araç kutusunda bulunan WebBrowser kontrolünü formumuza eklememiz gerekmektedir. Bu kontrolün davranış biçimi bir tarayıcınınkinden farklı değildir. WebBrowser kontrolü sayesinde formumuzda belirlenen alan içerisinde gitmesini istediğimiz adrese ait sayfayı görüntüleyebiliriz.

Timer Kontrolü

Timer kontrolü zamanla yapılması gereken işlemlerin olduğu veya ekranda zaman bilgisinin bulunması gibi durumlarda kullanılmaktadır. Kontrolde zamanın ilerleme durumu milisaniye cinsinden hesaplanır. Uygulamamıza Timer kontrolünü eklemek istediğimiz zaman, kontrol form üzerinde görüntülenmez. Timer kontrolünün simgesi tasarım ekranındayken uygulamanın arkaplanında çalışmak üzere formun alt tarafında açılan bir panel üzerine konumlanır. Bu panel üzerindeki simgeye çift tıklayarak kod editöründe kontrolün en sık kullanılan Tick olayı açılır. Bu olay içerisinde interval özelliğine bağlı olarak zamanlanan olaylar gerçekleştirilir.

ProgressBar Kontrolü

Windows formlarla çalışırken önemli kontrollerden biri olan ProgresBar, bir işlemin gerçekleştirilme esnasında ilerleme durumu hakkında bilgi vermeyi amaçlamaktadır. Bu ilerleme durumu bazen bir programın kurulumunu bazen de bir web sayfasının yüklenmesini gösterebilir. Projemizin özelliklerine bağlı olarak bu kontrolü ilişkilendireceğimiz seçenekler farklılaşabilir.

Özellikler panelinde bulunan Value değeri ProgressBar kontrolünün ne kadarlık bir kısmını boyalı olduğunu orantısız olarak ifade etmek için kullanılır. Bu özelliğin alabileceği değer aralığı yine özellikler panelinde bulunan Maximum ve Minimum özellikleriyle belirlenebilir. Varsayılan değer olarak 0 ile 100 arasında olacak şekilde ayarlanmıştır. ProgressBar kontrolünün minimum değerden maximum değere kadar hangi adım miktarıyla ilerleneceğini Step özelliği ile ayarlayabiliriz.

GroupBox Kontrolü

GroupBox kontrolü, uygulamalarımızda form üzerine yerleştirilen kontrolleri belirli gruplar halinde görüntülemek için kullanılmaktadır. Tasarım ekranında ilgili kontrolleri bir bütün olarak yansıtır ön plana çıkarabileceği gibi genellikle birbirinden etkilenen RadioButton gibi kontrollerin de ayrılmasına olanak sağlamaktadır.

Form üzerine yerleştirilen GroupBox kontrolü, başlık yazısı ve çerçeveye sahiptir. Başlık yazısı ile grup içinde bulunan kontroller hakkın bilgi verilebilir. Özellikler panelinde bulunan Text seçeneği ile

kontrole ait başlık yazısını düzenleme imkanına sahibiz.

ImageList Kontrolü

ImageList kontrolü projemizde kullanacağımız resim dosyalarının toplu olarak bir alan üzerinde tutulmasını ve tutulan bu resimlere diğer kontroller tarafından erişilmesini sağlamak amacıyla kullanılmaktadır. Araç kutusu üzerinde kontrole çift tıkladığımız zaman kontrol formumuz üzerinde görüntülenmeyecektir. Şekilde gösterildiği gibi ImageList kontrolüne ait simge, kullanıcı arabiriminin altında bulunan panel üzerine yerleşecektir.

ImageList kontrolünün içerisine resim ekleme, kontrolde var olan resimleri silme ve düzenleme işlemlerini gerçekleştirebiliriz. Bu işlemler özellikler panelinde bulunan Image seçeneğinin yanındaki butona tıklanarak açılan Görüntüler Koleksiyonu Düzenleyicisi penceresinden gerçekleştirilebilir.

DateTimePicker Kontrolü

DateTimePicker kontrolü, tarih ve saat bilgisi girilirken kullanılmaktadır. Bu kontrol nesnesini formumuza ekleyip çalıştırdığımız zaman sistem tarihi başlangıç değeri olarak görüntülenir. Çalışma anında form üzerinde bulunan DateTimePicker kontrolünü fare ile seçtikten sonra ok tuşlarını kullanarak görüntülenen tarih bilgisi seçim olacak şekilde değiştirilebilmektedir.

Çalışma anında kontrolün görüntüleyebileceği tarih aralığı sınırlandırılabilir. Bu işlem için özellikler panelinde bulunan MaxDate ve MinDate seçeneklerini kullanabiliriz.

MonthCalendar Kontrolü

MonthCalendar kontrolü, önemli tarihleri veya zaman dilimlerini işaretleyebilmemizi sağlamaktadır.

DateTimePicker kontrolüne benzer özellikleri bulunsu da çalışma anında açılır bir liste yerine tüm bir takvim halinde görüntülenmektedir. MonthCalendar kontrolü ve DateTimePicker kontrolü ile çalışırken DateTime sınıfına hakim olmamız gerekmektedir.

Sistem tarihini kullanmak için DateTime yapısının Today özelliğinden istifade edilebilir.

Sistem saatine ait tarih bilgisiyle beraber saat bilgisini de kullanabilmek için DateTime yapısına ait Now özelliğini kullanmamız gerekir.

DaysInMonth() metodu integer tipinde değer üretir. 2 parametre alacak şekilde kullanılmaktadır:

DateTime.DaysInMonth(YIL, AY); Burada metoda ait parantezler içinde girilecek olan ilk parametrede, gün sayısını öğrenmek istediğimiz Ay'a ait Yıl bilgisi olmalıdır. Metodun ikinci parametresinde ise gün sayısını öğrenmek istediğimiz Ay bilgisi yazılmalıdır.

IsLeapYear() metodu ise biraz daha genel bir kullanıma sahiptir. Yani herhangi bir yıla ait ayların kaç gün geçtiğine dair bir bilgiyi elde edemeyiz. Sadece belirtilen yılın gün sayısı hakkında bir yorum yapmamızı ve dolaylı olarak şubat ayı hakkında fikir sahibi olmamızı sağlamaktadır.

RichTextBox Kontrolü

Windows form uygulamalarında kullanıcının metin girişi yapabildiği TextBox kontrolü bazı durumlarda yetersiz kalabilmektedir. Örneğin biçimlendirme özellikleri TextBox kontrolünde kullanılamamaktadır.

Kullanıcının veri girişi yaptığı alanlarda daha gelişmiş seçenekler sunmak için RichTextBox kontrolü kullanılabilir. Bu kontrol sayesinde yazıların font özellikleri ve rengi değiştirilebilmektedir. Aynı zamanda metnin farklı yerlerinde farklı seçeneklerin uygulanması da sağlanabilmektedir..

RichTextBox kontrolü ile kullanıcının metin girebilmesine izin verildiği gibi daha önceden hazırlanan bir metin de kullanıcıya gösterilebilmektedir. Hazır bir metni RichTextBox kontrolünün içerisine ekleyebilmek için özellikler alanında bulunan Text seçeneğini kullanabiliriz. Aynı işlem kontrolün sağ köşesinde bulunan oka tıklanıp Metin Satırlarını Düzenle... seçeneğine tıklanarak da gerçekleştirilebilir. Açılan Dize Koleksiyonu Düzenleyicisi penceresine istenilen metinler yazılarak Tamam butonuna tıkla işlemi gerçekleştirilir. Böylelikle hazır bir metin çalışma anında kullanıcıya gösterilmiş olacaktır. Kullanıcı kendi yazdığı metinleri düzenleyebildiği gibi hazır sunulan bu metin üzerinde de işlemler gerçekleştirebilir.



DERS ADI Görsel Programlama I
ÜNİTE ADI Windows Form Kontrolleri III
ÜNİTE NO 10
YAZAR Öğr. Gör. DAHA ORHAN

İLERİ DÜZEY KONTROLLER

Visual Studio 2019 programında araç kutusunda benzer özellikleri taşıyan kontroller sekmeler halinde sunulmuştur. Bununla beraber ilk sekme olan Tüm Windows Formları seçeneğinde bütün kontroller listelenmiştir.

ListView Kontrolü

Listesini oluşturmak istediğimiz elemanların ListBox kontrolünde görünmesini sağlayabiliriz. Ancak bu kontrolün yetersiz olduğu durumlar olabilir. Örneğin elemanları sadece satırlar halinde değil sütunlar halinde de listelemek isteyebiliriz. Bu durumda ListView kontrolünü kullanmak gerekecektir. Aynı zamanda ListView kontrolü elemanların simge olarak listelenmesini sağlamaktadır.

Bu kontrole çalışma anında gerçekleştirilen bir etkileşim ile eleman ekleme imkanına da sahibiz. Bu işlemi gerçekleştirmek için Items.Add() metodunu kullanmamız gerekmektedir. Ayrıca çalışma anında kontrol üzerinde seçilen bir elemanın Text özelliğini kullanmak isteyebiliriz. String veri tipinde görüntülenen bu değere kontrolün SelectedItems özelliği ile ulaşılabilir.

TreeView Kontrolü

Form üzerindeki görüntüsü ListBox kontrolüne benzer şekilde olan TreeView kontrolünü, içindeki bilgilerimizi hiyerarşik bir yapıda görüntüleyerek kullanıcının seçimine sunmak için kullanabiliriz. Kontrolünün içine değer girebilmek için özellikler panelinde bulunan Nodes seçeneğine tıklamamız gerekmektedir. Aynı pencereye tasarım ekranındayken kontrolün sağ üst köşesinde bulunan ok'a tıkladıktan sonra açılan alan üzerinde Düğmeleri Düzenle... seçeneğiyle de ulaşmamış mümkündür. Bu işlemin ardından TreeNode Düzenleyicisi penceresi açılır. Bu pencerede bulunan Kök Ekle butonu ile kontrolümüzde görüntülemek istediğimiz ana seçenekleri oluşturabiliriz. Girilen her ana kök için seçim işlemi yapıldıktan sonra Alt Öğe Ekle butonuna tıklanarak alt seçenekleri belirlenebilir. Eklediğimiz kök ve alt öğelerin isimlerini, pencerenin sağ tarafında bulunan özellikler alanındaki Text seçeneğinden düzenleyebiliriz. Ayrıca kökler eklenirken sıralamanın veya hiyerarşinin yanlış yapıldığı durumlarda ilgili kök veya alt öğe seçildikten sonra TreeNode Düzenleyicisi penceresinde bulunan yukarı veya aşağı yönlü ok işaretlerinin bulunduğu butonlar yardımıyla düzenleme yapabilirsiniz.

TabControl Kontrolü

Windows form uygulamalarında çoklu formlarla çalışılabilir. Farklı formlar açarak yapacağımız işlemleri tek bir form üzerinden de gerçekleştirebiliriz. Yani form alanımız içerisinde sekmelerin olması ve bu sekmeler yardımıyla farklı veya aynı kontrolleri formumuza yerleştirebiliriz. Böyle bir durumu gerçekleştirebilmek için TabControl kontrolünü kullanmamız gerekmektedir.

Kontrole yeni sekmeler eklemek için özellikler panelinde bulunan TabPages seçeneğine tıklamamız gerekmektedir. Bu işlemin ardından tabPage Koleksiyon Düzenleyici penceresi açılacaktır. Kontrolde var olan sekmeler pencerenin sol tarafındaki Üyeler alanında görüntülenir. Pencerede bulunan Ekle butonuna tıklayarak yeni sekmeler ekleyebilir, Kaldır butonuna tıklayarak seçili olan sekme kaldırılabilir. Sekmelerin isimlerini düzenleyebilmek için tabPage Koleksiyon Düzenleyici penceresinde sağ tarafta bulunan özellikler alanındaki Text seçeneği kullanılmaktadır.

NotifyIcon Kontrolü

Hazırladığımız uygulamaya ait belirlediğimiz bir simgenin görev çubuğunda sistem saatin olduğu bölgede görüntülenmesini sağlayabiliriz. Forma eklemiş olduğumuz NotifyIcon kontrolü ile bu işlem gerçekleştirilmektedir. Eklediğimiz kontrol form üzerinde görüntülenmez. Formun alt tarafında açılan panel üzerinden kontrole erişebiliriz.

NotifyIcon kontrolüne ait bildirim ayarları kontrolün gizli olmadığı durumlarda yani visible özelliğinin True olduğu durumlarda ortaya çıkmaktadır. Bununla birlikte bildirim alanının ekranda ne kadar kalacağını da belirtmemiz gerekmektedir. Bu işlem ilk parametresi zorunlu olan ve dört parametre alabilen ShowBalloonTip(); özelliği ile gerçekleştirilmektedir. Özellikler panelinde bulunmayan bu seçeneğin parametreleri kod editöründen gerçekleştirilebilir. Zorunlu olan ilk parametresine, bildirimin kaç saniye ekranda görüneceğini milisaniye cinsinden yazmamız gerekmektedir.

MenuStrip Kontrolü

MenuStrip kontrolünü uygulamamıza eklediğimiz zaman hem formun üst kısmında hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır. Formun içine eklenen MenuStrip kontrolündeki ilgili alanları doldurarak tasarım özellikleri hızlı bir şekilde ayarlayabiliriz. Menüde bulunmasını istediğimiz ilk seçeneği Buraya Yazın kısmına tıklayarak doldurabiliriz. Bu işlemin ardından yeni menü seçeneklerini girmek için sağ tarafta bulunan kutulara menü isimlerini yazarak menülerinizi oluşturabilirsiniz.

Menü komutlarını girmek için seçilen menünün altında bulunan Buraya Yazın seçeneği ile bulunmasını istediğiniz komutları yazabilirsiniz. Benzer özellikler taşıyan komutların gruplandığını göstermek amacıyla komutlar arasında çizgiler konulabilir. Bu işlem, komutlar yazıldıktan sonra önceki komutlardan ayrı özellikler taşıdığı düşünülen komuta fare ile sağ tıklama işlemi yapıp açılan menüden Ekle>Seperator adımları takip edilerek gerçekleştirilmektedir.

Birçok uygulamada ortak bir şekilde kullanılan Dosya, Düzen gibi menüler bulunmaktadır. Bu menüleri bahsedilen yöntemlerle tek tek hazırlamak yerine Form üzerine eklenen MenuStrip kontrolünün sağ üst köşesinde bulunan ok'a tıklayıp Standart Öğe Ekle seçeneği işaretlenerek bunları formumuza ekleyebiliriz.

ContextMenuStrip Kontrolü

ContextMenuStrip kontrolü, hazırladığımız uygulamanın çalışma anında form üzerinde bulunan herhangi bir kontrol üzerinde fare ile sağ tuşu ile açılan bir menü hazırlamamızı sağlamaktadır. Böylelikle kontrol üzerinde sağ tuş ile açılan menüde hazırlanan komutlardan herhangi birini kullanarak kullanıcıya yaptırmak istediğimiz seçenekleri belirleyebiliriz.

ContextMenuStrip kontrolü ile hazırlanan açılan menü, ilgili kontrolün özellikler panelinde bulunan ContextMenuStrip seçeneği ile bağlanabilmektedir. Bu bağlantı işlemi gerçekleştirildikten sonra uygulamayı çalıştırdığımız zaman kontrol üzerinde fare ile sağ tuşa tıklayarak hazırladığımız menüyü görüntüleyebiliriz.

ToolStrip Kontrolü

Araç çubuğunun üzerine birçok uygulamada sıklıkla kullanılan seçenekleri hazır olarak ekleyebilirsiniz.

ToolStrip kontrolünün sağ üst köşesindeki ok'a tıkladığımız zaman açılan menüden Standart Öğe Ekle seçeneğini kullanmanız gerekmektedir. Bu işlemin ardından Yeni, Aç, Kaydet, Yazdır, Kes, Kopyala, Yapıştır ve Yardım olmak üzere 8 adet kullanıma hazır buton otomatik olarak eklenecektir. Bu seçeneklerden istediğinizi Delete tuşunu kullanarak silebilir, sürükleyip bırak yöntemi ile yerini değiştirebilir veya ToolStrip kontrolünde bulunan düğme ile yeni öğeler ekleyerek düzenleme işleminizi gerçekleştirebilirsiniz.

Özellikle butonlardan oluşan araç kutularının görselliğini tamamlayabilmek için kontroller üzerine resimler eklenebilmektedir. Resim eklenilmek istenen kontrol seçildikten sonra özellikler panelinde bulunan Image seçeneği kullanılmalıdır. Açılan Kaynak Seç penceresinde Yerel kaynak veya Proje kaynak dosyası seçeneklerinden her hangi biri tercih edilerek resim yükleme işlemi gerçekleştirilir ve Tamam butonuna tıklanır.

StatusStrip Kontrolü

Uygulamanın çalışma anında kullanıcıya bilgi vermek amacıyla StatusStrip kontrolü kullanılabilir. Bu kontrol form üzerine eklendiği zaman hem formun alt tarafında hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır.

StatusStrip kontrolü üzerinde anlık olarak ilerleyen durumun bilgisi, StatusLabel kontrolü eklenerek kullanıcıya gösterilebilmektedir. Ayrıca çalışma daha önceden durum çubuğunda gösterimi sağlanan seçeneklerin bilgisi de DropDownButton kontrolü kullanılarak listelenebilmektedir.



DERS ADI Görsel Programlama I
ÜNİTE ADI Diyalog ve Mesaj Pencereleeri
ÜNİTE NO 11
YAZAR Öğr. Gör. DAHA ORHAN

MESSAGEBOX

Uygulamalar çalışırken üretilen sonuçların ve bilgilerin kullanıcıya ve programcıya gösterilmesi amacıyla MessageBox penceresi kullanılmaktadır. Bu pencere yardımıyla kullanıcıya gösterilecek olan metnin uzunluğu en fazla 1024 karakter olabilmektedir. Bu pencerede zorunlu olan ilk parametre yani mesaj bilgisi String veri türünde olmalıdır. Kullanıcıya bilgi vermek için program akışı içerisinde MessageBox sınıfına ait Show metodunu kullanmamız gerekmektedir.

MessageBox pencereleeri birden fazla butona sahip olduğunda varsayılan değer olarak ilk buton aktif konumda bulunur. Enter tuşuna bastığımız zaman aktif olan butonun işlevi gerçekleştirilir.

Uygulamalarımızda aktif olan butonun farklı bir buton olmasını ayarlayabiliriz. Bunun için DefaultButton kodunu kullanmamız gerekmektedir. Üç butonun olduğu bir pencerede ilk buton için DefaultButton1, ikinci buton için DefaultButton2, üçüncü buton için de DefaultButton3 kodunu yazmamız gerekir.

Bazı durumlarda kullanıcının görmesini istediğimiz metni, satırlar halinde sunmak isteyebiliriz. Bunun için metin bilgisine yer verdiğimiz parametre kısmında yeni satır anlamına gelen NewLine ifadesi veya “\n” karakteri kullanılmalıdır.

DİYALOG PENCERELERİ

Diyalog pencereleeri kullanıcının etkileşime girmesini sağlayan ve açılan pencereyi yanıtlamasını bekleyen yapıdaki grafiksel denetim öğeleridir. Windows form uygulamalarına ekleyebilmek için araç kutusunda bulunan İletişim Kutuları sekmesi kullanılabilir. Bu alanda yer alan bütün diyalog pencereleeri ShowDialog metodu kullanılarak görüntülenmektedir. Aynı zamanda kullanıcıdan gelen yanıt yani basılan tuş DialogResult özelliği kullanılarak alınmaktadır.

ColorDialog

Color Dialog kontrolü kullanıcının renk seçme işlemini gerçekleştirmesi için kullanılan pencerenin açılmasını sağlamaktadır. Renk başlığı adı altında açılan pencere üzerinden temel renklerden biri seçilebileceği gibi Özel Renk Tanımla seçeneği de kullanılabilir. Özel renk tanımla seçeneği ile açılan alan üzerinde fare imleci ile herhangi renk seçimi yapılabilmektedir. Ek olarak seçilecek olan renge ait Kırmızı, Yeşil ve Mavi kodları manuel olarak da girilebilmektedir. Eğer belirlen bu renk sıklıkla kullanılıyorsa Özel Renklere Ekle butonu kullanılarak Özel Renkler arasına eklenebilir.

Çalışma anında ColorDialog penceresi özelleştirilebildiği gibi varsayılan ayarlarına geri dönülmesi de sağlanabilmektedir. Bu işlem için kontrol Reset metodunu kullanmak gerekir.

FolderBrowserDialog

Bilgisayarda bulunan bir klasörü seçmek için FolderBrowserDialog kontrolünü kullanmak gerekmektedir. Bu kontrol sayesinde bilgisayarda seçilen bir klasörün yolunu özellikler panelinde bulunan SelectedPath seçeneğinde tutabiliriz. FolderBrowserDialog uygulamaya eklendikten sonra SelectedPath özelliğine tıklayarak Klasöre Gözet penceresi açılmaktadır.

Klasöre Gözet penceresi bilgisayarda bulunan klasörleri hiyerarşik bir yapı içinde göstermektedir. Bu alan üzerinde klasörlerin yanındaki oklara tıklanarak alt klasörler görüntülenip gizlenebilmektedir. Aynı zamanda bir klasör seçildikten sonra Yeni Klasör Oluştur butonuna tıklanarak ilgili klasör altında yeni klasörler oluşturulabilmektedir. Eğer kullanıcının böyle bir yetkiye sahip olmasını istemiyorsanız, özellikler panelinde bulunan ShowNewFolderButton seçeneğine False değerini atamanız gerekmektedir. Yapılan işlemleri onaylamak için Tamam, gerçekleştirmeden çıkmak için İptal butonu kullanılmalıdır.

FontDialog

Araç kutusu üzerinde bulunan FontDialog kontrolü sayesinde uygulamamızdaki metinlerin yazı tipi özelliklerini belirleyebilir ve değiştirebiliriz. Kontrolü formumuza ekledikten sonra özellikler panelinde bulunan Font seçeneğine tıklayarak Yazı Tipi penceresini açabiliriz.

Yazı tipi penceresinde metinler üzerinde değişiklik yapabileceğimiz Yazı Tipi, Yazı tipi stili, Boyut, Efektler, Örnek ve Yazı alanları bulunmaktadır. Bu alanlar sistemde kurulu olan seçenekleri içermektedir. Gerçekleştirilen değişikliklerin ön izlemesi Örnek alanında görüntülenmektedir. Yapılan

bu deęişikliklerin uygun olduęunu düşünüyorsanız Tamam butonuna tıklayarak onaylayabilir veya deęişikliklerin etkin hale gelmemesi için İptal butonuna tıklayarak formunuza dönebilirsiniz.

OpenFileDialog

OpenFileDialog kontrolü, kullanıcının bilgisayarda kayıtlı olan dosyaları açması için kullanılan pencerenin ekranda görüntülenmesini sağlamaktadır. Çalışma anında pencere başlığı varsayılan deęer olarak “Aç” adıyla karşımıza gelmektedir. Pencere başlığını deęiştirmek için özellikler panelinde bulunan Text alanını düzenlememiz gerekir. Aynı zamanda bu pencerede bulunan Dosya adı bölümüne atanmak istenen deęer, özellikler panelindeki FileName seçeneęi ile ayarlanabilmektedir.

ShowDialog metodu kullanılarak ekrana getirilen dosya açma penceresinde listelenecek dosya türlerinin sınırlı olması sağlanabilmektedir. Bu işlem için Filter özelliğini kullanmak gerekir. Filtreleme işleminde öncelikle görüntülenecek dosya türüne ait bir açıklama metni, sonra dikey çubuk, daha sonrada dosya uzantısı belirtilmelidir. Bu yöntem kullanıldığı zaman tek bir dosya türüne ait dosyalar listelenmektedir. Listede farklı dosya türlerinin de görüntülenmesi, dikey çubuk konularak yapılabilmektedir. Bu şekilde yapılacak olan filtreleme işlemi özellikler panelinden ayarlanabileceęi gibi kod editöründe de gerçekleştirilmektedir.

SaveFileDialog

SaveFileDialog kontrolü sayesinde kullanıcının herhangi bir dosyayı seçerek kaydetmesini sağlayan bir pencere açılmaktadır. ShowDilaog metodu kullanılarak açılan Farklı Kaydet penceresi ile var olan bir dosyanın açılması ve var olan bir dosyanın üzerine yeni bir dosya yazılması veya yeni bir dosya oluşturulması sağlanabilir. Aynı zamanda pencere içinde bulunan Yeni Klasör butonuna tıklayarak ilgili kök klasörünün içinde yeni klasörler oluşturulabilir.

Kullanıcının farklı kaydet penceresini açtıktan sonra var olan bir dosya adını belirterek kaydet butonuna tıkladığı zaman bir uyarı ile karşılaşması sağlanabilir. Bu işlem için özellikler panelinde bulunan OverwritePrompt özelliğini True olarak ayarlaması gerekmektedir.

INPUTBOX

Kullanıcı tarafından bilgi girişini sağlayabileceğimiz bir dięer pencere de InputBox penceresidir. Ancak C# içinde InputBox penceresinin bulunmadığı belirtilmişti. Bu pencereyi projelerimizde kullanabilmek için öncelikle referans olarak Microsoft.VisualBasic sınıfını eklememiz gerekmektedir. Projemize ekleyebileceğimiz referansların olduęu pencereye Proje menüsü altında bulunan Başvuru Ekle... seçeneęi ile ulaşabiliriz.

Başvuru yöneticisi penceresinin sol tarafında bulunan Derlemeler ve Framework hiyerarşisi takip edilip, açılan listeden Microsoft.VisualBasic referansı işaretli konuma getirilmelidir. Tamam butonuna tıklayarak eklemiş olduęumuz referansları Çözüm Gezgini panelinde bulunan Başvurular başlığı altında görebiliriz.

InputBox Kullanımı ve Parametreleri

Veri girişini sağlamak amacıyla kullanılan bu pencere genellikle aldığı deęeri bir deęişken üzerinde saklamaktadır. Kullanım yöntemi beş farklı parametre alacak şekildedir. Bu parametrelerden ilki zorunlu olup dięerlerinin kullanımı tercihe baęlıdır.

Mesaj metni

String veri türünde deęer alabilen ve InputBox penceresinin zorunlu olan parametresidir. Kullanıcının ilgili alana gireceęi bilgi hakkında yönlendirme işlemini gerçekleştirmek amacıyla kullanılmaktadır. Eęer bu alana gireceğiniz deęer bir deęişkenin içinden alınmıyorsa, yazacaęınız metni Çift Tırnak (“”) içine alarak yazmanız gerekmektedir. Bununla beraber ilk parametrenin deęerini, farklı türde bir deęişkendten alabilmek için dönüştürme (convert) işlemi uygulanmalıdır.

Pencere başlığı

Windows form uygulamalarında çalışılan pencerenin amacını veya özelliğini belirtmek için başlık belirleme işlemi sıklıkla kullanılmaktadır. InputBox penceresinin başlığını ayarlayabilmek için de, kullanım şeklinde gösterilen ikinci parametrenin doldurulması gerekmektedir. Bu parametre de ilk parametrede olduęu gibi String türünde deęer almaktadır.

Varsayılan deęer

Genellikle doldurulacak alan hakkında kullanıcıya örnek sunmak için kullanılmaktadır. İlk iki parametrede olduęu gibi String türünde deęer almalıdır. Bu alan doldurularak çalıştırılan uygulamada InputBox penceresi bu bilgiyi kullanıcıya seçilmiş halde göstermektedir. Bunun sebebi girilen deęerin hemen üzerine yazılmasını sağlamak içindir.

X koordinatörü

InputBox penceresinin ekran üzerindeki konumlanacaęı alan belirlenebilir. Bunun için dördüncü parametre, InputBox penceresinin ekranın soluna göre konumlanacaęı alanı ayarlamamızı sağlamaktadır. Buraya girilecek olan deęer sayısal olmakla beraber mesafe piksel cinsinden ayarlanmaktadır.

Y koordinatörü

InputBox penceresinin ekranın soluna göre konum bilgisi belirlenebildiğı gibi ekranın üst kısmına göre hangi konumda açılacaęı da belirlenebilmektedir. Bu işlem için InputBox kullanım şeklinde gösterilen

beşinci yani son parametre kullanılmalıdır. Dördüncü parametrenin kullanım şekli ile aynıdır. Bu iki parametre boş bırakıldığı takdirde pencere varsayılan bir konum üzerinde açılacaktır.



DERS ADI Görsel Programlama I
ÜNİTE ADI Dosya ve Klasör İşlemleri
ÜNİTE NO 12
YAZAR Öğr. Gör. İSHAK METEHAN SİS

DOSYA ve KLASÖR İŞLEMLERİ

Sabit disk, CD gibi belleklere kaydedilen veri ya da bilgiler “dosya” olarak adlandırılır. Verilerin organizasyonu sağlamak, dosyaların karışmasını önlemek, aranan bilgiye daha çabuk ulaşabilmek için klasör denilen yapılardan faydalanılır. Örneğin bilgisayardaki her bir resim dosya, bu resimlerin bir arada tutulduğu yapı ise klasör olarak adlandırılır.

Dosya veya klasör işlemleri programlama dilleri ile yapılabilir. Programlama dillerinde yer alan girdi çıktı işlemlerine ingilizcede Input/Output kelimesinin kısaltması olan I/O işlemleri denir. Klasör ve dosyalama işlemleri de bir girdi çıktı işlemi olduğu için C# dilinde girdi çıktı işlemleri için kullanılan System.IO kütüphanesi kullanılır. Klasör işlemleri için System.IO kütüphanesinde de yer alan Directory kullanılırken, dosya işlemleri için bu kütüphane içinde yer alan File sınıfı kullanılmaktadır.

Klasör İşlemleri

Görsel programlamada klasör işlemleri System.IO namespace bulunan Directory sınıfı ile yapılmaktadır.

Klasör Oluşturma

Klasör oluşturmak için Directory sınıfının “CreateDirectory” metodu kullanılır.

Klasör Silme

Klasör silmek için için Directory sınıfı ile birlikte “Delete” metodu kullanılır. Eğer klasör boş ise aşağıdaki örnekte de görüldüğü gibi Delete metodu ile klasör kolayca silinebilir. Dosya boş ya da dolu olması gözetilmeksizin silinmek istiyorsa silinecek klasör yolu belirtildikten sonra true komutunun eklenmesi gerekir.

Klasör Taşıma

Klasör ve dosya taşıma işlemi için “Move” metodu kullanılır. Burada dikkat edilmesi gereken nokta klasör taşınırken kaynak ve hedef klasörün aynı dizinde olması gerekir. Dosya taşıma işlemi için böyle bir kural geçerli değildir.

Klasör Varlığını Kontrol Etme

Herhangi bir klasörün belirtilen yolda olup olmadığı “Exists” metodu ile kontrol edilir. Exists metodu geriye true ya da false değeri döndürür.

Klasör Varlığını Kontrol Etme

Klasör oluşturulma ya da değiştirilme tarihi gibi klasör üzerinde yapılan işlemler hakkında bilgiye ihtiyaç duyulabilir. Bu durumda klasör kodları ile zaman kodları birlikte kullanılarak ihtiyaç duyulan bilgiler elde edilebilir.

Klasör Listeleme

Belirtilen adresteki tüm klasörlerin isimlerini elde etmek için GetDirectory metodu kullanılır.

Dosya İşlemleri

Görsel programlamada dosya işlemleri System.IO isim uzayında bulunan File sınıfı ile yapılmaktadır.

Dosya Oluşturma

Dosya oluşturmak için File sınıfının “Create” metodu kullanılır.

Dosya Silme

Görsel programlamada dosya işlemleri için File sınıfının “Delete” metodu kullanılmaktadır. Silinen dosyalar geri dönüşüm kutusuna gönderilmez.

Dosya Kopyalama

Dosya kopyalama işlemi için File sınıfının “Copy” metodu kullanılır.

Dosya Taşıma

Dosya taşıma işlemi için File sınıfının “Move” metodu kullanılır.

Dosya Varlığını Kontrol Etme

Herhangi bir dosyanın olup olmadığı “Exists” metodu ile kontrol edilir. Exists metodu geriye true ya da false değeri döndürür.

Dosya Bilgilerini Elde Etme

Var olan bir dosyanın oluşturulma tarihi, değiştirilme tarihi ya da boyutu gibi bilgiler elde edilebilir. Daha önce ki bölümde klasör bilgilerini elde etmek için kullanılan `GetCreationTime()`, `GetLastAccessTime()`, `GetLastWriteTime()` gibi metotlara `File` sınıfı ile birlikte kullanıldığında dosyalar için de geçerli olur.

Dosya Okuma ve Yazma İşlemi

Dosya okuma ve yazma işlemleri için `FileStream`, `StreamReader` ve `StreamWriter` sınıfları kullanılmaktadır.

FileStream Sınıfı Ve Metotları

`FileStream` sınıfı dosya oluşturmak için kullanılır.

- `FileStream fS = new FileStream (string dosyaAdresi, FileMode dosya_modu);`

İlk parametre dosyanın adresini ikinci parametre ise dosyanın ne amaçla açıldığını göstermektedir.

`FileStream` sınıfı örnekte verildiği gibi sadece 2 parametreden oluşmamaktadır. `FileAccess`, `FileShare`, `FileOptions` gibi farklı parametreler alabilir.

File Mode

Dosya açılacak mı yeniden mi oluşturulacak bu seçenekleri belirler. `FileMode.Open`, `FileMode.Append`, `FileMode.Create`, `FileMode.CreateNew`, `FileMode.OpenOrCreate`, `FileMode.Truncate` gibi seçenekleri bulunmaktadır.

FileAccess

Dosyaya nasıl erişileceğini belirler. `FileAccess.Read`, `FileAccess.Write`, `FileAccess.ReadWrite` gibi seçenekleri bulunmaktadır.

FileShare

`FileShare` ile dosyanın diğer stream'ler tarafından nasıl paylaşılacağı belirlenir. `FileShare.None`, `FileShare.Read`, `FileShare.Write`, `FileShare.ReadWrite` gibi seçenekleri bulunmaktadır.

FileOptions

`FileOptions` parametresi ile dosya seçenekleri belirlenir. `FileOptions.None`, `FileOptions.WriteThrough`, `FileOptions.Asynchronous`, `FileOptions.RandomAccess`, `FileOptions.DeleteOnClose`, `FileOptions.Encrypted`

`FileStream` sınıfının `BeginRead`, `BeginWrite`, `Seek`, `Flush`, `EndRead`, `EndWrite`, `Read`, `ReadByte`, `Write`, `WriteByte`, `Close`, `Finalize`, `Dispose`, `GetType`, `ToString`, `Unlock`, `SetLength` metotları yaygın olarak kullanılmaktadır.

Ayrıca `Length`, `Name`, `Position`, `CanWrite`, `CanRead`, `Handle`, `ReadTimeout`, `WriteTimeout`, `IsAsync`, `CanSeek` özellikleri de `FileStream` sınıfının yaygın kullanılan özelliklerindendir. Metotlara ve özelliklere ait detaylara ünite içinde yer verilmiştir.

StreamReader Sınıfı ve Metotları

`StreamReader` sınıfı `txt`, `html`, `xml` gibi uzantılı dosyalar içerisindeki metinleri okumak için kullanılır.

`StreamReader` sınıfını kullanabilmek için ilk olarak `FileSteram` tanımlanır.

`SreamReader` sınıfında `ReadLine`, `ReadToEnd`, `Peek`, `Read`, `GetType`, `ToString`, `Close` gibi metotlar bulunmaktadır.

StreamWriter Sınıfı ve Metotları

`StreamWriter` sınıfı ile dosyalara yazma işlemi yapılır. `SreamWriter` sınıfında `Flush`, `WriteLine`, `Write`, `Close`, `GetType`, `ToString` gibi metotlar yaygın olarak kullanılmaktadır.



DERS ADI Görsel Programlama I
ÜNİTE ADI Menü Tasarımı ve Yazdırma İşlemleri
ÜNİTE NO 13
YAZAR Öğr. Gör. İSHAK METEHAN SİS

MENÜ TASARIMI

Görsel Programlamada yer alan nesneler ile form üzerinde çeşitli özelliklere sahip menüler oluşturulabilir. Bunun için “MenuStrip”, “ContextMenuStrip”, “ToolStrip” ve “StatusStrip” olmak üzere 4 farklı nesne kullanılmaktadır. Farklı nesneler kullanılarak farklı özelliklere sahip menüler oluşturulabilir. Bu nesnelerin ortak noktası ise hepsi Strip yani Şerit nesnesidir ve menü üzerinde şerit görünümü verirler.

MenuStrip

MenuStrip birçok uygulamada kullanılan genelde üst kısımda yer alan ana menü ve altında ki alt menülerden oluşan bir yapıdır. Çoğu programda görülen üst menüler MenüStrip nesnesi ile yapılmıştır. Bir Notepad sayfası açıldığında üst tarafta görülen Dosya, Düzen, Görünüm gibi seçenekler MenuStrip’e örnek olarak verilebilir.

MenuStrip windows formuna toolboxtan sürükleyip bırak yöntemi ya da kod ile menuStrip oluşturarak eklenebilir. Bu bileşen üzerinde istenilen özellikler eklenerek özelleştirilmiş menüler oluşturulabilirken, erişim tuşları, kısa yol tuşları, onay işaretleri, resimler ve ayırıcı çubuklar ekleyerek menülerin kullanılabilirliği ve okunabilirliğini iyileştirebilir.

Özelleştirilmiş menüler tasarlamak için Checked, CheckOnClick, Enabled, ShortcutKeys, ShowShortcutKeys, ShortCutKeyDisplayString, ToolTipText, DisplayStyle, Items, TextAlign, TextDirection gibi özellikler kullanılmaktadır.

MenuStrip kontrolü projede kullanılacak ise ilk olarak form üzerine araç kutusundan MenuStrip eklenir.

Daha sonra menüde yer almasını istediğimiz seçenekler Buraya Yazın kısmına tıklanarak menü oluşturulur. Çoğu uygulamada Dosya, Düzen, Araçlar gibi seçenekler menülerde ortak olarak kullanılmaktadır. Bu tip ortak menüler MenüStrip kontrolünün sağ üst köşesinde yer alan kulakçığa tıklanıp Standart Öğe Ekle seçeneği seçilerek kolayca oluşturulabilir. MenuStripTask menüsü ekleme penceresinde RenderMode, Dock ve GripStyle olmak üzere üç farklı seçenek bulunmaktadır.

ContextMenuStrip

Herhangi bir kontrole farenin sağ tuşu ile tıklanıldığında açılan menüler oluşturmak için ContextMenuStrip kontrolü kullanılır. Böyle bir menü oluşturmak için ilk olarak araç kutusundan ContextMenuStrip seçilerek form üzerine eklenir. Daha sonra MenuStrip’te olduğu gibi Buraya Yazın kısmına, kullanıcıya gösterilmek istenen seçenekler yazılır. Uygulama çalıştırılıp sağ tuşa tıklanınca menünün açılması için oluşturulan ContextMenuStrip kontrolünün bağlantısının yapılması gerekir.

ToolStrip

ToolStrip, genelde uygulamalarda üst kısımda yer alan araç çubuğunu oluşturmak için kullanılan bir kontroldür. ToolStrip form üzerine eklendikten sonra Özellikler penceresinin Items kısmından araç çubuğunda olması istenilen öğeler seçilerek ekleme yapılır.

MenuStrip kontrolünden farklı olarak burada ToolStripButton, ToolStripLabel, ToolStripSplitButton, ToolStripDropDownButton, ToolStripSeparator, ToolStripComboBox, ToolStripTextBox, ToolStripProgressBar gibi öğeler de eklenebilmektedir.

StatusStrip

Genelde kullanıcıya bilgi vermek için kullanıldığından Bilgi Şeridi olarak da adlandırılır. Formun en altına konumlanır. Kullanıcıya anlık tarih ve saat bilgisi, aktif kullanıcı bilgisi gibi bilgileri göstermek için kullanılır. StatusStripLabel, StatusStripProgressBar, StatusStripDropDownButton ve StatusStripSplitButton gibi 4 farklı öğesi bulunmaktadır.

YAZDIRMA İŞLEMLERİ

C# ile geliştirilen uygulamaların çoğunda yazıcıdan çıktı alma işlemi de yapılır. Böylece kullanıcı gerek duyduğunda uygulama içerisinde yazdırma işlemi yapabilir. Yazdırma işlemi sırasında yazı boyutunu, rengini, konumunu ayarlamak için “Font”, “SolidBrush”, “PrintDocument” gibi farklı sınıflar kullanılmaktadır.

PrintDocument

PrintDocument sınıfı yazdırma işlemleri için kullanılan bir sınıftır. PrintDocument kullanılabilmesi için programa System.Drawing.Printing namespace eklenmelidir.

Yazdırma işlemi için “Print” metodu kullanılır. Print metodu çağrıldığında yazdırılacak dokümanın özelliklerini belirleyen “PrintPageEventArgs” sınıfı içerisinde yer alan PrintPage olayı tetiklenir.

PrintPageEventArgs Sınıfı

Yazdırma işlemleri sırasında yazdırılacak dokümanın özelliklerinin belirlenmesi, yazdırmanın nasıl yapılacağı gibi işlemler için PrintPageEventArgs sınıfına ait özellikler kullanılmaktadır. Bu sınıf altında Graphics, HasMorePages, PageBounds, PageSettings, Cancel gibi özellikler bulunmaktadır.

PrinterSettings Sınıfı

Yazdırılacak dokümanın nasıl yazdırılacağı hakkında bilgi almak için PrinterSettings sınıfı kullanılmaktadır. CopyCollate, DefaultPageSettings, Duplex, PrinterName, IsDefaultPrinter, IsValid, PaperSizes, FromPage, CanDuplex gibi özellikler kullanılmaktadır.

Font Sınıfı

Yazdırılacak metnin stilini, font tipini, boyutunu ayarlamak için kullanılır. Bu sınıf altında Bold, Italic, Height, Style, Underline, Size gibi özellikler bulunmaktadır.

Brush Sınıfı

Fırça oluşturup şekillerin içini doldurmak için kullanılan abstract bir sınıftır.

PrintDialog Sınıfı

Kullanıcılar herhangi bir belgeyi yazdırırken yazdırma özelliklerinin yer aldığı bir pencere açılır. Bu pencerede yazıcı ismi, yazdırılacak sayfa numaraları, kopya sayısı gibi ayarlar yapılabilir. C#'ta da yazdırma işlemleri sırasında “PrintDialog” sınıfı ile bu pencereye benzer bir pencere oluşturulup gerekli ayarlamalar yapılabilir.

Uygulamada PrintDialog sınıfı kullanacak ise ilk önce ToolBox'tan forma PrintDialog kontrolü sürükleyip bırak yöntemi ile eklenmelidir. Daha sonra kod içerisinde nesnesi oluşturulur.

AllowSelection, ShowHelp, ShowNetwork, PrintToFile, AllowSomePages gibi özellikler bulunmaktadır.

PrintPreviewDialog Sınıfı

Yazdırılacak olan dokümanın ön izlemesi “PrintPreviewDialog” kontrolü ile elde edilebilir.

Baskı ön izleme işlemi için ilk olarak araç kutusundan PrintPreviewDialog ve PrintDocument seçilerek form üzerine eklenir ve birbirleriyle iletişim kurmaları sağlanır. Böylece PrintDocument'de yer alan veriler PrintPreviewDialog kısmına aktarılmış olur.



DERS ADI Görsel Programlama I
ÜNİTE ADI Çoklu Formlarla Çalışma
ÜNİTE NO 14
YAZAR Öğr. Gör. İSHAK METEHAN SİS

C# FORM YAPISI

C#’ta arayüz tasarlanırken çeşitli form kontrollerinden ve bu kontrollerin özelliklerinden yararlanılmaktadır. Kontroller form içinde bulunan nesneler olarak tanımlanabilir.

Form Özellikleri

Windows Form uygulaması açıldığında Visual Studio tarafından projeye otomatik olarak bir form atanır.

Form kontrolünün özelliklerine Özellikler penceresinden erişilebilmektedir.

AcceptButton: Enter tuşuna basılınca hangi tuşun işlem yapacağını belirtir. AutoScroll: Form üzerindeki nesneler forma sığmadığı durumda otomatik olarak scrollbar eklenip eklenmeyeceği bu özellik ile belirlenir.

AutoScrollMargin: Eğer formda scrollbar var ise scrollbar'ın genişliği ve yüksekliğini ayarlar.

AutoSize: Formun içindeki kontrollere göre formun boyutunun otomatik olarak ayarlar.

BackColor: Form’un zemin rengini değiştirmek için kullanılır. Custom, Web ve System olmak üzere 3 seçeneği mevcuttur.

BackgroundImage: Form’un arka plan resmini ayarlar.

BackgroundImageLayout: Forma eklenen arka plan resminin nerede ve nasıl duracağını belirler. None, Tile, Center, Stretch, Zoom olmak üzere 4 farklı özelliği mevcuttur.

CancelButton: “ESC” tuşuna basılınca form üzerinde çalışacak buttonu ayarlar.

Enabled: Bu özellik ile formun kullanıcı tarafından kullanılıp kullanılmayacağı, form üzerindeki elemanların aktif olup olmayacağı belirlenir

Font: Form üzerine eklenen elemanların yazı tipi özellikleri ayarlanır.

FontDialog: Sistemde o anda yüklü olan yazı tiplerini gösterir.

FormBorderStyle: Formun boyutların değiştirilip değiştirilmeyeceğini ayarlar.

HelpButton: Başlık çubuğunda Yardım düğmesinin gösterilip gösterilmeyeceği ayarlanır.

Locked: Bu özellik ile hem formun boyutu hem de konumu değiştirilmez hale getirilir.

Language: Form’un dilini belirler. Default değeri sistemin kurulu olduğu dildir.

Is MDI Container: MDI (Multiple Document Interface) Çoklu Form Arayüzü demektir. “True” değeri seçilirse formun içinde başka bir form açılabilir demektir. TopMost: Form açıldığında başka form varsa, diğer tüm formların üzerinde gözükp gözükmeyeceğini ayarlar.

Text: Formun sağ üstteki ikonun yanındaki yazıyı değiştirmek için kullanılır. TopMost: Form açıldığında başka form varsa, diğer tüm formların üzerinde gözükp gözükmeyeceğini ayarlar.

WindowState: Formun tam ekran mı yoksa normal boyutlarda mı gösterileceğini ayarlar. Normal, Minimized, Maximized özellikleri mevcuttur.

StartPosition: Uygulama çalıştırıldığında Formun, ekranın neresinde açılacağını belirtir. Manuel,

CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds, CenterParent gibi seçenekleri mevcuttur.

Size: Formun boyutlarını ayarlar.

Name: Bu özellik kullanılarak formun adı değiştirilebilir.

Form özelliklerinin yanı sıra işlem yapılmasını sağlayan Click, Closing, KeyDown, KeyUp, Load gibi olayları vardır.

Form Kontrolleri ve Özellikleri

TextBox: Kullanıcının çok satırlı veri girmesini sağlar. MultiLine, PasswordChar, ReadOnly, MaxLenght, ScrollBars gibi özellikleri mevcuttur.

Button: Kullanıcının işlem yapmak için tıklayabileceği düğme olarak tanımlanabilir.

Label: Kullanıcıya bilgi vermek için kullanılır. Label üzerinde yer alan metin kullanıcı tarafından düzenlenemez, TextAlignin, Image, ImageAlign gibi özellikleri mevcuttur.

CheckBox: Kullanıcının bir ya da birden fazla seçeneği seçmesine olanak tanır. ComboBox: Açılır bir menüde verilerin görüntülenmesini ve kullanıcıya veri girişi yapmayı sağlar.

CheckedListbox: Her bir öğenin yanında onay kutusuyla birlikte bir öğe listesi görüntüler.

Selectedindex, Checkeditems.count, Selecteditem gibi özellikleri mevcuttur.
DataGridView: Verilerin, hücrelerden ve satırlardan oluşan tablolarda ki gösterimini sağlayan bir kontroldür. Rows, Columns, DataSource, MultiSelect gibi özellikleri mevcuttur.
DateTimePicker: Tarih veya saat bilgisini gösterir.
GroupBox: Form üzerinde yer alan diğer kontrollerin gruplanmasını sağlar.
ListBox: Kullanıcının bir listeden bir veya daha fazla öğe seçmesini sağlayan kontroldür.
OpenFileDialog: Kullanıcıların bir iletişim kutusu kullanarak dosyaları açmasına izin verir.
PageSetupDialog: Kullanıcının yazdırma işlemleri sırasında sayfa yapısını ayarlamayı sağlar.
PrintDialog: Kullanıcılar herhangi bir belgeyi yazdırırken yazdırma özelliklerinin yer aldığı pencerenin açılmasını sağlar.
PrintDocument: Yazdırma işlemleri için kullanılır.
PrintPreviewDialog: Yazdırma işlemi sırasında Baskı Ön İzleme için kullanılır.
RadioButton: Kullanıcıya iki veya daha fazla seçenek sunmayı sağlar.
RichTextBox: Kullanıcıların metin girmesine, görüntülemesine ve güncellemesine izin verir.
MenuStrip: Form üzerine menü eklenmesini sağlar.
ContextMenuStrip: Form üzerinde, farenin sağ tuşu ile tıklanınca görünen menü eklenmesini sağlar.
ToolStrip: Form üzerinde araç çubuğu menüsünün eklenmesini sağlar.
StatusStrip: Kullanıcıya bilgi vermek için kullanılan Bilgi Şeridi olarak da adlandırılan menünün eklenmesini sağlar.

ÇOKLU FORMLAR

Formlar Arası Geçiş

Bir Visual Studio projesi otomatik olarak bir form ile açılmaktadır. Projeye ikinci bir form, menüden sırası ile PROJECT-Add Windows Form denilip açılan pencereden Windows Form seçeneği seçilip formun ismi girilerek eklenmektedir.

İkinci formun ekranda görünmesi için ikinci form türünde bir nesne oluşturup bu Show ya da ShowDialog komutu ile gösterilmektedir.

Formlar Arası Veri Aktarımı

C#'ta formlar arası veri gönderme işlemi Form açarken, Form kapatırken, Class kullanarak, Sql ile formlar arası veri gönderme olacak şekilde 4 farklı şekilde yapılabilmektedir.

Formlar açarken formlar arası veri gönderimi

Form açılırken diğer forma veri göndermek için kullanılır.

Formlar kapatırken formlar arası veri gönderimi

Bu yöntem genelde işlem sonunda, önceki forma veri göndermek için kullanılır.

Class kullanarak formlar arası veri gönderimi

Bir class oluşturulur ve bu class içinde diğer forma gönderilecek değişkenler tanımlanır.

SQL kullanarak formlar arası veri gönderimi

SQL de yer alan tablodaki aktarılacak veriler diğer formdan çağrılarak veri aktarımı gerçekleştirilmiş olur.