

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKÎ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

GÖRSEL PROGRAMLAMA KURULUMU

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. GEREKLİLİKLER	3
1.1. .NET Framework	3
1.2. Visual Studio .NET	3
1.3. Sistem Gereklilikleri	4
1.3.1. Visual Studio .NET Programının Kurulması	4
1.4. Windows ve Web Formları	8
1.5. İnternet Tarayıcısı, HTML	11
1.6. GUI (Graphical User Interface)	12
UYGULAMA FAALİYETİ	13
ÖLÇME VE DEĞERLENDİRME	14
ÖĞRENME FAALİYETİ-2	15
2. WINDOWS UYGULAMASI	15
2.1. Proje, Kaynak Dosya, Solution	15
2.1.1. Proje Çeşitleri	16
2.2. MSIL (Microsoft Intermedia Language)	17
2.3. Assembly (Bütünlük) ve Native Code (Doğal Kod)	18
2.4. .NET Framework Class Library ve Common Language Runtime	18
2.5. Integrated Development Environment (IDE)	20
2.6. Dosya Aç ve Kaydet Diyalog Kutuları	20
2.7. *.vbproj, *.vb ve *.sln Dosya Uzantıları	21
2.8. Form Tasarım, Kod Düzenleyici, “Solution Explorer”, Araç Çubuğu, Araç Kutusu ve Özellikler Penceresi	22
2.8.1. Solution Explorer	23
2.8.2. Form Tasarım Penceresi	24
2.8.3. Kod Penceresi	24
2.8.4. Menü ve Araç Çubukları	25
2.8.5. Toolbox (Araç Kutusu) Penceresi	26
2.8.6. Properties (Özellikler) Penceresi	27
2.9. “Tool *Options* “ Penceresi	28
UYGULAMA FAALİYETİ	30
ÖLÇME VE DEĞERLENDİRME	31
ÖĞRENME FAALİYETİ-3	32
3. NESNELER	32
3.1. Araç Kutusu, Özellikler Penceresinin Ayarları	32
3.1.1. Araç Kutusu Penceresi	32
3.1.2. Özellikler Penceresinin Ayarları	40
3.2. Form Tasarımındaki Nesne Boyutlandırma Tutamacı	41
3.3. Yerleşim Araç Çubuğu	42
3.4. Form ve Nesnelerin Ortak Özellikleri (name, text)	45
3.5. Form Özellikleri	47
3.6. Nesnelerin Özellikleri	51
UYGULAMA FAALİYETİ	53
ÖLÇME VE DEĞERLENDİRME	54
MODÜL DEĞERLENDİRME	56
CEVAP ANAHTARLARI	58
KAYNAKÇA	59

AÇIKLAMALAR

KOD	482BK0066
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Görsel Programlama Kurulumu
MODÜLÜN TANIMI	Kurulum ve gereklilikler ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Bilişim Teknolojilerinin Temelleri modüllerini bitirmiş olmak
YETERLİK	Görsel programlama dilinin kurulumunu yapmak.
MODÜLÜN AMACI	Genel Amaç: Bu modül ile gerekli ortam sağlandığında görsel programlama dilinin gerekliliklerini düzenleyebilecek, basit bir Windows uygulaması yapabileceksiniz. Amaçlar: 1. Görsel programlama dilinin gerekliliklerini sağlayabileceksiniz. 2. Windows ortamı için program geliştirebileceksiniz. 3. Genel kontrol nesnelere ile çalışabilecek ve araç kutusunu özelleştirebileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Bilgisayar laboratuvarı ve bu ortamda bulunan; görsel programlama için gerekli donanıma sahip bilgisayar, lisanslı işletim sistemi programı, kâğıt ve kalem hazır bulundurulmalıdır.
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Modül içinde ve sonunda verilen öğretici sorularla edindiğiniz bilgileri pekiştirecek, uygulama örneklerini ve testleri gerekli süre içinde tamamlayarak etkili öğrenmeyi gerçekleştireceksiniz. Sırasıyla araştırma yaparak, grup çalışmalarına katılarak ve en son aşamada alan öğretmenlerine danışarak ölçme ve değerlendirme uygulamalarını gerçekleştireceksiniz.

GİRİŞ

Sevgili Öğrenci,

Günümüzde bilgisayar kullanım oranı gün geçtikçe artıyor ve artan kullanıcı sayısı yeni ihtiyaçları, yeni talepleri beraberinde getiriyor. Bütün bu istekler karşısında sürekli yeni ürünler / teknolojiler geliştiriliyor ve piyasaya sunuluyor. Bu talebi karşılamak için ortaya çıkan teknolojilerden biri de Microsoft'un birkaç yıldır geliştirmekte olduğu "Kişileri, kurumları ve sistemleri birbirine bağlayan yazılımlar" olarak tanımlanan .NET teknolojisidir.

.NET yeni bir stratejinin adıdır. .NET, Windows tabanlı yazılım geliştirmeyi tamamen değiştiren, uğraştığımız sorunlara çözüm bulan yeni teknolojiler grubudur. Önümüzdeki yıllarda yapılacak uygulamalar için bir tasarıdır. .NET teknolojisi geliştiricilerin işlemleri daha az kod yazarak, daha kısa sürede gerçekleştirmelerini, çok daha yüksek performans ve güvenlik ile çalışmalarına olanak tanır. .NET'in asıl amacı, İnternet Teknolojileri kullanarak Web servisleri aracılığıyla diğer uygulamalar ile haberleşmektir. .NET'in başlangıcı, .NET Framework'tür. .Net Framework'ü herhangi bir programlama işi için büyük bir fonksiyon koleksiyonu olarak düşünülebilirsiniz. Visual Basic, öncelikle Windows uygulamaları için tasarlanan basit bir dil olmuştur. Visual Basic diline başlamadan önce bir çok şeyi öğrenmek zorunluluğu yoktur. Fakat Visual Basic ile .Net Framework nesnelere ulaşmak çok sınırlı bir şekilde olmaktadır. Visual Basic dilinin en eski versiyonu ile yapılamayan ancak diğer dillerde yapılan hemen her şey yeni versiyonu ile yapılabilmektedir.

Bu modülü başarı ile bitirdiğiniz takdirde, Visual Studio.NET'in bilgisayara kurulumunu yapabilecek, program geliştirebilecek, genel kontrol nesnelere ile çalışabilecek ve araç kutusunu özelleştirebileceksiniz.

Modülü başarı ile tamamlayacağınıza inanıyor ve modülün hayatınızda olumlu değişiklikler yapmasını arzu ediyor, çalışmalarınızda başarılar diliyorum.

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu modül ile gerekli ortam sağlandığında, görsel programlama dilinin gerekliliklerini sağlayabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

- Görsel programlama dillerini diğer programlama dilleri ile karşılaştırınız. Sonuçları rapor haline getirip arkadaşlarınız ile paylaşınız.

1. GEREKLİLİKLER

1.1. .NET Framework

.NET Framework, farklı programlama dilleri ve kütüphaneleri ile birlikte daha kolay Windows tabanlı uygulamalar oluşturmak için Microsoft tarafından yapılan geliştirme ve uygulama ortamıdır. .NET Framework, Windows tabanlı uygulamalarda gerekli olan alt yapıyı sağlar.

- Standart network (ağ) protokol ve şartnamelerini destekler
- Farklı programlama dillerini destekler
- Farklı dillerde geliştirilen kütüphane programlarını destekler.

Windows Server 2003 ve Windows XP gibi farklı platformları destekler. .NET Framework, Common Language Runtime (CLR) ve Framework Class Library (FCL) olmak üzere iki ana parçadan meydana gelmektedir. Bunlarla ilgili detaylı bilgi ileride verilecektir.

1.2. Visual Studio .NET

Visual Studio .NET, Windows ve Web uygulamalarını geliştirmek için bir ortamdır. En büyük özelliklerinden birisi IDE (Integrated Development Environment-Tümleşik Geliştirme Ortamı) dediğimiz ortak bir uygulama geliştirme platformu sunmasıdır. Visual Basic .NET, uygulamalarınız için kullanacağınız dillerden sadece biridir. Gerçekte Visual Studio .NET birçok dile ev sahipliği yapacak şekilde tasarlandı ve birçok firma, Visual Studio .NET içerisindeki tümleşik birçok dille çalışıyor. İster C#, J#, C++ ile uygulama geliştirin, ister Visual Basic .NET ile; kullanacağınız en etkin ve ortak uygulama geliştirme platformu Visual Studio .NET olacaktır.

Öyleyse Visual Studio .NET ile diğer programlama dilleri arasındaki fark nedir? Visual Studio .NET uygulamaları geliştirmek için gerekli tüm araçları sağlayan bir ortamdır. Programlama dilleri Windows uygulamalarının sadece bir yönüdür. Uygulamaların sanal arabirimi özel bir dile bağımlı değildir. Uygulamalarınızın arabirimleri programların uygulamaları için kullandıkları dil koduna bakılmadan tüm programlar tarafından aynı zamanda kullanılabilir. Visual Studio .NET, veri tabanına bağlantı sağlamak, nesnelere kontrol etmek, bilgilere yeniden ulaşma ve hatta herhangi bir dilden ulaşabileceğiniz nesnelere depolamak için araçlar sağlar.

1.3. Sistem Gereklilikleri

Visual Basic .NET programını kurmak için bilgisayarınızda, programın ihtiyaç duyduğu donanımın minimum özelliklere ya da daha fazla özelliklere sahip olması gerekir.

Sistem için minimum özellikler aşağıdaki gibi olmalıdır.

- **İşlemci:** 600 megahertz (MHz) Pentium serisi
- **Ram:** 192 megabytes (MB)
- **Disk Alanı:** 3GB (Boş alan)
- **Çözünürlük:** 800 x 600 256 RenkÇözünürlük/Renk Derinliği
- **İşletim sistemi:** Windows NT, Windows 2000, Windows XP, Windows Server 2003 işletim sistemlerinden biri. Windows 9x ve Me üzerinde Visual Basic.NET kurulumu imkânsız değildir, ancak Microsoft tarafından kesinlikle desteklenmemektedir.

1.3.1. Visual Studio .NET Programının Kurulması

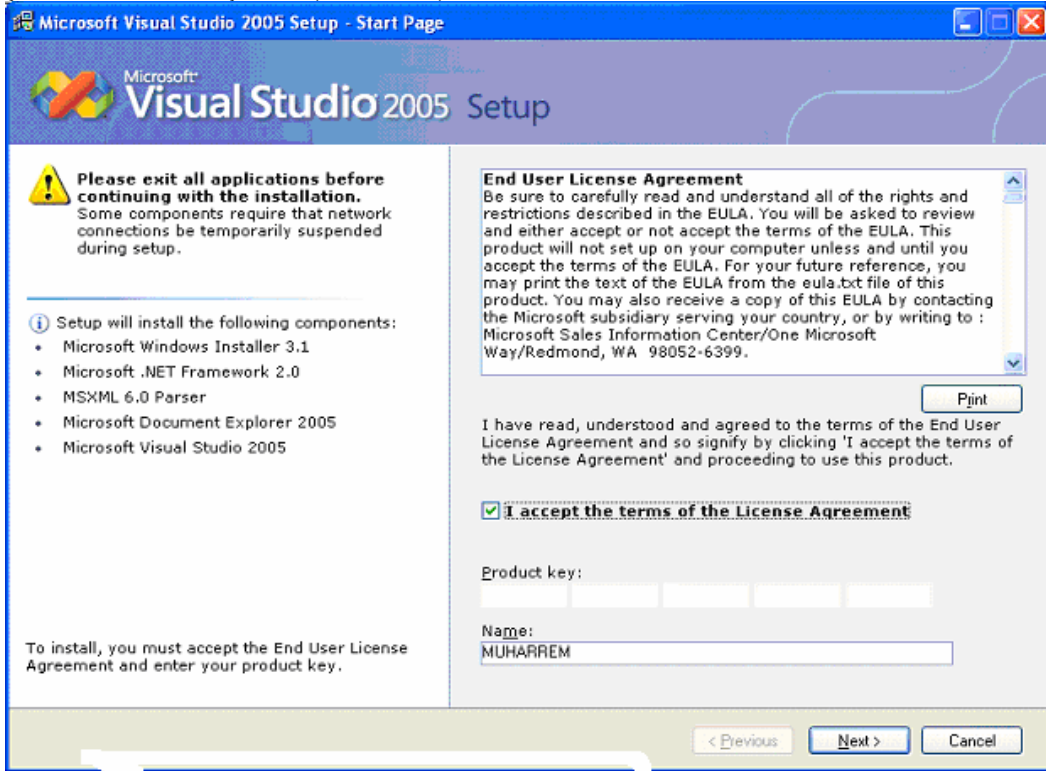
Visual Studio .NET paketinde Visual Basic .NET, Visual C++ .NET, ve Visual C# .NET programlama dilleri yer almaktadır. İsteğe bağlı olarak tüm programlama dillerini kurabileceğiniz gibi sadece bu modülde incelenecek olan Visual Basic .NET'i de kurabilirsiniz.

Visual Studio .NET CD/DVD'sini bilgisayarınıza takınca ekrana Visual Studio .NET kurulum penceresi gelir (Resim1.1).



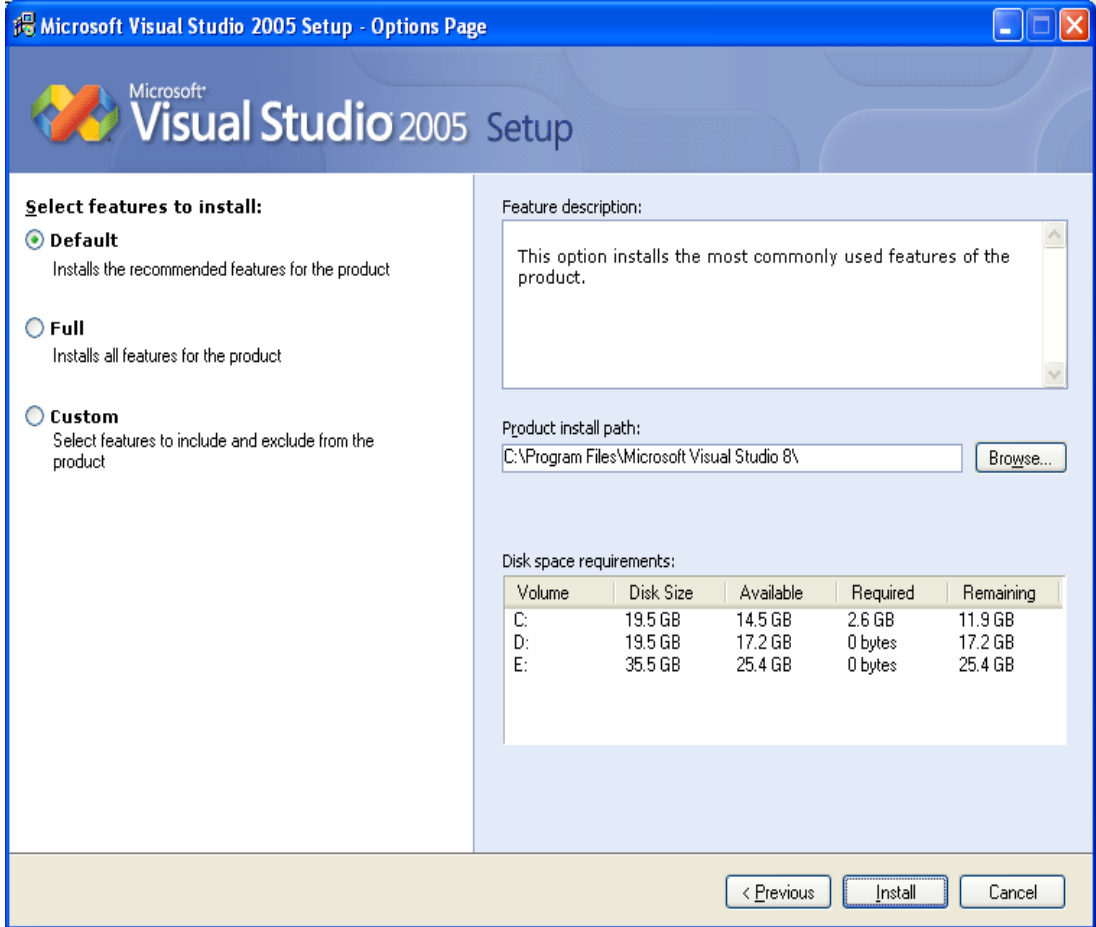
Resim1.1: Visual Studio .NET kurulum penceresi

Visual Studio .NET kurulum penceresinde “Install Visual Studio 2005” sekmesi seçilerek kurulum başlatılır (Resim 1.2).



Resim 1.2: Visual Studio .NET son kullanıcı lisans sözleşmesi penceresi

Lisans sözleşmesini kabul etmek ve kurulumu devam etmek için “**I accept the agreement (Sözleşmeyi kabul ediyorum)**” yazan seçeneği işaretliyoruz. Böylece Next butonu aktif hale geliyor. Next butonunu tıkladığınızda kurulum seçenekleri, kurulumun nereye yapılacağı ve bilgisayarınızdaki sabit sürücülerin disk kapasitesini gösteren pencere gelecektir (Resim 1.3).

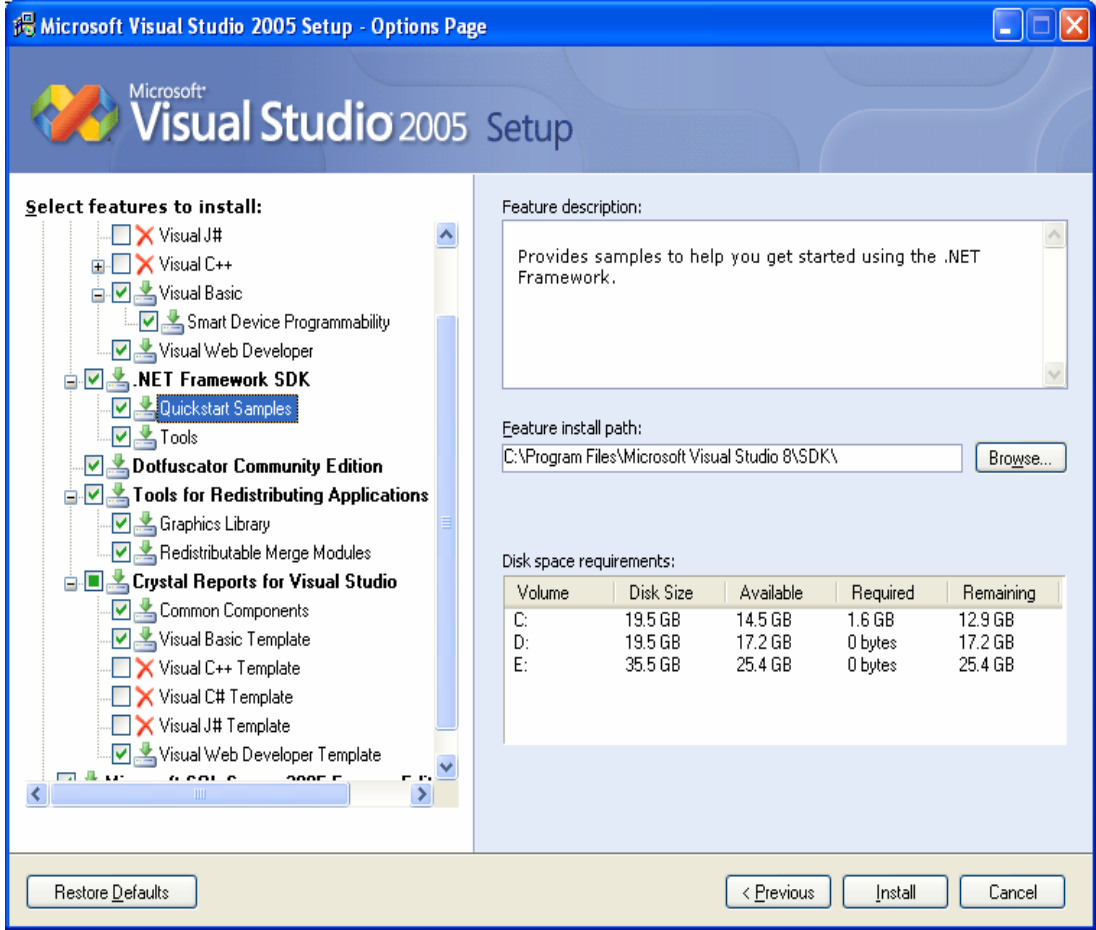


Resim 1.3: Visual Studio .NET kurulum seçenekleri penceresinin görünümü

Kurulum seçenekleri penceresinde Default, Full ve Custom seçenekleri bulunur.

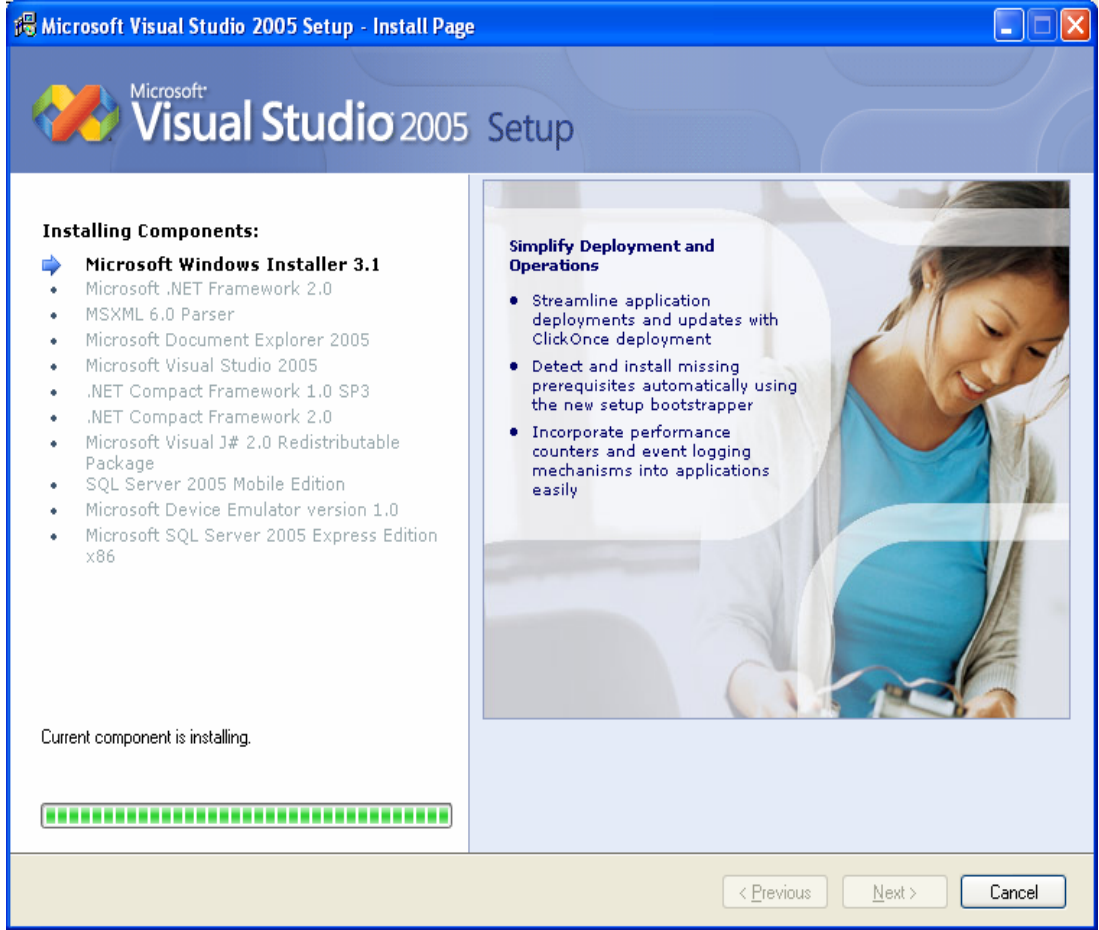
- **Default:** Ürün için tavsiye edilen özelliklerin kurulumu
- **Full:** Üründeki tüm özelliklerin kurulumu
- **Custom:** Üründeki tanımlı özellikleri seçme veya iptal etme

Bilgisayarınıza Visual Studio .NET 2005 içerisinde bulunan programlama dillerinden sadece Visual Basic .NET programını kurabilmeniz için “Custom” seçeneğini işaretlemeniz yeterli olacaktır (Resim 1.4).



Resim 1.4: Custom seçeneği seçildikten sonra kurulum penceresine ait görünüm

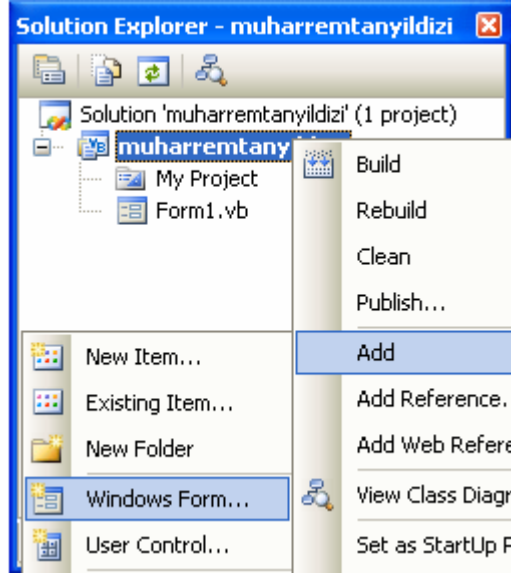
Resim 1.4'te verilen yapıya benzer şekilde kurulum özelliklerini belirledikten sonra “Install” düğmesine tıklayınca kurulum başlayacaktır (Resim 1.5).



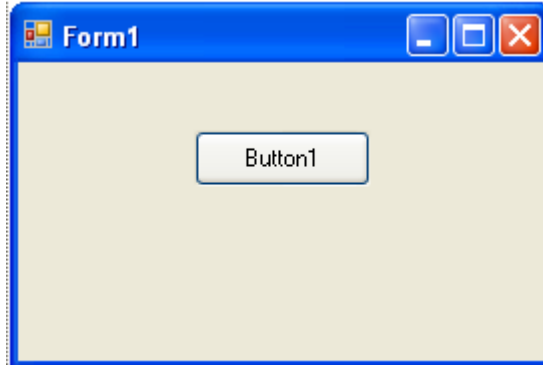
Resim 1.5: Kurulum başlama görüntüsü

1.4. Windows ve Web Formları

Windows formları, projenin kullanıcı arabirimidir. Hazırlayacağınız projelere bilgi giriş ve çıkışını bu formlar ve üzerine yerleştireceğiniz nesnelere yardımıyla yapacaksınız. Visual Basic .NET'te form eklemek için Solution Explorer penceresindeki projeyi kaydettiğiniz adaya sağ tıklayınca gelen menüden sırasıyla "Add/Windows Form" komutları verilir (Resim 1.6, 1.7).



Resim 1.6:Windows Form ekleme penceresi

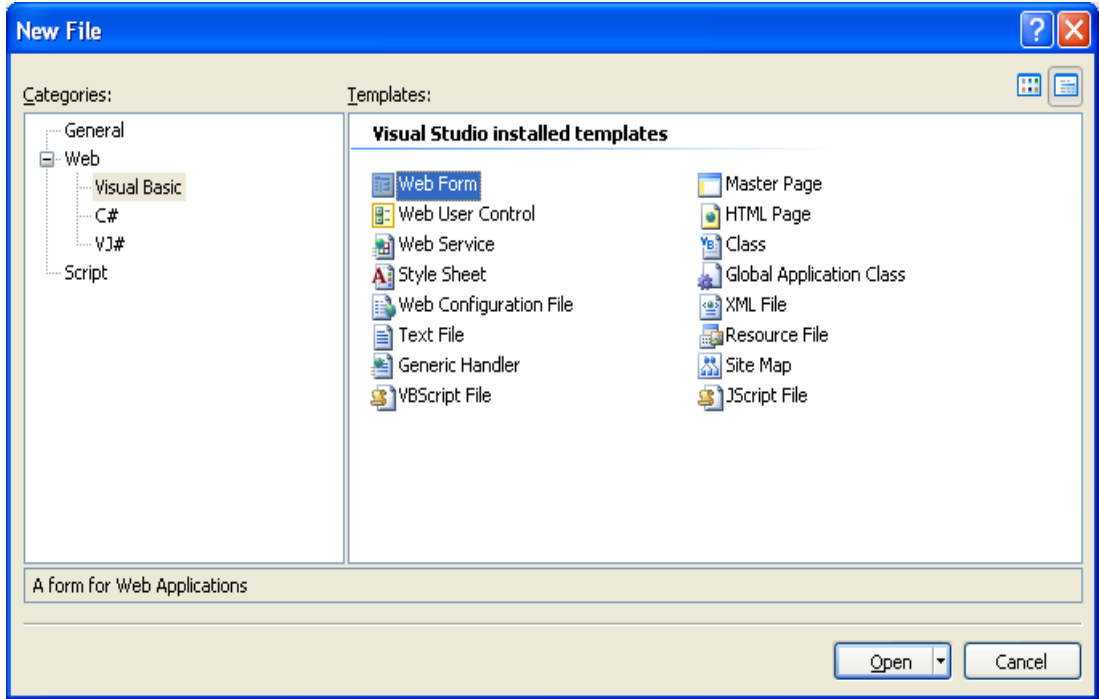


Resim 1.7:Windows Formu

Programın kullanıcı arabirimini yani formu görüntülemek için Solution Explorer penceresindeki View Designer düğmesini, kodlarını görüntülemek için ise View Code düğmesini tıklayabilirsiniz.

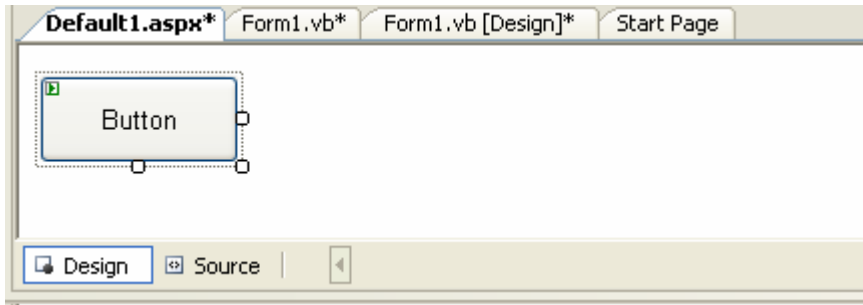
Visual Basic .NET ile programlama yaparken Web Formlarını da kullanabilirsiniz. Web formları iki bileşenden oluşur; görsel kısım ve kod kısmı. Görsel kısım içlerinde HTML elemanlarını ve Web Form kontrollerini barındırır. Hem Görsel kısım hem de Kod kısmı .aspx dosyamızın içinde yer alır.

Web Form eklemek için File Menüsü / New / File komutları sırasıyla verilir (Resim 1.8).

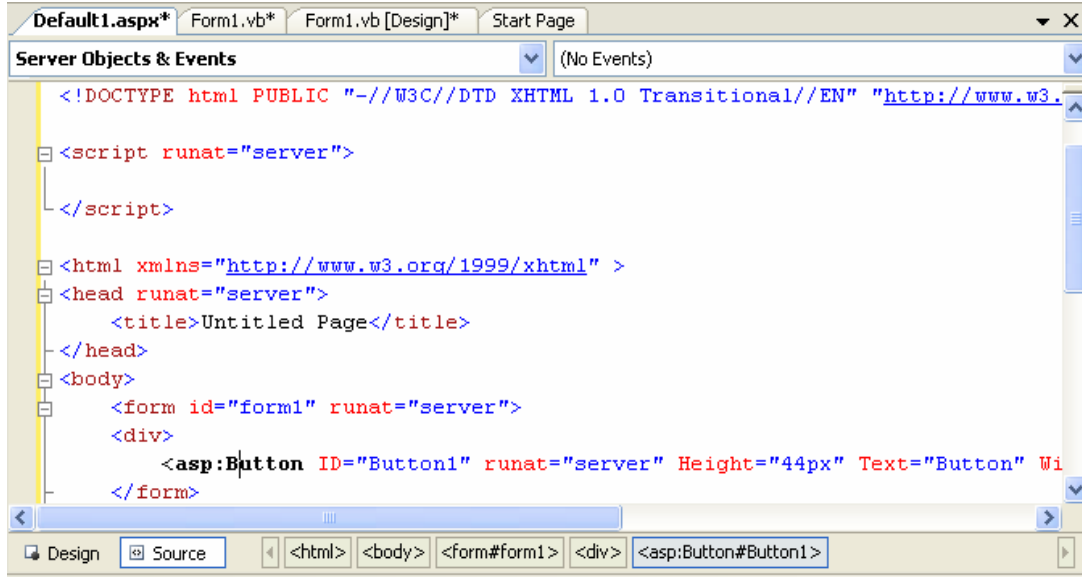


Resim 1.8: Web Form ekleme penceresi

Web formu projenize ekledikten sonra tasarıma ve kaynak kod yazmaya imkân veren bir pencere karşınıza çıkar (Resim 1.9,1.10).



Resim 1.9: Web Form Tasarım (Design) penceresi



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml" >
<script runat="server">
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Button ID="Button1" runat="server" Height="44px" Text="Button" Width="100px" />
</div>
</form>
</body>
</html>
```

Resim 1.10: Web Form Kaynak Kod (Source) penceresi

Web formlarını ve kodlarını ayrı ayrı yazsak bile aslında bunun ikisi tek bir birim olarak ele alınır. Web formları .NET Framework'u tarafından derlendikten sonra tek bir sınıf olarak ortaya çıkar. Ortaya çıkan yeni sınıfımız Visual Basic .NET'in Page sınıfından türetilmiş bir sınıf olup ek olarak içinde sizin kodlarınız, Web form kontrolleri ve statik HTML kodları yer alır.

Web formlarının web sunucusunda çalışma şekli normal bir web sayfası veya herhangi bir Web programlama modeline çok benzer. Önce sayfa ,Web sunucusundan istenir. Sunucu sayfayı bulur ve eğer sayfa dinamik sayfa ise (asp, php, jsp veya aspx gibi) gerekli işleri dinamik içerik oluşturmadan sorumlu bileşen yapar ve kullanıcının tarayıcısına yollar. Fakat ASP.NET'te .NET Framework birçok işi sizin yerinize yapar. Mesela tarayıcınızın yapısına göre en uygun sonuç üretilir ve Web sunucusuna, oradan da size yollar.

1.5. İnternet Tarayıcısı, HTML

Hyper Text Markup Language (HTML) belgelerin birbirlerine nasıl bağlanacaklarını ve belge içindeki metin ve resimlerin nasıl yerleşeceklerini belirleyen ve etiket (tag) denilen kod parçalarından oluşan bir sistemdir. HTML, internet üzerinde Web sayfası oluşturmak için kullanılan bir dildir. HTML dosyalarının aktarımı için HTTP (Hyper Text Transfer Protocol) kullanılır. HTML dosyaları sunucu bilgisayarın sabit diskinde .html ya da .htm uzantısı ile saklanır. Htm dosyaları düz yazı dosyalarından başka bir şey değildir.

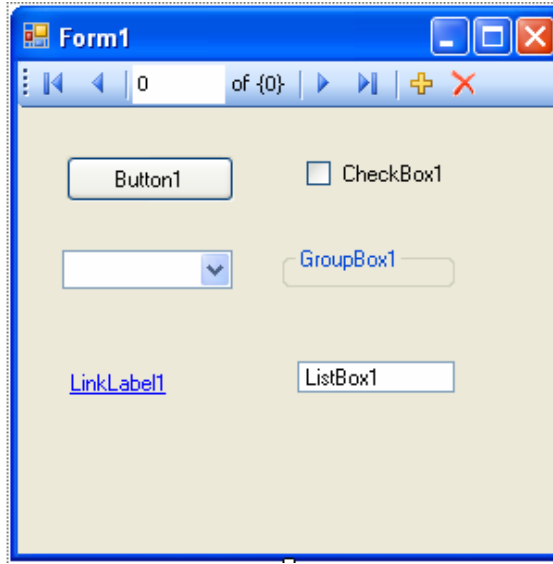
Visual Basic .NET ile programlama yaparken HTML Page sayfalarını da kullanabilirsiniz. HTML Page tıpkı Web Form gibi iki bileşenden oluşur; tasarım ve kaynak kod kısmı. Tasarım kısmı içerinde HTML elemanlarını ve Web Form kontrollerini barındırır. Hem tasarım kısmı hem de kaynak kod kısmı aynı .htm dosyanızın içinde yer alır.

HTML sayfası eklemek için File Menüsü / New / File / HTML Page komutları sırasıyla verilir.

1.6. GUI (Graphical User Interface)

Çeşitli Windows uygulamaları geliştirmek için formlar kullanılır. Bu formlar programın görsel kısmını oluşturup pencereler, menüler, düğmeler vb araçlar yoluyla kullanıcı ile iletişim kurmayı sağlar. Form ve üzerindeki kontrol elemanları beraberce grafiksel kullanıcı arabirimi (GUI) olarak adlandırılmaktadır.

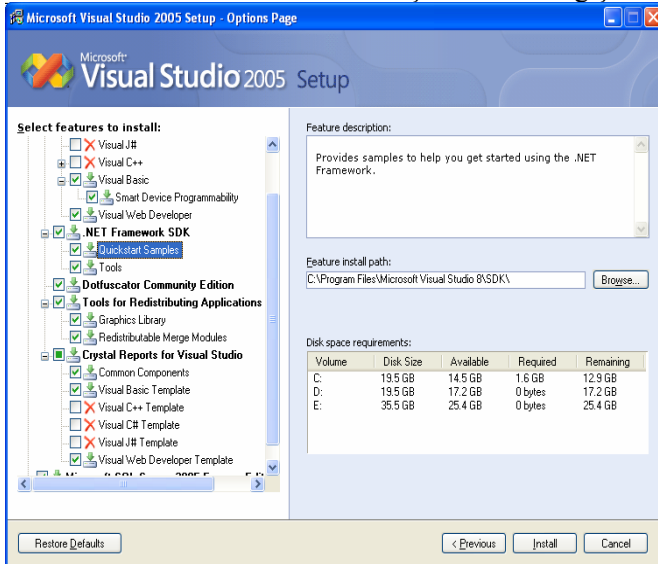
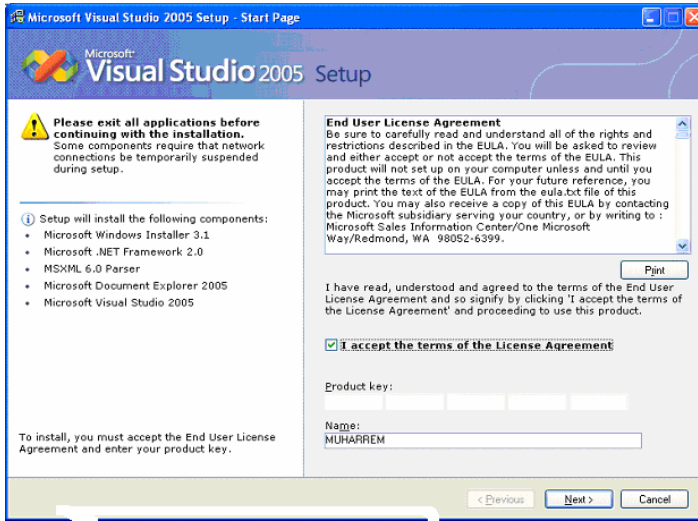
GUI, formlar aracılığıyla programlar çalıştırmayı, komutlar vermeyi, bilgisayarla etkileşim kurmayı sağlayan görsel temelli çalışma ortamıdır.



Resim 1.11: Grafiksel kullanıcı arabirimi (GUI)

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Visual Studio .NET programının kurulabileceği işletim sistemlerini inceleyerek en uygun olarak karar verdiğinizin özelliklerini defterinize yazınız.</p>	<p>Farklı işletim sistemlerine kurulum yapılması denenebilir.</p>
<p>2. Visual Studio .NET programını bilgisayarınıza kurunuz.</p>	<p>Visual Studio .NET kurulurken Default, Full ve Custom seçenekleri arasındaki farklılık göz önünde bulundurulabilir.</p>
<p>3. Kurulum sırasında kurulum seçeneklerini değiştiriniz.</p>	



ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki sorulardan; ilk 9 soruda verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız. Diğer sorular için uygun şıkkı işaretleyiniz.

1. Visual Basic. NET’in kurulumu için minimum işlemci 600 megahertz (MHz) Pentium serisi olmalıdır. ()
2. Visual Basic. NET’in kurulumu için minimum RAM 256 MB olmalıdır. ()
3. Visual Basic. NET’in kurulumu için minimum 2 GB boş disk alanına ihtiyaç vardır.()
4. Visual Basic. NET’in kurulumu için minimum 800*600 256 Renk Çözünürlüğüne ihtiyaç vardır.. ()
5. Visual Basic. NET Windows 98 üzerinde sorunsuz kurulabilir. ()
6. Web Form eklemek için File Menüsü / New / File komutları sırasıyla verilir.()
7. Form ve üzerindeki kontrol elemanları beraberce grafiksel kullanıcı arabirimi GUI (Graphical User Interface) olarak adlandırılmaktadırlar. ()
8. Visual Studio .NET paketinde Visual Basic .NET, Visual C++ .NET, ve Visual C# .NET programlama dilleri yer almaktadır. ()
9. .NET Framework, Common Language Runtime (CLR) ve Framework Class Library (FCL) olmak üzere iki ana parçadan meydana gelmektedir. ()
10. Aşağıdakilerden hangisi kurulum esnasında özelleştirme yapabilmemizi sağlar?
A) Default
B) Custom
C) Full
D) Install

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internette veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

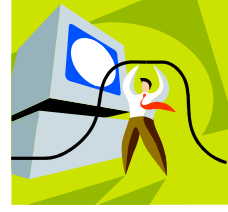
ÖĞRENME FAALİYETİ-2

AMAÇ

Bu modül ile gerekli ortam sağlandığında, Windows ortamı için program geliştirebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

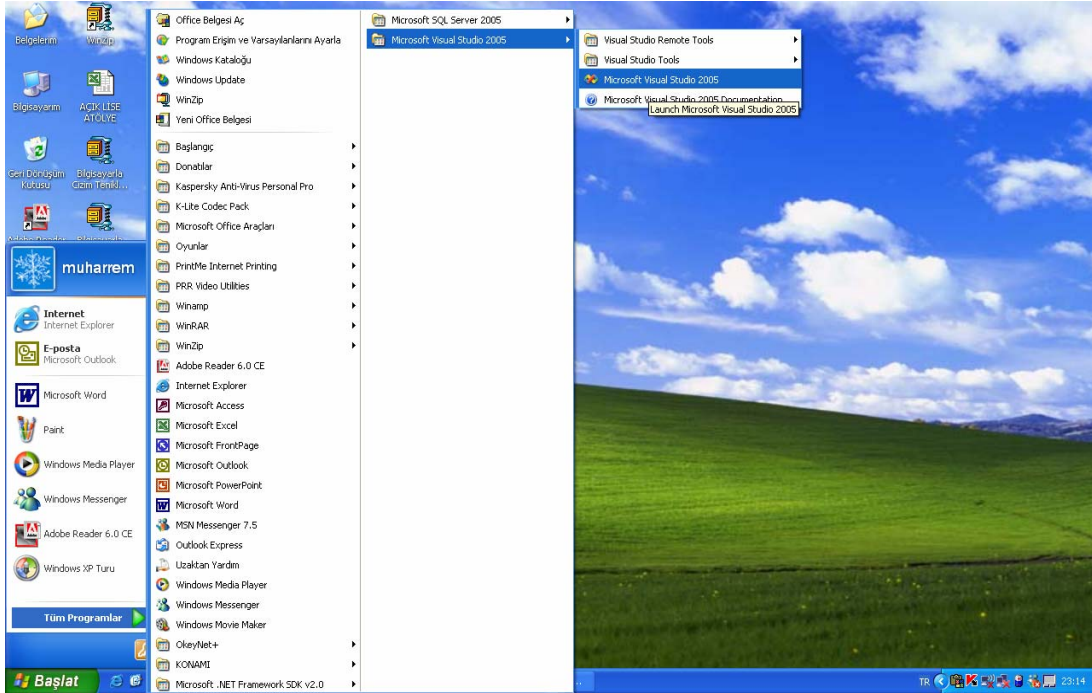


- Daha önce kullandığımız Windows uygulamalarının hepsinde ortak olarak kullanılan menü ve araç kutularında bulunan komutların görevlerini araştırınız. Sonuçları rapor haline getirip arkadaşlarınız ile paylaşınız.

2. WINDOWS UYGULAMASI

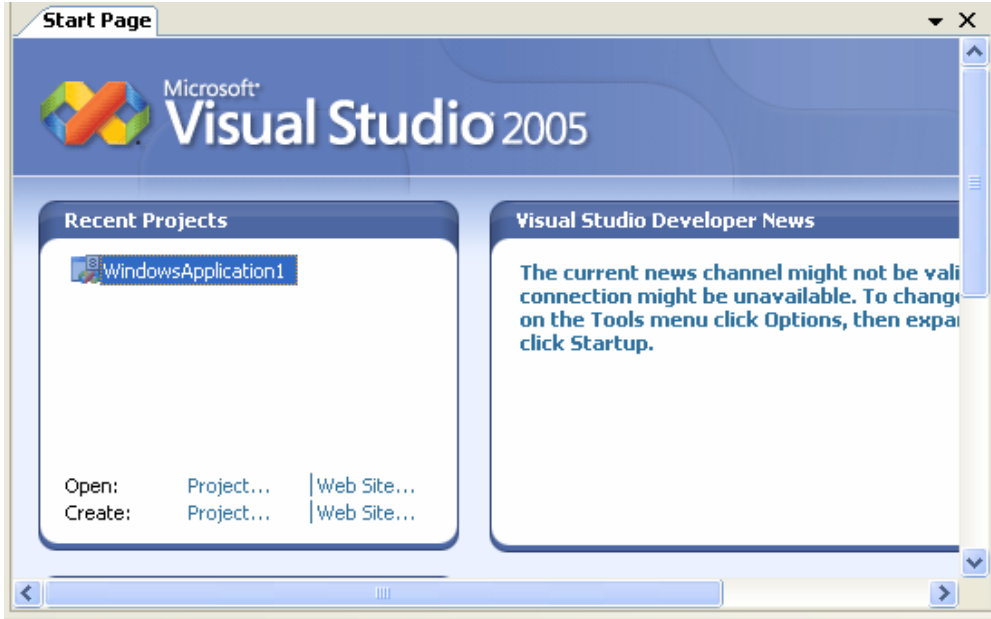
2.1. Proje, Kaynak Dosya, Solution

Visual Basic .NET’i çalıştırmak için programı kurduktan sonra Başlat/Programlar/Microsoft Visual Studio/Microsoft Visual Studio işlem sırası izlenir.



Resim 2.1:Visual Studio .NET’i başlatma ekranı

Program çalıştırıldığında önce aşağıdaki Start Page diyalog kutusu ekrana gelir (Resim2.2).



Resim 2.2:Visual Studio .NET Start Page ekranı

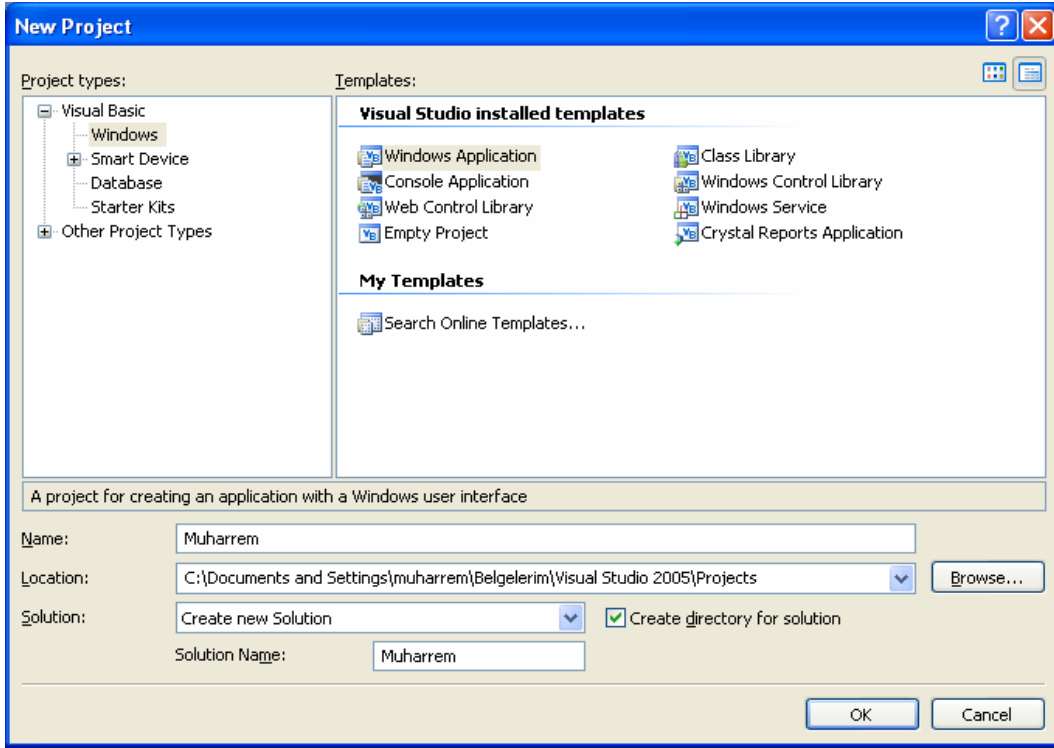
Giriş sayfasında bulunan Recent Projects kısmında bulunan Open Project (Proje Aç) ile daha önce kaydedilmiş bir projeyi açabilirsiniz. Create Project (Proje Oluştur) ile de yeni bir proje oluşturmak için ilk adımı atabilirsiniz.

Getting Started seçeneği ise Visual Studio .NET ile ilgili bilgilere ve örnek uygulamalara ulaşmak için kullanılacak seçenekler bulunur.

2.1.1.Proje Çeşitleri

Standart araç çubuğunda bulunan New Project ya da Recent Project kısmında bulunan Create Project seçeneği tıklandığında hazırlayacağınız programlara uygun projeler seçilebilir (Resim 2.3).

- **Windows Application:** İçerisinde form bulunan Standart Windows uygulamalarıdır.
- **Console Application:** Konsol uygulamaları geliştirebileceğiniz, içerisinde form bulunmayan proje tipidir.
- **Web Control Library:** Web sayfalarında kullanmak istediğiniz kontrollerinizi tasarlayabileceğiniz proje tipidir.
- **Empty Project:** Bu proje tipi ile çeşitli Windows uygulamaları geliştirebilirsiniz.



Resim 2.3:New Project(Yeni Proje) Penceresi

- **Class Library:** Çeşitli sınıflar oluşturarak Windows tabanlı uygulamalarda çalıştırabileceğiniz projedir.
- **Windows Control Library:** ActiveX denetimi gibi Windows uygulamalarında çalışabilecek kontroller oluşturabilirsiniz.
- **Windows Service:** Windows arka planında çalışan servisleri ve bunların kodlarını içinde bulunduran proje tipidir.
- **Crystal Reports Application:** Kristal raporlar oluşturabilirsiniz

Visual Basic .NET projelerinize otomatik olarak isim verecek ve “Belgelerim\Visual Studio 2005\Projects “ klasörü altında yine kendi adında bir klasör oluşturarak kaydedecektir. İsterseniz projenin adını ve konumunu değiştirebilirsiniz.

2.2. MSIL (Microsoft Intermedia Language)

MSIL (Microsoft Intermediate Language) yani Microsoft Ara Dili , nesneye yönelik programlamayı destekleyerek, Visual Basic .NET’te yazmış olduğunuz kodların ilk derleme anında dönüştürüldüğü bir ara dildir.

MSIL işlemciden bağımsız komut setinden oluşmaktadır. Bu komut seti içerisinde nesnelere yüklenmesi, depolanması ve çalıştırılmasını sağlayan komutların yanı sıra aynı zamanda nesnelere üzerinde metod çağırmasını sağlayan komutlar da yer almaktadır. .NET

Framework ile yazılmış bir kaynak kodun derlenmesi sonucu MSIL kodu oluşmaktadır. NET uygulamalarının kaynak kodları çalışır hale gelebilmek için iki derleme aşamasından geçmektedir. İlk derlemede kaynak kodunuz, **MSIL** (Microsoft Intermediate Language) ara dili ile, Assembly (sembolik makine) diline yakın bir dile dönüştürülmektedir. Oluşan .exe ve .dll dosyalar yürütülebilir kodlar yerine ara dil içermektedir. Bu dosyaları çalıştırmak istendiğinde, devreye CLR girecektir. Bu ara dildeki talimatlar, ikinci derleme aşamasına girecektir ve doğal makine koduna dönüşecektir. Bu ikinci derleme, uygulamanın çalışacağı yerde olacağından oluşacak doğal kod o ortama özgü olacaktır.

2.3. Assembly (Bütünlük) ve Native Code (Doğal Kod)

Assembly; herhangi bir mikroişlemciye ait sembolik makine dilidir. Makinenin anlayabildiği tek dil 0 ve 1'dir. Makine dili çok hızlıdır. Çünkü yapılan tek işlem hexadecimal bir sayıyı binary sayıya çevirmektir.

.NET 'teki Assembly kavramı ile herhangi bir mikroişlemciye ait makine dili birbirlerinden çok farklı şeyleri ifade etmektedir. Net 'te derlenen sınıf kütüphaneleri (class library 'ler) ismine Assembly denilen dll dosyalarını oluşturur.

Bu anlamda bir Assembly; bir ya da daha fazla fiziksel dosyayı barındıran bir koleksiyondur. Bu dosyalar; Assembly içerisindeki derlenmiş sınıfların kodları, resimler, resource dosyaları ve diğer binary dosyalardır.

Daha önceden Visual Basic programı ile, bilgisayarınızda hazırladığınız bir uygulamayı, kopyalayıp başka bir bilgisayara taşıdığınızda uygulamanın çalışmadığını gördünüz. Bunun nedeni; uygulamayı taşıdığımız bilgisayardaki işletim sisteminin, uygulamanızın kullandığı bileşenler hakkında herhangi bir bilgiye sahip olmamasıydı. Bu bilgiyi alamamasının nedeni kimlik bilgilerini (isim, versiyon bilgisi vb.) registry'e (kayıt defteri) yazmalarındadır.

Visual Basic.NET'te ise geliştirilen Assembly tekniği sayesinde bir Visual Basic .NET programı bulunduğu dizin ile birlikte (.Net Framework'ün kurulu olduğu) başka bir bilgisayar kopyalandığında çalışacaktır. Çünkü .Net Assembly'leri; olarak kimlik bilgilerini registry'e yazmak yerine kendi üzerlerinde taşır.

2.4. .NET Framework Class Library ve Common Language Runtime

.NET Framework aşağıdaki bileşenleri içerir.

➤ **Framework Class Libraries:**

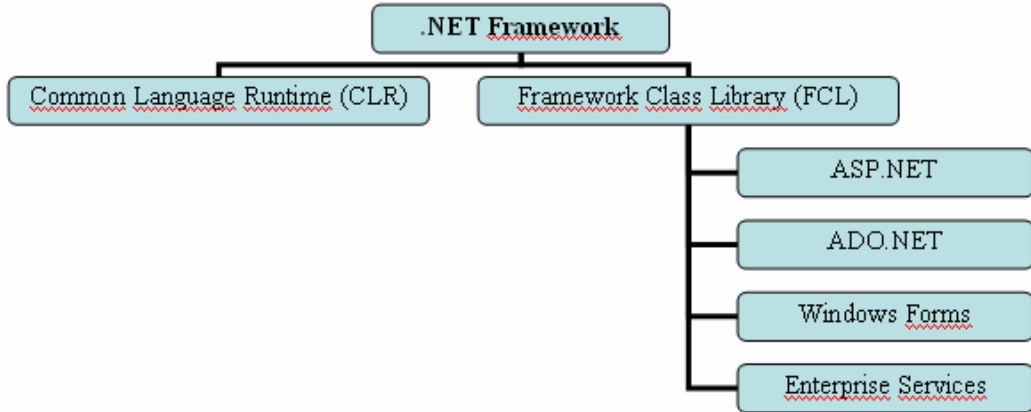
Temel sınıf kütüphanelerinden sorumludur. Nesneye dayalı işlevsel kütüphane fonksiyon paketlerini içerir. Giriş/çıkış, string işlemleri, güvenlik yönetimi, ağ iletişimi, thread yönetimi, metin işleme ve kullanıcı arayüz tasarımı gibi standart fonksiyonları kapsamaktadır.

ADO.NET sınıfları; OLE DB, ODBC, Oracle ve SQL Server arayüzleri üzerinden veriye erişimi sağlamaktadır. XML sınıfları ise XML manipulasyonu, XML veri içerisinde arama yapabilme ve dönüşüm fonksiyonları gibi temel fonksiyonlara sahiptir. ASP.NET sınıfları Web tabanlı uygulamalar ve Web servisleri geliştirebilmek için destek sağlamaktadır. Windows Forms sınıfları ise masaüstü uygulamaları geliştirebilmeyi sağlamaktadır.

Sınıf kütüphaneleri, .NET Framework dilleri arasında ortak, tutarlı bir geliştirme arayüzü sağlamaktadır.

Sınıf kitaplıklarında ASP.NET, ADO.NET, Windows Formları ve Enterprise Services teknolojileri yer almaktadır.

- **ASP.NET:** Web servislerine tam destek vererek, Web erişimli uygulamaları geliştirmek için kullanılır.
- **ADO.NET:** İlişkisel veri tabanı yönetim sistemlerinde yer alan verilere erişmek için kullanılır.
- **Windows Forms;** Her bir .NET Framework programlama dili için, Windows tabanlı arabirimler yapmak için standart sınıflardan meydana gelir.
- **Enterprise Services;** COM+ servislerine erişmek için ilgili sınıfları içerir.



Şekil 2.1: .NET Framework altında çalışan teknolojiler

- **Common Language Runtime (CLR):**

Common Language Runtime çalışma zamanı servislerinden sorumludur. .NET uygulamalarının değişik sistemler üzerinde çalışmasını sağlayan çalışma zamanı kütüphaneleri sağlar. Uygulamaları yönetmeye yardımcı olacak servisleri sağlayan tarafsız geliştirme ve uygulama ortamlarını içerir. Programlama dillerinin entegrasyonu (language integration), güvenlik (security), bellek (memory), süreç (process) ve thread (iş parçacığı) yönetimi çalışma zamanı servislerindedir.

CLR'nin sağladığı ve sorumlu olduğu bazı servisler aşağıda listelenmektedir:

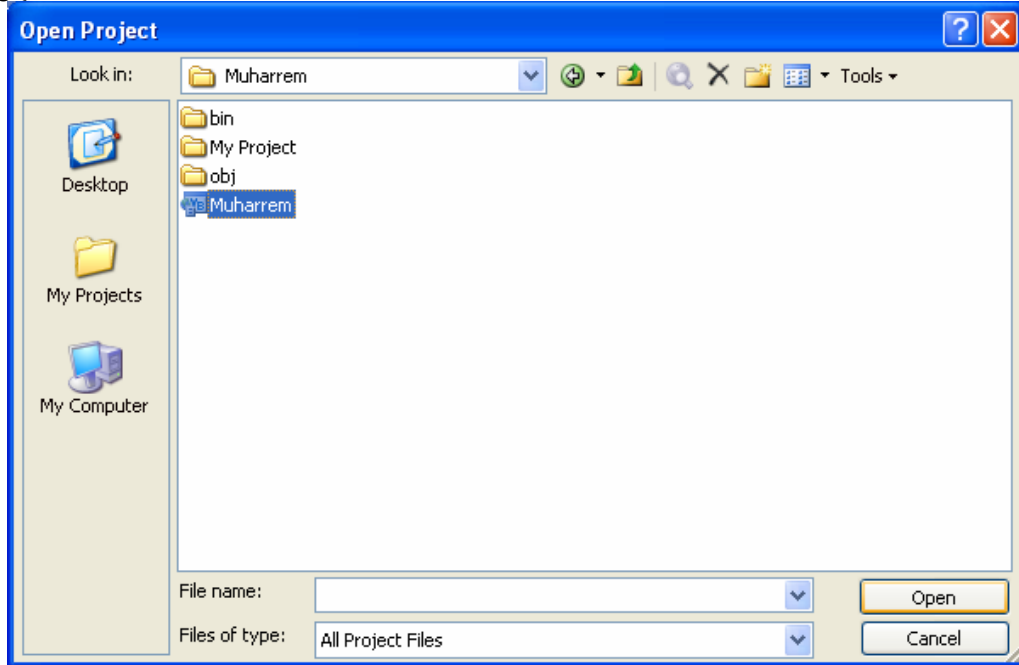
- Code Management- Kod yönetimi
- Type Safety -Tip güvenliğinin doğrulanması
- Intermediate Language kodunun doğal makine koduna (native code) dönüşümünün sağlanması
- Managed Object -Yönetilebilir nesnelerin bellek yönetimi
- Code Access Security- Kod erişim güvenliğinin uygulanması
- Exception Management -İstisna yönetimi

2.5. Integrated Development Environment (IDE)

IDE’de birçok görsel araç vardır. Bu araçlar görsel menüler tasarlamaya izin verir. Bir menünün tasarımı herhangi bir kod içermez ve tıklama işlemi ile gerçekleştirilebilir. Elbette herhangi bir dili kullanarak menülerin ardına istenilen kod ilave edilebilir. IDE’nin amacı program geliştiricilerin kod yazmadan önce görsel araçlar ile mümkün olan şeyleri yapmaya izin vermektir. IDE, uygulamalarınız için tasarlama, çalıştırma ve hata ayıklamayı sağlar.

2.6. Dosya Aç ve Kaydet Diyalog Kutuları

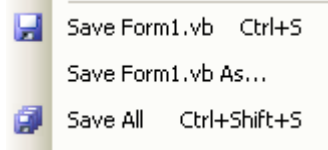
Daha önce çalıştığımız bir projeyi açmak istiyorsanız File menüsündeki Open / Project butonu tıklanır. Butonu tıkladığımız anda karşınıza “belgelerim” klasöründe yer alan Visual Studio Projects klasöründeki projelere ait klasörler listelenir (Resim 2.4). Projelere ait klasörlerin başka bir dizinde açılmasını istiyorsanız, genel olarak bunun yolunu değiştirebilirsiniz.



Resim 2.4: Visual Basic .NET Open Project penceresinin görüntüsü

Eğer yeni bir proje açmak istiyorsak **File** menüsündeki **New Project** butonu tıklanır. Yeni projelerin neler olabileceğini daha önce öğrenmiştiniz.

Üzerinde çalıştığımız bir projeyi kaydetmek için ise File Menüünde bulunan üç farklı kaydetme seçeneğini isteğinize göre kullanabilirsiniz (Resim 2.5).



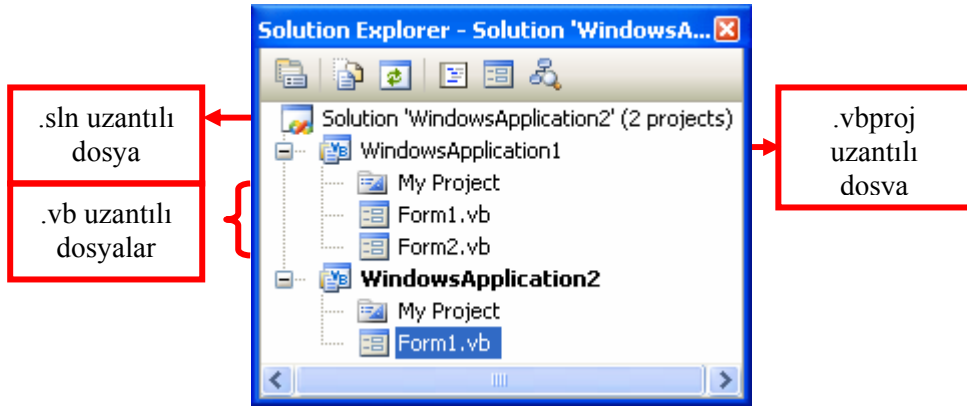
Resim 2.5: Visual Basic .NET kaydetme komutları

Bunlardan birincisi **Save Form1.vb**'dir. Bu komut ile sadece aktif olan Windows Formunu kaydetmek için kullanılır. Bir diğer komut olan **Save Form1.vb As** ise, aktif olan formun ismini veya kayıt yerini değiştirerek kaydetmeye yarar. Üçüncü komut **Save All**'dir. Bu komut ise, projemiz dahilinde tüm dosyaların kaydedilmesini sağlar.

2.7. *.vbproj, *.vb ve *.sln Dosya Uzantıları

Visual Basic .NET, hazırlayacağınız programları projects (projeler) ve solution (çözümler) olmak üzere tümleşik bir ortamda oluşturur. Solution birden fazla projeyi içerdiğinden aynı anda daha fazla proje ile çalışabilmenizi sağlayacaktır.

Visual Basic .NET'te hazırlayacağımız programlar değişik uzantılarda kaydedilir. Kaydedilen bu dosyaların hepsi program çalıştığında Solution Explorer penceresi altında yer alır (Resim 2.6).

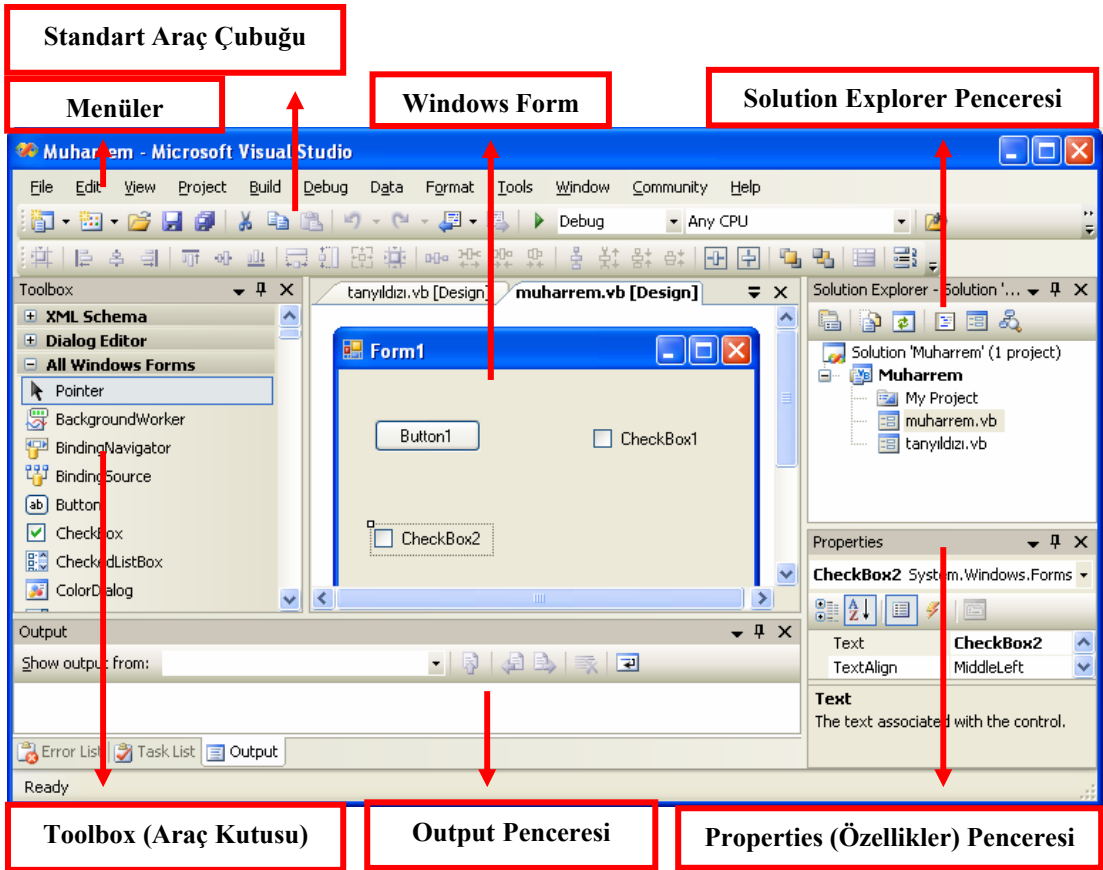


Resim 2.6: Visual Basic .NET Solution Explorer penceresi

- **.sln:** Solution Explorer penceresinde görülen projeler ve bunlara ait dosyaların kısa yollarının kaydedildiği dosyanın uzantısıdır.
- **.vbproj:** Üzerinde çalıştığınız projenin uzantısıdır. İçerisinde projenin yapısını oluşturan windows formları ve görsel sayfaları içerir.
- **.vb:** Projemiz ile ilgili olan tasarımın yapıldığı ve kaynak kodlarının yazıldığı windows formlarının uzantısıdır.

2.8. Form Tasarım, Kod Düzenleyici, “Solution Explorer”, Araç Çubuğu, Araç Kutusu ve Özellikler Penceresi

Visual Basic .NET ekranının genel görüntüsü aşağıdaki gibidir (Resim2.7).



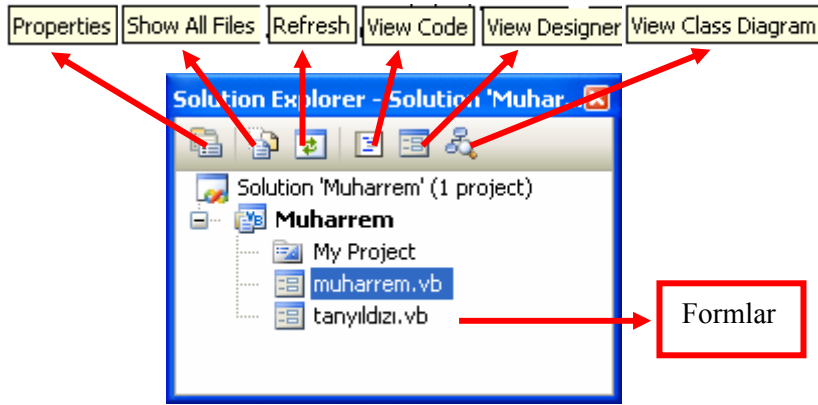
Resim 2.7: Visual Basic .NET ortamının genel görüntüsü

2.8.1. Solution Explorer

Solution Explorer, projenizde kullandığınız tüm dosyaları düzenlemiş olarak gösteren ve birbirleri arasında kolay geçiş yapmanızı sağlayan arabirimdir. Üzerinde çalıştığımız projeye eklenen form ve nesnelerin neler olduğunu görmek istiyorsanız **Solution Explorer** penceresine bakabilirsiniz. Visual Studio.Net' in sağ tarafında bulunur. Eğer ikinci bir proje Visual Studio.NET' e dahil edilirse bu da Solution penceresinde görülür.

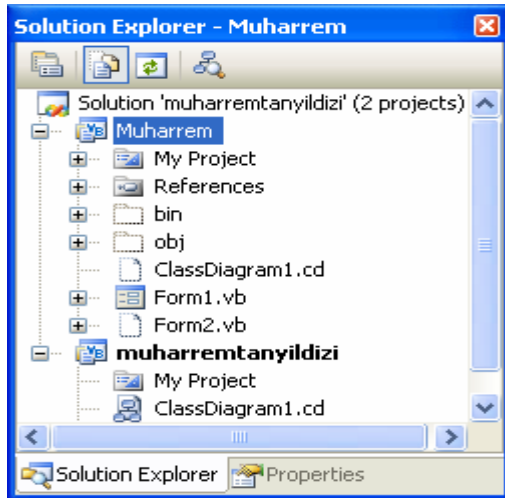
Solution Explorer penceresini görünür hale getirmek isterseniz “**Ctrl+Alt+L**” kısayol tuşunu veya “**View (Görünüm)**” menüsünden Solution Explorerı tıklayarak seçebilirsiniz.

Solution penceresi ile, kod penceresini, form tasarımını, Properties penceresini ve tüm dosyaları görüntüleyebilirsiniz.



Resim 2.8: Visual Basic .NET Solution Explorer penceresi

Solution Explorer penceresindeki ilk seçeneğe sağ tıklayıp Add/Existing Item komutunu verince ikinci bir proje ekleyebilirsiniz.

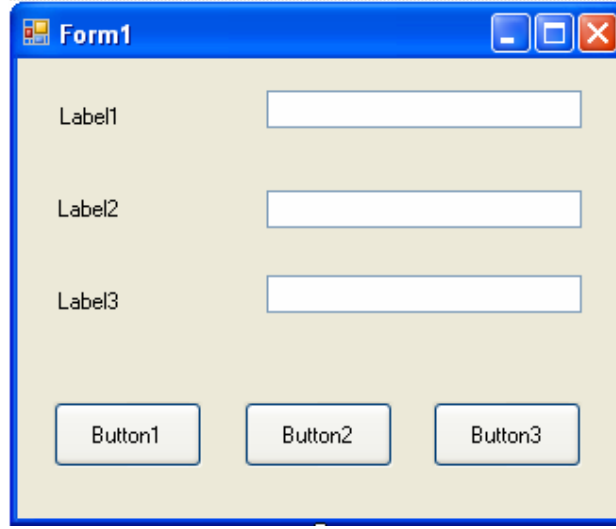


Resim 2.9: Açık olan iki projenin görüntüsü

Çalışılan iki projeden hangisinin başlangıç projesi olacağına yine biz karar verebiliriz. Hangi projeyi aktif yapacaksanız onun üzerinde sağ tıklayıp **Set Up StartUp Project** komutunu vermeniz gerekmektedir.

2.8.2. Form Tasarım Penceresi

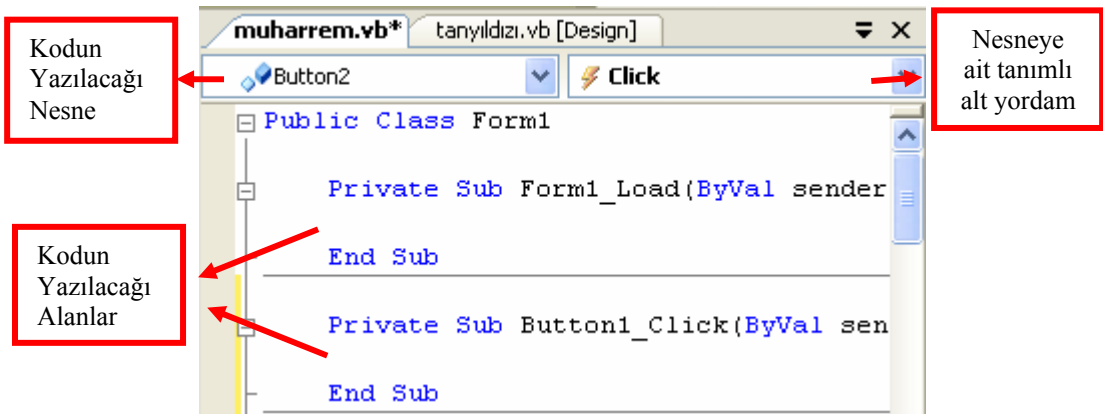
Bu form programa ait kullanıcı arabiriminin oluşturulduğu yerdir. Yani tüm denetimler öncelikle bu form üzerine yerleştirilir ve yerleştirilen denetimler bu forma ait birer nesne olarak çalışır. Burada oluşturulan form görüntüsü, program çalıştığı zaman ekranda görülür.



Resim 2.10: Form Tasarım penceresi


2.8.3. Kod Penceresi

Programımızda bulunan denetim elemanlarına ait olay alt yordamlarının bulunduğu ve program kodlarının yani asıl programın yazıldığı bölümdür. Visual diller programcının kendi kodunu yazacağı bir editör sunar. Üzerinde çalışılan form veya kontrol elamanı çift tıklatıldığında kod penceresi karşımıza gelecektir (Resim 2.11).



Resim 2.11: Kod penceresi

Visual Basic .NET'te program kodları yani programın çalışmasını sağlayan ve kullanıcı tarafından yazılmış ifadeler Code (Kod) penceresi içerisine yazılır. Bu pencereye geçmenin iki yolu vardır.

- Solution Explorer penceresinde yer alan  (View Code) ikonuna basarak geçilebilir.
- Form veya kod penceresi açılmak istenen form üzerindeki herhangi bir nesne üzerinde çift tıklayarak da geçilebilir.

Bu bölüme yazılan kodlar nesnelere için önceden tanımlanmış alt yordamlar şeklindedir. Bu alt yordamlar her nesne için değişiklik göstermektedir. Alt yordamlar program kod penceresi bölümünde Private Sub ile başlayıp End Sub ile bitmektedir. Kodlar Private deyimi ile End Sub deyimi arasında yazılmaktadır. Bir kodun çalışması ise nesneye ait tanımlı olayın gerçekleşmesi ile olur.

2.8.4. Menü ve Araç Çubukları

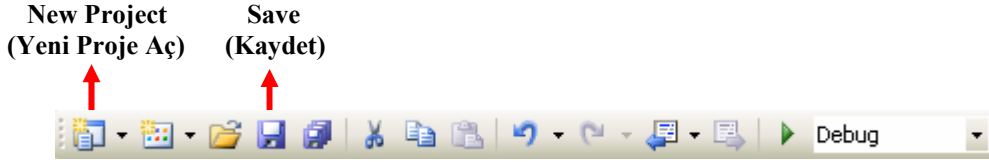
Visual Basic .NET ortamını etkili bir şekilde kullanabilmek için gerekli olan komutlar menü çubuğunda yer alan menü seçenekleri altında bulunmaktadır.

File Edit View Project Build Debug Data Tools Window Community Help

Resim 2.12: Visual Basic .NET menü çubuğu

- **File menüsü:** Proje açmak, kapatmak, kaydetmek, yazdırmak gibi dosyalama işlemlerinin yapıldığı menüdür.
- **Edit menüsü:** Yapılan işlemleri geri ve ileri almak, bilgileri kesmek, kopyalamak, yapıştırmak, aramak gibi işlemlerin yapıldığı menüdür.
- **View menüsü:** Pencere ve araç çubuklarını gizleyip göstermek için kullanılan menüdür.
- **Project menüsü:** Projeye form, modul, referans eklemek ve proje dosyalarını yönetmek için kullanılan menüdür.
- **Debug menüsü:** Programları adım adım çalıştırmak, hata ayıklamak vb. işlemler için kullanabileceğiniz menüdür.
- **Data menüsü:** Veri tabanları ile çalışmak için kullanabileceğiniz menüdür.
- **Format menüsü:** Formların ve form üzerindeki elamanların görünüşü ve hizalanması için kullanabileceğiniz menüdür.
- **Tools menüsü:** Projede yeni bir fonksiyon tanımlamak, üzerinde çalışılan pencerelerin özelliklerini değiştirmek için kullanabileceğiniz menüdür.
- **Window menüsü:** Pencere ayarlarını değiştirmek ve istenilen pencereye geçiş yapmak için kullanılan menüdür.
- **Community menüsü:** MSDN aracılığıyla forumlara katılabileceğiniz menüdür.
- **Help menüsü:** Yardım seçeneklerine, .NET ile ilgili son bilgilere ulaşabileceğiniz menüdür.

Menüler altında bulunan komutlara kolayca erişebilmenizi araç çubukları sağlar. Çalışma sayfanızda bulunmayan herhangi bir araç çubuğunu sırasıyla View Menüsü/Toolbars üzerine gelerek ekleyebilirsiniz. Araç çubukları üzerinde bütün komutlar bulunmamakla birlikte istenilirse herhangi bir komut düğmesini ekleyebilirsiniz.

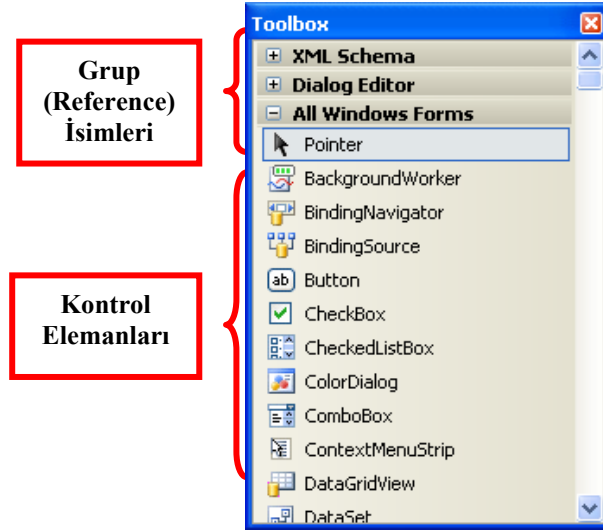


Resim 2.13: Visual Basic .NET Standart Araç Çubuğu

2.8.5. Toolbox (Araç Kutusu) Penceresi

Visual Basic.Net'te kullanacağınız bir diğer pencerede **ToolBox** penceresidir. Visual Basic .Net penceresinin sol tarafında bulunur ve gizlenmiştir. Fare ile üzerine gelindiğinde bu pencere açılır.

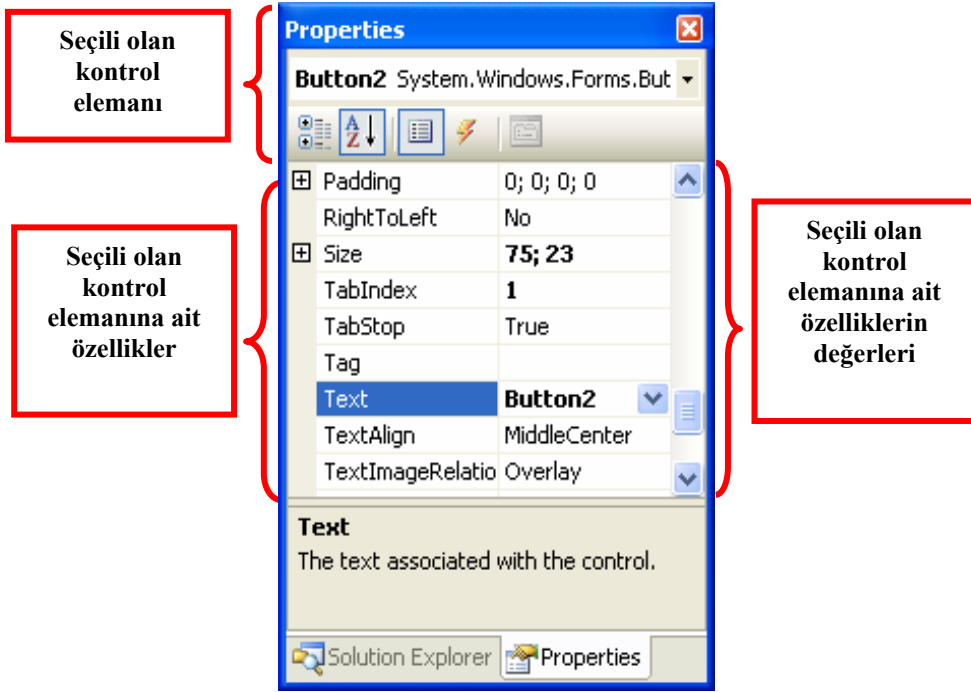
Projeleriniz formlar ve bu formlar üzerinde bulunan nesne dediğimiz elemanlardan oluşmaktadır. Bu nesnelere kontrol elemanı veya denetim elemanı denmekte olup araç kutusu üzerinde bulunmaktadır (Resim 2.14). Projenize istediğiniz herhangi bir kontrol elemanını, üzerine çift tıklayarak ya da sürükleyip bırakarak ekleyebilirsiniz.



Resim 2.14: Visual Basic .NET Toolbox Araç Kutusu

2.8.6. Properties (Özellikler) Penceresi

Bu kısım programlarınız için bir, oldukça önemli bir anlam ifade etmektedir. Bilindiği üzere Visual Basic .NET dilinde program yaptığımız zaman programınıza ait form üzerinde bulunan bütün denetim elemanlarının tamamı (Label, Text, Command button, Liste kutusu vb.) Visual Basic .NET derleyicisi tarafından birer nesne olarak algılanır. Form üzerine yerleştirmiş olduğumuz nesnelerin ve formun özellikleri bu pencerede listelenir. **Propeties** genel bir penceredir (Resim 2.15).

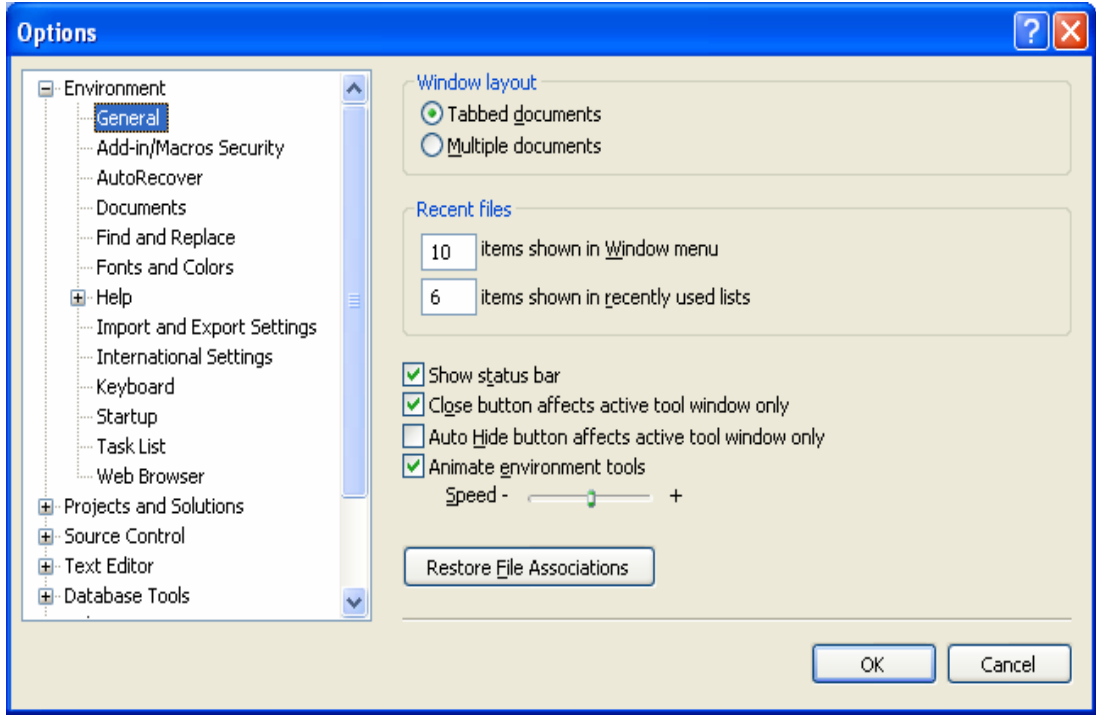


Resim 2.15: Visual Basic .NET Properties penceresi

Properties penceresi eğer ekranda yoksa F4 fonksiyon tuşu ya da **View** menüsünden **Properties Window** tıklanarak ekrana yerleştirilir.

2.9. “Tool *Options* “ Penceresi

Bu seçenekle üzerinde çalıştığımız pencerelerin birçok özelliğini değiştirebiliriz. Bu seçeneği seçtiğimizde aşağıdaki pencere görüntülenir (Resim 2.16).



Resim 2.16: Visual Basic .NET Tools/Options penceresi

Bu pencerede bulunan bazı seçenekler ve görevleri aşağıda açıklanmıştır.

➤ Environment:

• General:

- **Tabbed documents:** Açık olan sayfaları çoklu sayfa biçiminde gösterir.



Resim 2.17: Visual Basic .NET sayfa görüntüleri

- **Multiple documents:** Sayfaların her biri ayrı bir pencerede açılır (Resim2.18).



Resim 2.18: Visual Basic .NET sayfa görüntüleri

- **Recent Files:** File menüsünde Recent Files/Projects komutu altındaki son kullanılan dosya/projelerden kaç tanesinin sıralanacağını belirtir.
- **Show Status Bar:** Durum çubuğunun ekranda görünüp görünmemesini sağlar.
- **Add-in/MACros Security:**

Makroların çalışmasına, bileşenlerin bir URL adresinden yüklenmesine izin verilip verilmeyeceğini ayarlar.
- **AutoRecover:**

Visual Basic .NET dosyalar üzerinde çalışırken belirtilen süre dolunca otomatik olarak işlemlerinizi kaydetmeye ve otomatik olarak kaydedilen işlemleri belirtilen süre zarfında hafızada tutmayla ilgili işlemlerin ayarlanmasını sağlar.
- **Documents:**

Sayfanın doğru yüklenip yüklenmediğinin kontrolünün yapılması ile ilgili ayarların yapıldığı yerdir.
- **Find and Replace:**

Bilgi ve uyarı mesajlarının gösterilmesi ile ilgili ayarların yapıldığı yerdir.
- **Fonts and Colors:**

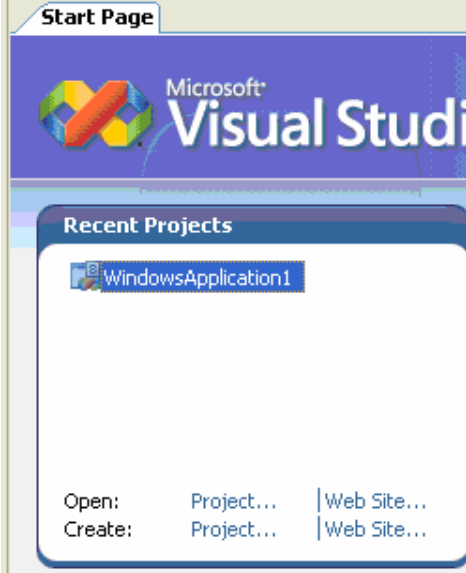
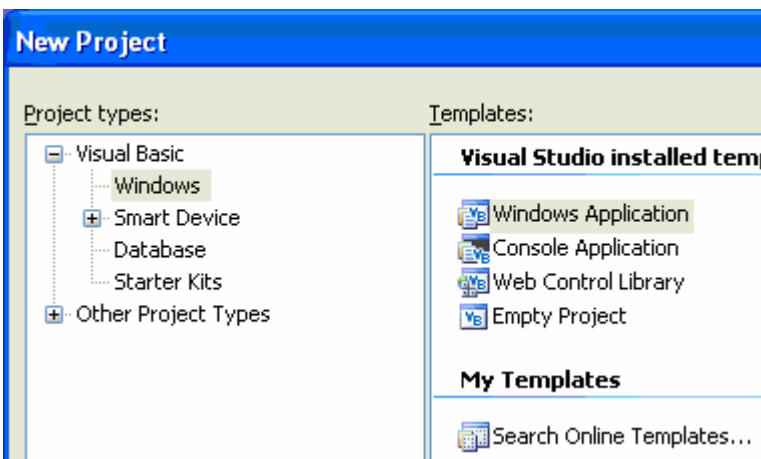
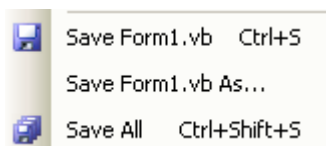
Kod penceresinde yazılan ifadelerin renkleri ve font büyüklükleri ayarlanır.
- **Help:**

Hazırladığınız dosyaların nereye kaydedileceğinin, Visual Basic .NET açıldığında başlangıç sayfasının hangi sayfa olacağını ayarlandığı yerdir. Ayrıca Visual Basic .NET tarafından kullanılan dili gösterir.
- **HTML Designer:**

Hazırlayacağımız HTML sayfalarının açılış sayfasının görsel sayfa mı kod sayfası mı olacağını ayarlar.
- **Windows Forms Designer:**

Form üzerindeki noktaların gösterilip gösterilmeyeceğinin ayarlanabildiği yerdir.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Yeni bir proje açınız.</p> 	<p>Create Project ile yapılmalıdır.</p>
<p>2. Projenin türünü seçiniz.</p> 	<p>Hazırlayacağınız programın kullanılacağı ortama göre seçilir.</p>
<p>3. Açık bir projeyi kaydediniz.</p> 	<p>Sağ buton menüsünü kullanabilirsiniz.</p>

ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki sorulardan; sonunda parantez olanlar doğru / yanlış sorularıdır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız. Şıklı sorularda uygun şıkkı işaretleyiniz.

1. Solution Explorer penceresini görünür hale getirmek istersek View (Görünüm) menüsünden Solution Explorer tıklanarak seçilir. ()
 2. Eğer yeni bir proje açmak istiyorsak **File** menüsündeki **Open Project** butonu tıklanır. ()
 3. Form üzerine yerleştirmiş olduğumuz nesnelerin ve formun özellikleri Properties penceresinde listelenir. ()
 4. **Windows Application:** İçerisinde form bulunan Standart Windows uygulamalarıdır.()
- Properties penceresini ekranda görüntülemek için hangi kısayol tuşu kullanılır?
A) F5
B) F3
C) F4
D) F8
- Solutions penceresini ekranda görüntülemek için hangi kısayol tuşu kullanılır?
A) Ctrl+Alt+L
B) F3
C) Ctrl+Alt+M
D) F5

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-3

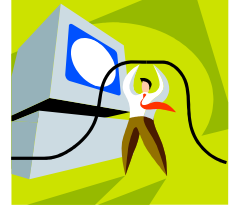
AMAÇ

Bu modül ile gerekli ortam sağlandığında, genel kontrol nesneleri ile çalışabilecek ve araç kutusunu özelleştirebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

- Daha önce kullanmış olduğunuz işletim sistemi windows uygulamalarında bulunan pencerelerdeki buton, işaret kutuları, açılan kutular vb. nesnelerin görevlerini araştırınız. Sonuçları rapor haline getirip arkadaşlarınız ile paylaşınız.

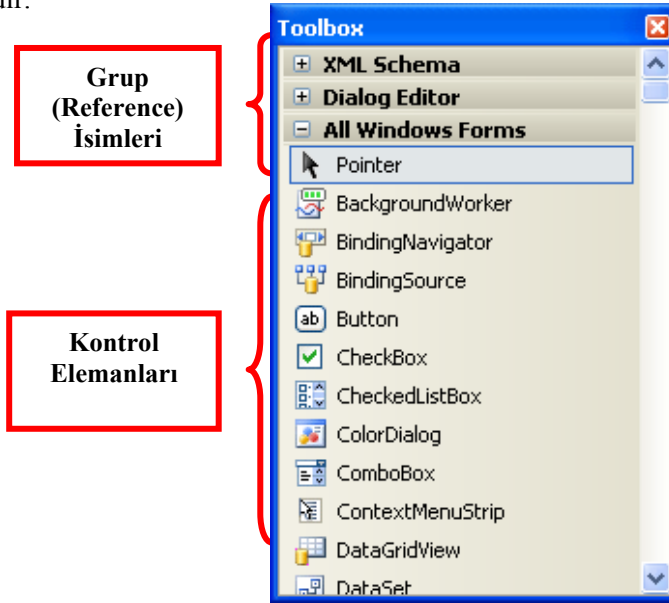


3. NESNELER

3.1. Araç Kutusu, Özellik Penceresinin Ayarları

3.1.1. Araç Kutusu Penceresi

Daha öncede belirtildiği gibi projeleriniz formlar ve bu formlar üzerinde bulunan nesne dediğimiz kontrollerden oluşmaktadır. Araç kutusu üzerinde bulunan bu kontroller referanslar altında bulunur. Bu kontroller araç kutusu üzerinde kategorize edilip listelenmektedir. Sırasıyla XML Schema, Dialog Editor, All Windows Forms şeklinde sıralanmaktadır.



Resim 3.1: Toolbox Araç Kutusu Penceresi

Kontrolleri, form üzerine yerleřtirmenizin 3 yolu vardır :

- Toolbox'taki kontrolü tıklayınız. Fare, form üzerine getirilince + iřareti belirecektir. Bu + iřareti kontrolün sol üst köřesini belirtmektedir. Farenin sol tuřuna basılı tutarak fareyi saę alt köřeye doęru hareket ettiriniz. Fareyi bırakınca, bir dikdörtgen oluřacaktır.
- Toolbox'taki kontrolü çift tıklayınız. Kontrol varsayılan büyüklükte formun sol üst köřesinde oluřur.
- Toolbox'taki kontrolü tıklayınız. Fare form üzerine getirilince + iřareti belirecektir. Bu + iřareti kontrolün sol üst köřesini belirtmektedir. Tekrar farenin sol tuřuna tıklanırsa, varsayılan büyüklükteki kontrol form üzerinde oluřacaktır.

Kontrolleri toplu olarak bir yerden başka bir yere taşımak, benzer özelliklerini toplu olarak deęiřtirmek (mesela rengini, fontunu gibi) veya hizalamak (sola, saęa vb. gibi) için form tasarımı sırasında bazen birden fazla kontrolle aynı anda çalışmanız gerekebilir.

➤ **Birden fazla kontrolü seçmek**

Birkaç yöntem vardır. Eęer kontroller birbirine yakınsa, kontrolleri fare ile çerçeve içine alıp seçebilirsiniz.

Bir başka yöntem de Ctrl veya Shift tuřuna basılı tutarak seçmek istedięiniz kontrollerin tıklanmasıdır.

Eęer kontrollerin hepsini seçmek istiyorsanız, kolayca Ctrl + A tuř kombinasyonunu kullanılabılırsınız.

➤ **Birden fazla kontrolün seçimini iptal etmek**

Çoklu kontrol seçimini bitirmek için, formun boş bir yerine (kontrol olmayan) tıklamanız yeterli olacaktır.

➤ **Kontrolleri grup olarak hareket ettirmek**

Çoklu kontrol seçimi yaptıktan sonra, bunları grup olarak hareket ettirmek için, fareyi seçilmiş kontrollerden birinin üzerinde bastırarak, kontrolleri sürükleyebilirsiniz.

➤ **Çoklu kontrolün özelliklerini deęiřtirmek**

Çoklu kontrol seçimi yapıldıktan sonra, saę taraftaki Properties penceresinden istenilen özellikleri deęiřtirilebilirsiniz. Örneęin BackColor özellięi deęiřtirilirse, bu tüm gruptaki kontrollere uygulanır. Bazı özellikler boş gözükecektir. Bu özellikler grup olarak deęiřtirilemeyecek anlamına gelir. Örneęin Name ve Text özellikleri.

➤ **Kontrollerin yeniden boyutlandırılması ve hizalanması**

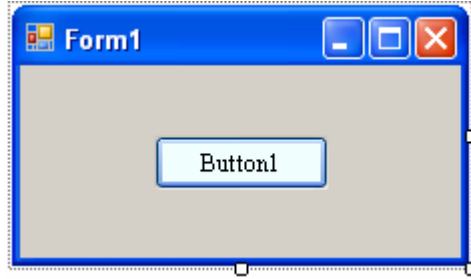
Grup kontrol seçimi yapıldıktan sonra yeniden boyutlandırılması ve hizalanması, Layout araç çubuęu ile çok kolaydır. Gruptaki kontrolleri bu araç çubuęundaki ikonlar yardımıyla, aynı genişlik (width) ve yükseklik (height)'te yapabilirsiniz. Kontrolleri yukarı, ařaęı, sola veya saęa hizalayabilirsiniz.

3.1.1.1. Toolbox'ta Bulunan Bazı Kontrol Elemanları

Toolbox'ta (Araç Kutusu) bulunan ve projelerinizi hazırlarken kullanacağınız kontrol elemanlarından bazılarını tanıyalım.

- **Button kontrolü**  Button

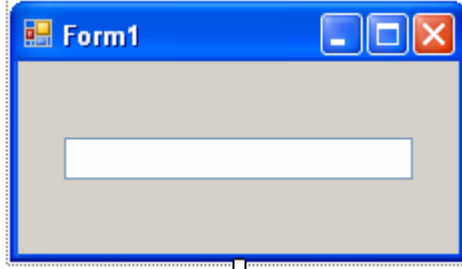
Button kontrolü, bir işlemin başlatılabilmesi için kullanılan kontroldür (Resim 3.2). Örneğin “Tamam” veya “İptal” vb.




Resim 3.2: Button kontrolünün formdaki görüntüsü

- **TextBox kontrolü**  TextBox

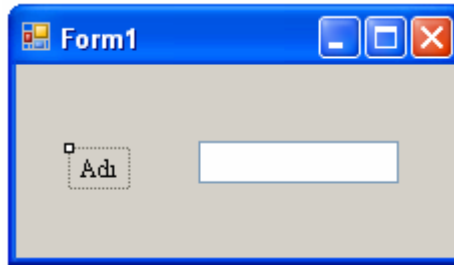
TextBox kontrolü, form üzerinde tasarım sırasında, programın çalışma anında girilen veya kod içinde atanan bilginin gösterilmesi için kullanılır. Gösterilen bilgi değiştirilebilir.



Resim 3.3: TextBox kontrolünün formdaki görüntüsü

- **Label (Etiket) kontrolü**  Label

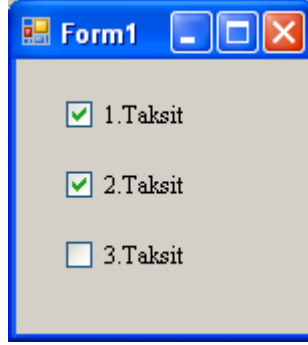
Label kontrolü, form üzerinde bilginin gösterilmesi için kullanılır (Resim 3.4).



Resim 3.4: Label kontrolünün formdaki görüntüsü

➤ **CheckBox** CheckBox ,



Windows' ta çok kullanılan kontrollerden biri de kullanıcının belirli özellikleri aktif veya pasif hale getirmek için kullanıldığı eleman CheckBox' lardır. Anlaşırılığı ve kullanımı kolay olması sebebiyle kolay kullanılabilir arabirimler oluşturmak için oldukça faydalı bir kontroldür (Resim 3.5).



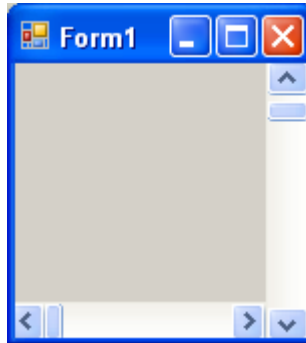
Resim 3.5: CheckBox kontrolünün formdaki görüntüsü

➤ **Timer (Zamanlayıcı) kontrolü**  Timer

Programda belirli bir süre aralıklar ile aktif hale gelip belirli işleri yapabilmek için kullanılan bir kontroldür. Bu kontrol, tasarım zamanda ekranda görülmesine rağmen çalışma esnasında görülmez.

➤ **Horizontal & Vectiral ScrollBar (Kaydırma Çubuğu)**  HScrollBar -  VScrollBar

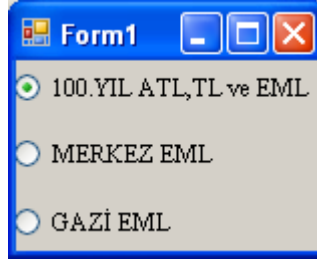
Biri yatay diğeri dikey olmak üzere her iki kaydırma çubuğu da aynı özelliklere sahiptir ve kullanım amacına göre amaçları da değişir. Örneğin bir veri tabanında kayıtlar arasında ileri geri gitmek için kullanılabilir (Resim 3.6).



Resim 3.6: Horizontal & Vectiral ScrollBar (Kaydırma Çubuğu) kontrollerinin formdaki görüntüsü

➤ **RadioButton (Seenek Dğmesi) kontrol**  RadioButton

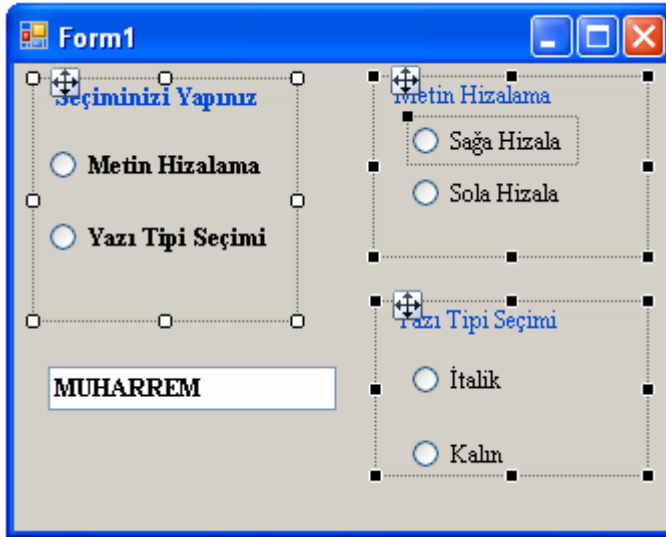
RadioButton kontrol CheckBox kontrolnden farklı olarak birkaç seenekten sadece birini seme imkânı veren bir kontroldür. Bu kontroln tek başına kullanılması anlamsızdır. Birkaç seenekten birini seme imkânı veren bir kontrol olduėu için en az iki tane birlikte kullanılmalıdır. Gruptaki RadioButton dğmelerinden biri seildiėinde diėeri kendiliėinden seilmiş özelliėini kaldırır. Yani aynı anda bir grupta iki tane iřaretli dğme bulunmaz. Yalnızca bir seenek seebiliriz ve bilgisayarımız da o seeneėe gre hareket eder (Resim 3.7).



Resim 3.7: RadioButton (Seenek Dğmesi) kontrolnn formdaki grnts

➤ **GroupBox (Grup Kutusu) kontrol**  GroupBox

Bu kontrol tek başına deėil, diėer kontrolleri gruplayabilmeniz için kullanılır. Kontrolleri bu kontrolle gruplamanızın birok avantajı vardır. Bu ereveler iine konan kontroller, ereveye baėımlıdır ve konuları bu ereve dıřına ıkamaz. Özellikle birkaç kontrol birden grnr veya grnmez yapmak için hepsinin Visible özelliėini deėiřtirmek yerine erevenin Visible özelliėini deėiřtirerek aynı iřlemi bir adımda yapmış oluruz. Aynı şekilde tařınma özelliėinde de ereve tařınarak iřlem tek adımda gerekleřtirilebilir. Yani kullandığımız GroupBox kontroln form ierisindeki bařka bir form gibi dřenebiliriz.



Resim 3.8: GroupBox (Grup Kutusu) kontrolnn formdaki grnts

GroupBox' larla gruplanan kontrollerin koordinatları artık forma göre değil, grup kontrolünün sol üst köşesine göre belirlenir. Form üzerine yerleştirilmiş bir kontrolü taşıyarak bir GroupBox kontrolü üzerine getirmekle o kontrol gruplanmış olmaz. GroupBox kontrolü içerisine bir kontrol yerleştirirken önce GroupBox kutusunu seçiniz. Eğer Form üzerinde bulunan kontrolleri GroupBox kontrolü içerisine almak istiyorsanız o zaman Form üzerindeki kontrolleri kesiniz (Cut) ve Frame kontrolünü seçtikten sonra buraya yapıştırınız (Paste). Ayrıca Frame, kontrolleri sadece bazı şartlarda gösterilmesi gereken kontrolleri de bir arada tutarak bunların kolayca gizlenip gösterilmesini sağlar.

➤ **PictureBox (Resim Kutusu) Kontrolü**

Bu kontrol elemanı Bitmap, Icon, Metafile, Jpeg ve Gif gibi resimleri görüntülemek için kullanılır. Ayrıca metotlar kullanılarak PictureBox içine çizimlerde yapılabilmektedir. Bu kontrolün bir diğer özelliği ise aynen GroupBox kontrolünde olduğu gibi diğer kontrolleri gruplandırmasıdır. Kod penceresine aşağıdaki gibi bir komut satırı yazabilirsiniz.

```
Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal  
    PictureBox1.Picture = LoadPicture("E:\Resimler\Tanyildizi.Bmp")  
End Sub
```

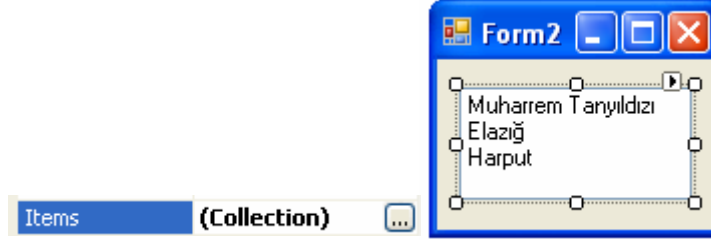
➤ **Form**

Windows arabiriminin en temel kontrolü formlardır. Windows'ta hemen her program formlar üzerinde çalışır. Zaten Windows kelimesinin Türkçe anlamı Pencere de bu formlardır. Boyutlandırılabilir özelliği sayesinde aynı ekranda tek bir program olmak zorunda değildir. Formun Properties kısmını formun alt programlarında yazarken Formun ismini kullanmak zorunda değilsiniz, direk Properties ismini vermeniz yeterlidir. Yani Form1.Name ile sadece Name, Form1' in alt programlarında aynı etkiye sahiptir.

Programınızda birden fazla form bulunacaksa ilk olarak ana form çalışacaktır. Diğer formları programınızda kullanacağınız Form2.Show gibi bir yöntemle aktif hale getirmelisiniz. Bir formdan başka bir formun bir özelliğine ulaşabilmek için kontrol adından önce formun adı da verilmelidir. Örneğin Form2 üzerindeki Label1'e ulaşmak için Form2.Label1.Özellik şeklinde kullanılır. Programınızda birden fazla formunuz varsa ilk oluşturduğunuz form ana formdur ve program çalışmaya o form ile başlar. Eğer başlangıçta çalışacak formu değiştirmek istiyorsanız; Solution penceresindeki projemize sağ tıklayıp Properties komutu seçilir. Daha sonra Startup Form listesinden istediğiniz formu seçebilirsiniz.

➤ **ListBox (Listeleme Kutusu) kontrolü**

Visual Basic' in sağladığı dizilerinizi gösterebileceğiniz kontrollerdendir. Elemanları listelemek, sıralamak amacı ile kullanılan bir kontroldür. Listeye eleman eklemek için Properties penceresindeki Items özelliği kullanılabilir.



Resim 3.9: ListBox kontrolünün formdaki görüntüsü

➤ **ComboBox (Açılan Liste) kontrolü**  ComboBox

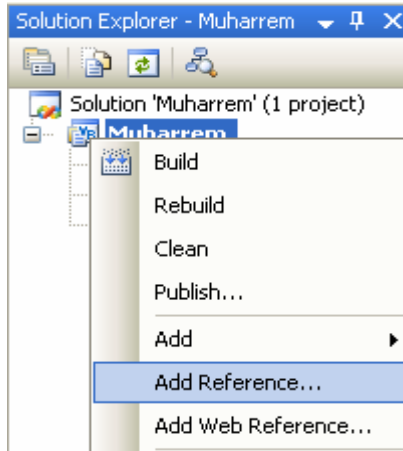
Genellikle değerleri daha önceden belli olan elemanların seçimi için kullanılır. Liste kutusuna benzer ancak listedeki elemanlardan sadece seçileni ekranda görüntülenir. Aşağı doğru açılabilen bir liste kontrolüdür. ComboBox kontrolüne eleman ekleme ve silme işlemi ListBox' ta olduğu gibidir.



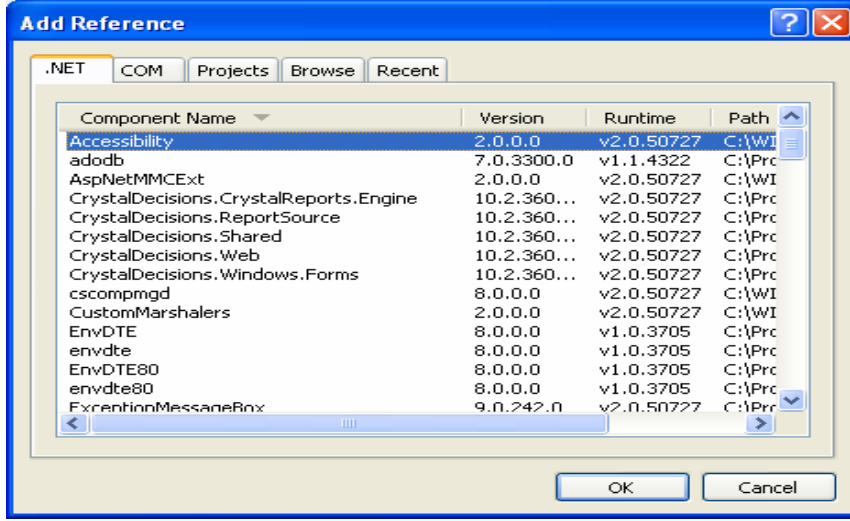
Resim 3.10: ComboBox (Açılan Liste) kontrolünün formdaki görüntüsü

3.1.1.2. Araç Kutusu Penceresinin Ayarları

Çalıştığınız projeye **ToolBox**' ta bulunmayan bir referans eklemek isterseniz Solution Explorer penceresinde **References** seçeneğine sağ tıklayarak **Add Reference** komutunu verebilirsiniz (Resim 3.11).



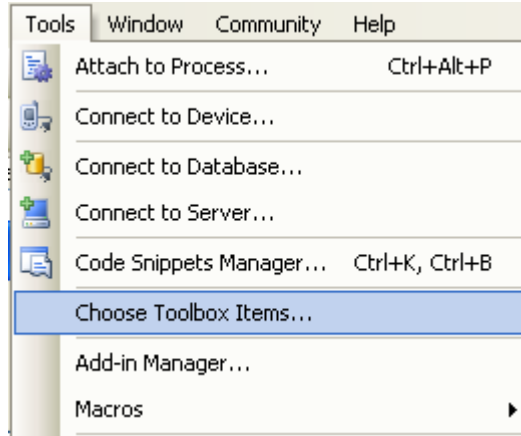
Resim 3.11: Add Reference komutunun görüntüsü



Resim 3.12: Add Reference penceresinin görüntüsü

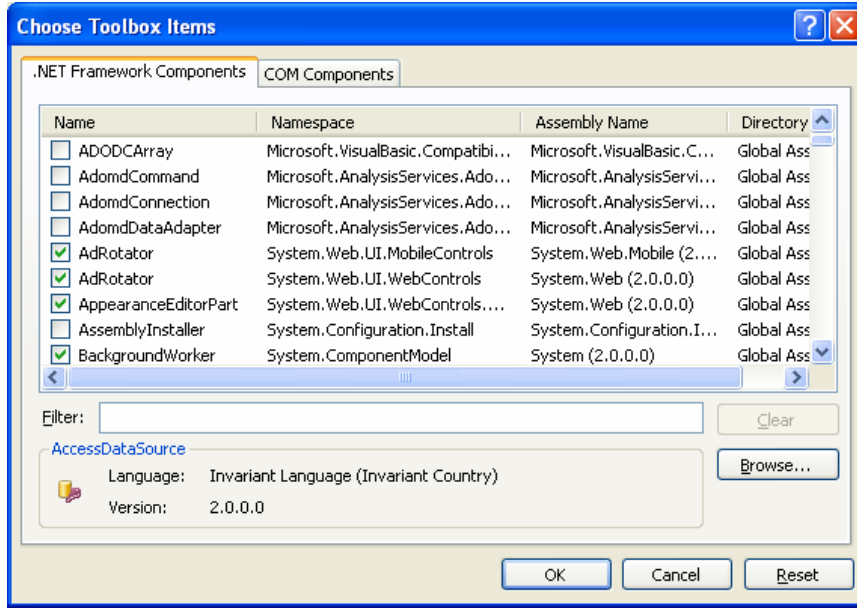
Projede kullanmak istediğiniz bileşeni seçip OK düğmesini tıklamanız yeterli olacaktır. Bu seçtiğiniz bileşende Solution Explorer penceresinin References bölümüne aktarılır.

Eğer projenizde kullanmak istediğiniz bir kontrolör Toolbox'ta mevcut değilse **Tools** menüsünden **Choose Toolbox Items** komutunu verebilirsiniz.



Resim 3.13: Tools/Choose Toolbox Items komutunun görüntüsü

Bu komutu verdikten sonra **Choose Toolbox Items** diyalog kutusu ekran gelir (Resim3.14).

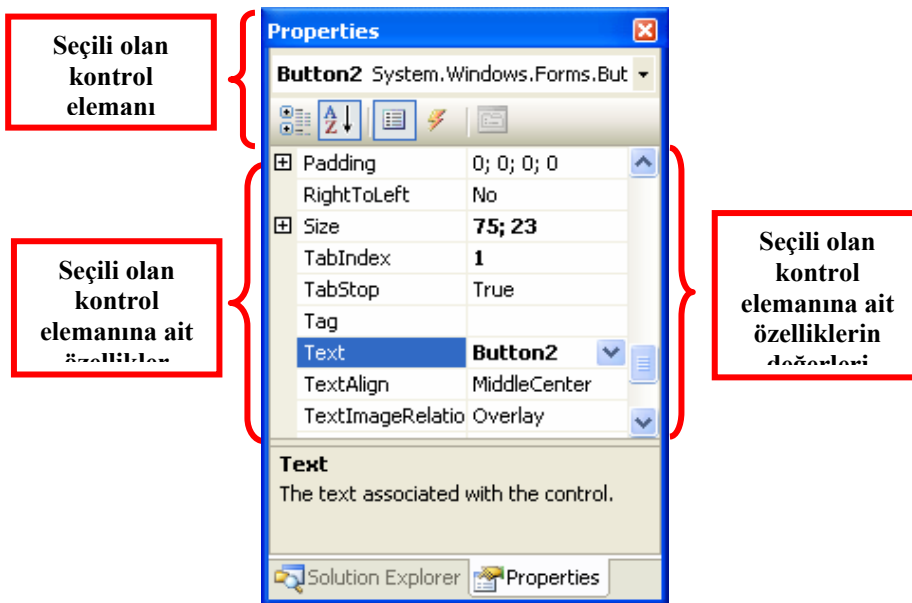


Resim 3.14: Choose Toolbox Items penceresinin görüntüsü





Bu diyalog kutusunda Toolbox menüsündeki mevcut kontroller işaretli mevcut olmayan kontroller ise işaretsiz olacaktır.

3.1.2. Özellikler Penceresinin Ayarları


Yeni bir proje oluşturduğumuzda Visual Basic.NET Form1 adında bir ana formu otomatik olarak oluşturur ve bu forma ait özellikler **Properties (Özellikler)** adlı bir pencerede bulunur (Resim 3.15). Properties penceresinde 4 tane ikon bulunur.

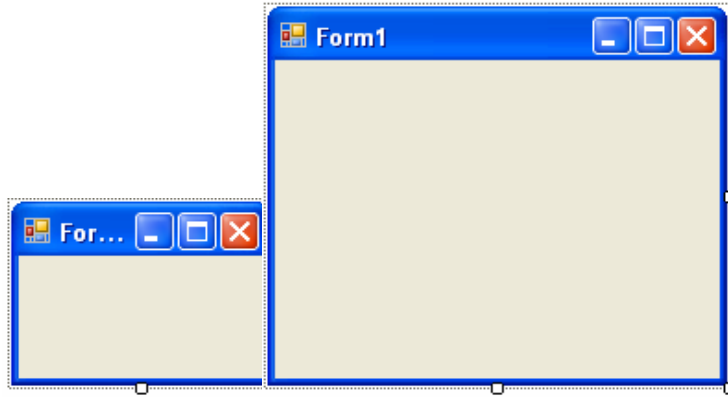


Resim 3.15: Visual Basic .NET Properties (Özellikler) penceresinin görünümü

-  : Özellikleri gruplamak için kullanılır.
-  : Özelliklerin alfabetik sıraya göre listelenmesi için kullanılır.
-  : Özelliklerin listelenmesi için kullanılır.
-  : Properties (Özellikler) penceresinde özellikleri listelenen nesne için tanımlı olan olayları listelemek için kullanılır.

3.2. Form Tasarımındaki Nesne Boyutlandırma Tutamacı

Visual Basic .NET'te kullanılan Windows Formların boyutlarını iki değişik yolla ayarlayabilirsiniz. Birincisi; forma bir kez tıkladıktan sonra etrafında beliren boyutlandırma tutamaçlarından tutarak ikincisi ise; Properties (Özellikler) penceresindeki  Size **328; 278** Size (Genişlik) seçeneğinden istediğiniz boyutu yazarak formun ebatlarını ayarlayabilirsiniz (Resim 3.16).



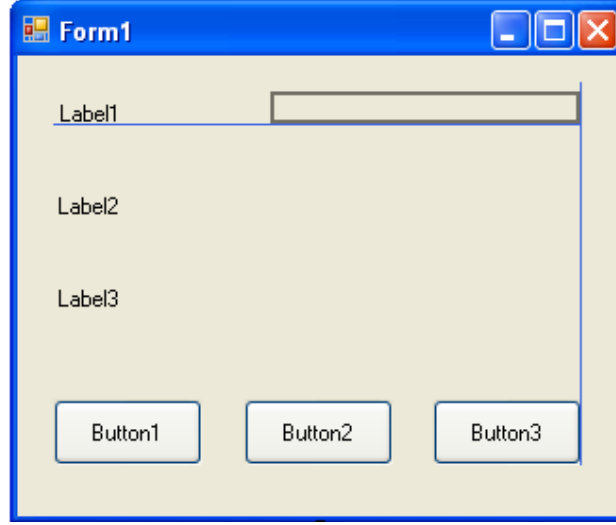
Resim 3.16: Windows formunun değişik ebatlardaki görünümü

Visual Basic .NET'te formun boyutlarının yanı sıra, forma yerleştirilen kontrol elemanlarının da boyutlarını ayarlayabilirsiniz.

Oluşturulmuş bir kontrolü taşıyabilmeniz için, fare kontrolün ortasındaiken dörtlü ok işareti çıktığında, farenin sol tuşuna basılı tutarak hareket ettirmeniz gerekmektedir. Kontrol istediğiniz yere gelince fareyi bırakırsınız. Forma yerleştirilecek olan kontrol elemanlarını grid çizgilerine gerek duymadan belli bir düzende yerleştirilebilirsiniz (Resim 3.17, 3.18).



Resim 3.17: Button kontrollerinin forma yerleştirilmesine ait görüntü



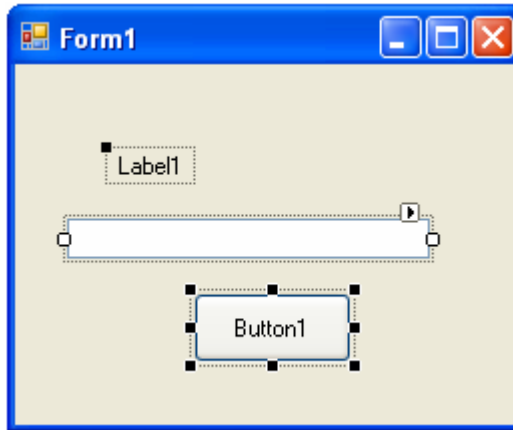
Resim 3.18: Değişik kontrollerin forma yerleştirilmesine ait görüntü

3.3. Yerleşim Araç Çubuğu

Yerleşim Araç Çubuğu (Layout), form üzerine yerleştirilen kontrol elemanlarının belli bir düzende yerleştirilmesini sağlar. Eğer, Yerleşim Araç Çubuğu (Layout) ekranda görünmüyorsa **View** menüsünden **Toolbars** üzerine gelip ve oradan Layout'u işaretleyebilirsiniz.




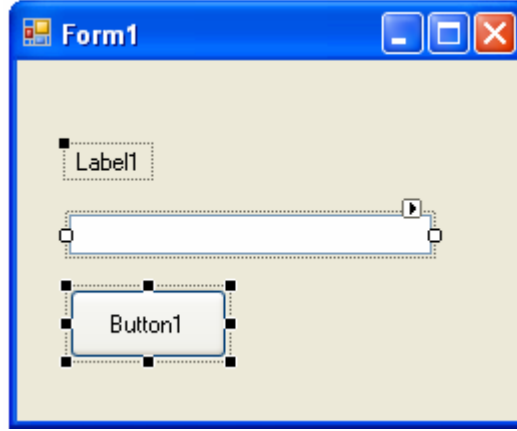
Resim 3.19: Visual Basic .NET Layout (Yerleşim) araç çubuğunun görünümü







Resim 3.20: Visual Basic .NET'te form görünümü

Form üzerine rastgele yerleştirilmiş olan kontrol elemanlarını Yerleşim Araç Çubuğu (Layout) ile düzenlerken öncelikle bir elemanı referans olarak almanız gerekmektedir. Daha sonra klavyeden Ctrl tuşuna basılı tutarak diğer elemanları seçer ve Yerleşim Araç Çubuğu üzerindeki komutlardan istediğinizi uygularsınız.

-  **Alings Left:** Tüm kontrol elemanlarını form üzerinde sola dayalı olarak yerleştirir (Resim 3.21).




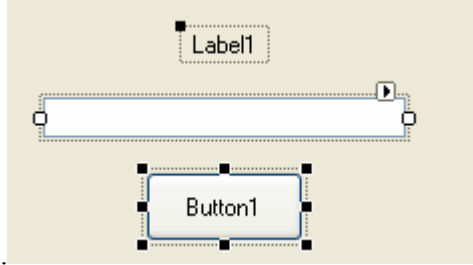
Resim 3.21: Align Left komutu ile sola dayalı yerleştirilen kontrol elemanlarının görünümü

-  **Align Right:** Tüm kontrol elemanlarını form üzerinde sağa dayalı olarak yerleştirir.
-  **Align Tops:** Referans alınan kontrol elemanının sol üst köşesine hizalı olarak elemanları yerleştirir.
-  **Align Bottoms:** Referans alınan kontrol elemanının sağ alt köşesine hizalı olarak elemanları yerleştirir.
-  **Align Middles:** Referans alınan kontrol elemanının orta noktasına hizalı olarak elemanları yerleştirir (Resim 3.23).






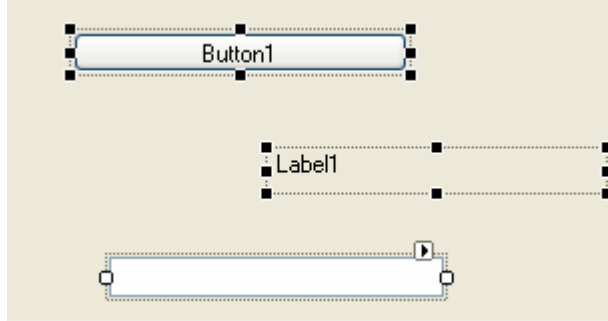
Resim 3.23: Align Middles komutu ile seçili kontrol elemanına ortalı yerleştirilen kontrol elemanlarının görünümü

-  **Alings Center:** Tüm kontrol elemanlarını form üzerinde ortalayarak yerleştirir (Resim 3.22).
















Resim 3.22: Align Center komutu ile forma ortali yerlestirilen kontrol elemanlarinin gorunumu

-  **Make Same Width:** Referans alınan kontrol elemanına göre diğer elemanların genişliğini ayarlar.
-  **Make Same Height:** Referans alınan kontrol elemanına göre diğer elemanların yüksekliğini ayarlar.
-  **Make Same Size:** Referans alınan kontrol elemanına göre diğer elemanların hem genişlik hem de yüksekliğini ayarlar (Resim 3.24).

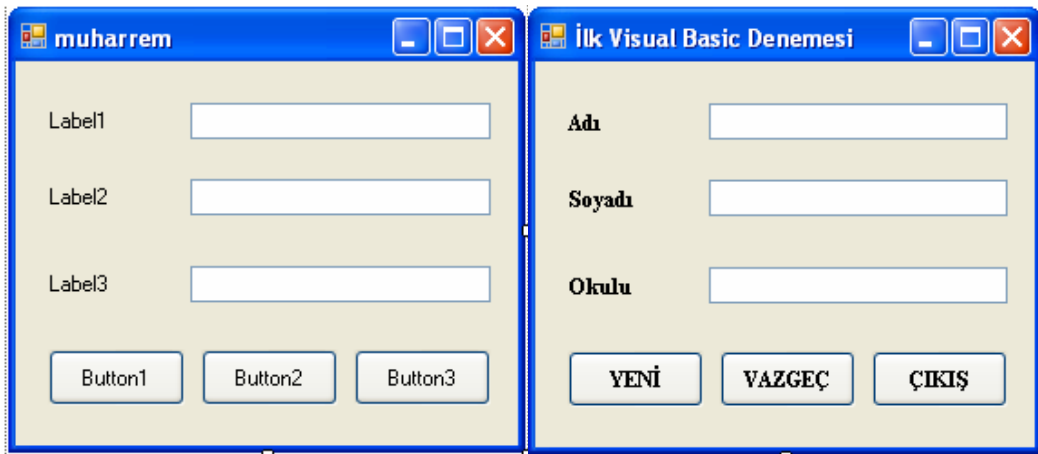


Resim 3.24: Make Same Size komutu uygulanmasına ait gorunum

-  **Make Horizontal Spacing Equal:** Kontrol elemanlarının birbirleri ile olan sağ ve sol boşluklarını eşit bir şekilde ayarlar.
-  **Increase Horizontal Spacing:** Kontrol elemanlarının birbirleri ile olan sağ ve sol boşluklarını eşit bir şekilde artırır.
-  **Decrease Horizontal Spacing:** Kontrol elemanlarının birbirleri ile olan sağ ve sol boşluklarını eşit bir şekilde azaltır.
-  **Remove Horizontal Spacing:** Kontrol elemanlarının birbirleri ile olan sağ ve sol boşluklarını kaldırır.

-  **Make Vertical Spacing Equal:** Kontrol elemanlarının birbirleri ile olan alt ve üst boşluklarını eşit bir şekilde ayarlar.
-  **Increase Vertical Spacing:** Kontrol elemanlarının birbirleri ile olan alt ve üst boşluklarını eşit bir şekilde artırır.
-  **Decrease Vertical Spacing:** Kontrol elemanlarının birbirleri ile olan alt ve üst boşluklarını eşit bir şekilde azaltır.
-  **Remove Vertical Spacing:** Kontrol elemanlarının birbirleri ile olan alt ve üst boşluklarını kaldırır.
-  **Center Horizontally:** Kontrol elemanlarını form üzerinde sağ ve sol boşlukları eşit olacak şekilde ayarlar.
-  **Center Vertically:** Kontrol elemanlarını form üzerinde alt ve üst boşlukları eşit olacak şekilde ayarlar.
-  **Bring to Front:** Seçili olan nesneyi en öne getirir.
-  **Send to Back:** Seçili olan nesneyi en arkaya gönderir.
-  **Tab Order:** Kontrol elemanlarının tab numaralarını değiştirmeye yarar.

3.4. Form ve Nesnelerin Ortak Özellikleri (name, text)



Resim 3.25: Form ve kontrol elemanlarının görünümü

Form üzerindeki deęişiklikleri Properties penceresinden gerçekleştireceęiz. Aşağıdaki işlemleri sırası ile uygulayalım.

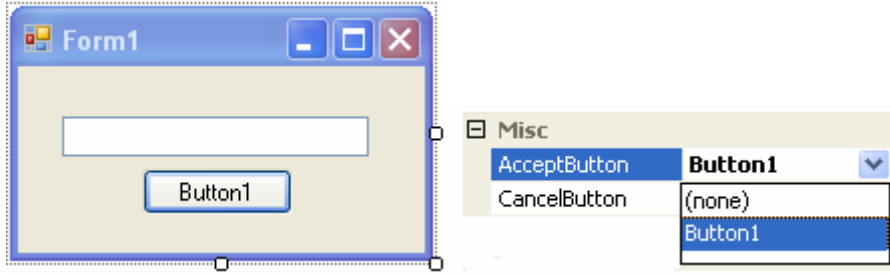
- Öncelikle Label1'i işaretleyip Properties kısmından Text özelliğini Adı olarak deęiştirelim. Aynı işlemi Label2 için Soyadı, Label3 için de Okulu olarak uygulayalım.
- Button1 denetimini işaretleyiniz ve Text özelliğini seçtikten sonra Button1 yazan kısma Yeni yazınız. Sonra aynı, işlemleri Button2 için Vazgeç, Button3 için de Çıkış olarak deęiştiriniz.
- Form' u işaretleyip Form' un Text özelliğini de "İlk Visual Basic Denemesi" olarak deęiştiriniz.

Şimdi bu aşamada elemanların rengi, üzerindeki yazıların renk, kalınlık, büyüklük ayarlamaları isimleri vb. özellikleri üzerinde oynama yapabilirsiniz. Şimdi çoęu eleman için aynı olan özelliklerden birkaçını açıklayalım.

- **Name:** Seçili kontrol elemanınızın program içerisindeki ismini belirler. (Name özelliğine göre elemana ait program kodları yazılmalıdır.)
- **Text:** Seçili kontrol elemanınıza program çalışırken kullanılacak ismi vermek için kullanılır.
- **BorderStyle:** Seçili elemanın form üzerinde 3 boyutlu görünümü sağlar.
- **BackColor:** Seçili elemanın zemin rengi seçimini sağlar.
- **Enabled:** Nesnenin aktif veya pasif olmasını sağlar.Yani üzerinde işlem yapılıp yapılmayacağını kontrol eder. False deęeri için nesne üzerinde işlem yapılmaz. True deęeri için nesneyi normal olarak çalıştırabilirsiniz.
- **Font:** Seçili nesne için yazı tipi seçimine olanak sağlar.
- **Size:** Nesnenin boyutu ile alakalı deęişiklik yapmamızı sağlar.
- **Location:** Nesnelerin koordinatını verir.
- **Visible:** Nesnenin görülüp görülmeme özelliğini ayarlar. False deęeri verildięi zaman nesne görülmez, True deęeri verildięi zaman görülür.

3.5. Form Özellikleri

- **AcceptButton-Enter:** Bu özelliğin aktif olabilmesi için öncelikle forma bir Button eklenmesi gerekmektedir.



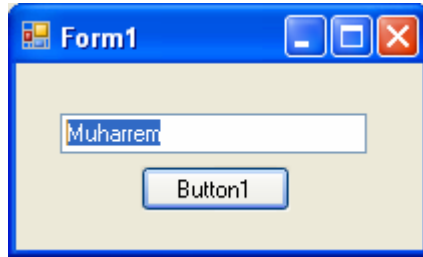
Resim 3.26: Form ve Accept Button komutunun görünümü

Button ekleyip formu seçince AcceptButton özelliğinde None ve seçili olan Button1 diye iki seçenek belirecektir. Eğer, Button1 seçeneğini aktif yaparsanız programınızı çalıştırdığımızda Button1'i Mouse ile tıklamak yerine Enter tuşu ile de kullanabilirsiniz.

Örnek olması açısından basit bir örnek verecek olursak, formunuza bir Button bir de TextBox ekleyiniz ve formun AcceptButton seçeneğini Button olarak ayarlayınız. Daha sonra Button'nun kod sayfasına geçip aşağıdaki program satırını yazınız.

```
Private Sub Button1_Click_1  
    TextBox1.Text = "Muharrem"  
End Sub
```

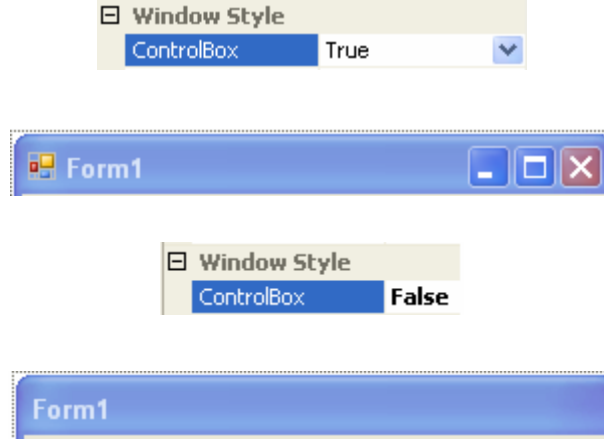
Şimdi programınızı çalıştırınız ve Button'a tıklamak yerine **Enter** tuşuna basınız.



Resim 3.27: Accept Button komutunun uygulanmasına ait görünüm

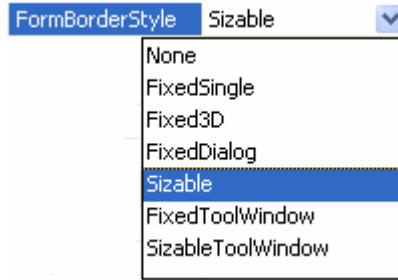
- **CancelButton-ESC:** Bu özellikte tıpkı AcceptButton gibidir. Tek farkı Button'a ESC tuşunu atmasıdır.

- **ControlBox**: Formların Properties (Özellikler) penceresindeki ControlBox özelliği “True” olarak belirlenmiştir. Ancak istenirse bu özellik “False” olarak değiştirilebilir. Eğer bu özelliği False yaparsanız formun sağ üst köşesinde yer alan simge durumuna küçültme, önceki boyut ve kapat düğmeleri görünmeyecektir (Resim 3.28).



Resim 3.28: ControlBox'ın uygulanmasına ait görünüm

- **FormBorderStyle**: Formunuzun ekran üzerindeki görünüşünü değiştirmek için Properties (Özellikler) penceresindeki **FormBorderStyle** kullanılır (Resim 3.29).



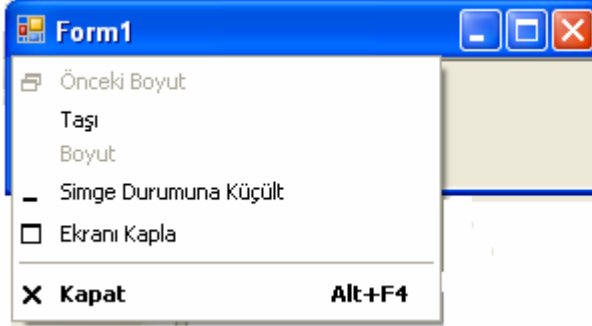
Resim 3.29: FormBorderStyle'in seçeneklerine ait görünüm

- **None** : Çerçeve yok (Resim 3.30)



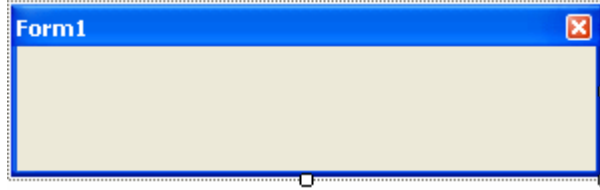
Resim 3.30: FormBorderStyle de None tercihinin uygulanmasına ait görünüm

- **FixedSingle:** Tek çizgili çerçeve
- **Fixed3D:** Üç boyutlu çerçeve
- **FixedDialog:** Formun sol üst köşesindeki diyalog kutusunu görünmez yapar (Resim 3.31).



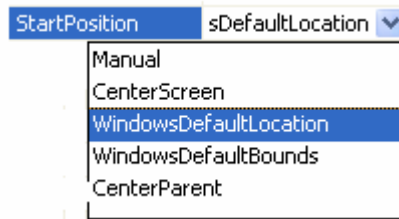
Resim 3.31: FromBorderStyle de FixedDialog tercihinin uygulanmasına ait görünüm

- **FixedToolWindow:** Formun sağ üst köşesindeki simge durumuna küçültme ve önceki boyut düğmelerini kaldırır (Resim 3.32).



Resim 3.32: FromBorderStyle de FixedToolWindow tercihinin uygulanmasına ait görünüm

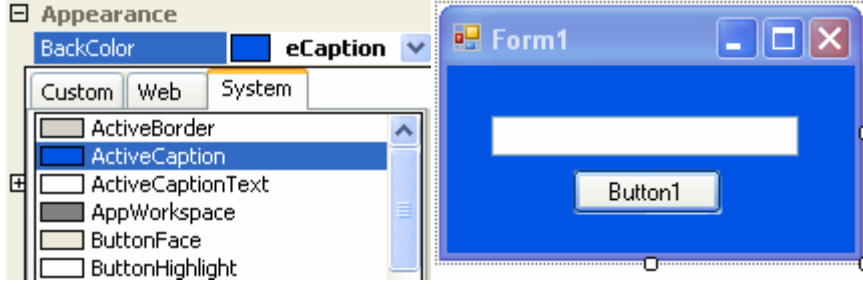
- **StartPosition:** Formların bir diğer özelliği olan “**StartPosition**”, programı çalıştırdığınız anda formun ekranın neresinde konumlanacağını belirlemenizi sağlar (Resim 3.33).



Resim 3.33: Start Position seçeneklerine ait görünüm

- **Manual:** Formu ekranın sol üst köşesine konumlandırır.
- **CenterScreen:** Formu ekranın tam ortasına konumlandırır.
- **WindowsDefaultBounds:** Formun boyutlarını büyütür.

- **BackColor:** Formun zemin rengini deęiřtirmeye yarar. Bu iřlemi yaparken açılan kutuya tıklayıp seçeneklerden herhangi birini seçebileceğiniz gibi, BackColor yazısının yanında bulunan renkli kutuya da tıklayarak istediğiniz rengi ayarlayabilirsiniz (Resim 3.34).



Resim 3.34: BackColor seçeneęi ve uygulanmasına ait görünüm

- **Enabled:** Formun aktif veya pasif olmasını sağlar. False yani pasif olması durumunda kullanıcı formu görür, ancak üzerinde işlem yapamaz.
- **Font:** Form üzerinde bulunan nesnelere için kullanılan yazının fontunu belirler.
- **Size:** Formun boyunu ve enini belirler. Bu iki deęer deęiřtirilerek formun boyutları ayarlanabilir.
- **ForeColor:** Form üzerinde bulunan nesnelere için kullanılan yazının rengini belirler.
- **RightToLeft:** Form başlığının ve nesnelere içindeki yazıların sağda ya da solda görünmesini sağlar.
- **RightToLeftLayout:** Eęer form RightToLeft seçeneęi "Yes" seçilip form başlığı sola alınmışsa, "True" aktif olur ve form başlığı ile formun kontrol düğmeleri yer deęiřtirir.

3.6. Nesnelerin Özellikleri

Visual Basic .NET'te nesnelere ait çeşitli özellikler vardır. Bu nesnelere biri de TextBox'tır. Bilgi girişleri TextBox nesnelere aracılığı ile yapılmaktadır. Toolbox penceresinde TextBox kontrolünü formunuza ekleyince Properties (Özellikler) penceresinde aşağıdaki özellikler bulunur.



Resim 3.34: Form ve Textbox kontrolüne ait görünüm

- **MultiLine:** True ise text kutusuna birden fazla satır girilebileceğini gösterir. False ise tek satır girilebilir.
- **Name:** Kontrol için istenilen isim yazılır. **Name** (İsmi) maksimum 40 karakter uzunluğunda olabilir. Name mutlaka bir harf ile başlamalıdır. Diğer karakterler harf, rakam veya _ (**alt çizgi**) olabilir.
- **TextAlign:** Nesne içerisindeki yazının sola, sağa veya ortaya yazılmasını sağlar.
- **ScrollBars:** Multiline özelliğinin True olması durumunda etkili olan bu özellik Text kutusu içerisinde bu özelliğin aşağı yukarı, sola sağa kaydırmak için kaydırma çubuklarının eklenmesini sağlar. None ise yok, Horizontal ise yatay, Vertical ise dikey ve Both ise hem yatay hem de dikey kaydırma çubukları eklenir.
- **MaxLength:** Text kutusuna girilebilecek maksimum karakter sayısını belirler. Buna 0 verilirse üst sınır 32 karakter olur.
- **PasswordChar:** Text kutusuna şifre girmek için kullanılır. Yani kullanıcı girdiği karakterlerin ekranda görülmesini istemiyorsa bu özellikten yararlanır. **PasswordChar** özelliğine karakter girerek kullanıcının girdiği bütün karakterlerin bu karakterle görülmesini sağlar.
- **Locked:** Text kutusunun bu özelliği True yapılırsa Text üzerinde hiçbir değişiklik yapılamaz.

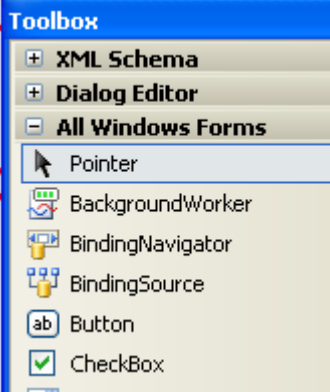
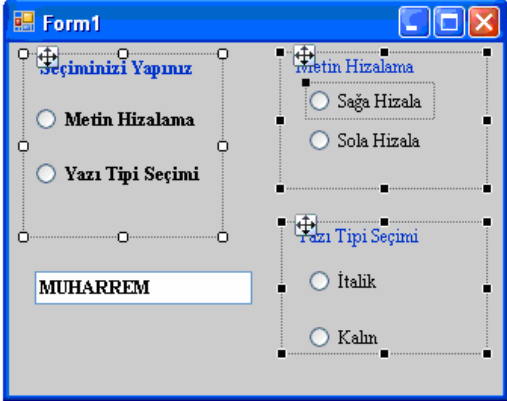
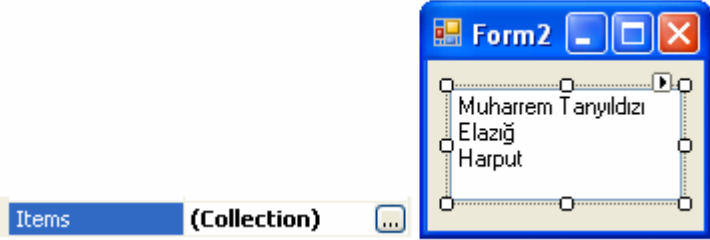
- **BorderStyle:** Nesnenin ekran üzerindeki sınırlarının çerçeve şeklini belirler.
 - **None :** Çerçeve yok
 - **FixedSingle:** Tek çizgili çerçeve
 - **Fixed3D:** Üç boyutlu çerçeve
- **Text :** Formun adını değiştirmeye yarar.
- **Enabled:** Nesnenin aktif veya pasif olmasını sağlar. False yani pasif olması durumunda kullanıcı nesneyi görür ancak üzerinde işlem yapamaz. Text kutusunun Enabled özelliği False ise kullanıcı giriş yapamaz.
- **Font:** Nesne için kullanılan yazının fontunu belirler.
- **ForeColor:** Yazının rengini belirler. Yukarıda anlatılan renk bileşimleri burada da geçerlidir.
- **BackColor:** Nesnelerin zemin rengini değiştirmeye yarar. Bu işlemi yaparken açılan kutuya tıklayıp seçeneklerden herhangi birini seçebileceğiniz gibi, BackColor yazısının yanında bulunan renkli kutuya da tıklayarak istediğiniz rengi ayarlayabilirsiniz.
- **Size:** Nesnenin boyunu ve enini belirler. Bu iki değer değiştirilerek kontrolün boyutları ayarlanabilir.
- **HelpButton:** Kontrolle ilgili yardım dosyasındaki konu numarası bu özellik ile belirlenir.
- **Icon:** Mouse göstergesinin nesne üzerine geldiğinde alacağı şekli belirler.
- **TabStop: True** ise kullanıcı bu nesneye Tab tuşuyla ulaşabilir. False ise Tab tuşuyla bu kontrol üzerine gelinmez, Mouse ile, yazılım yoluyla veya varsa kısayol tuşu ile gelebilir.

Yoğun bilgi girişi gerektiren formlarda sık kullanılmayacak kontrollerin TabStop özelliklerini False yaparsanız kullanıcının bilgi girişi daha kolay olacaktır. Çünkü genelde yoğun bilgi girişi olan yerlerde Tab tuşu sık kullanılır. Gereksiz yere ekrandaki bütün kontrolleri dolaşmamak için bazılarının bu özelliğini False yapmak gerekir.

- **TabIndex:** Form üzerinde kullanıcının ulaşabileceği her nesnenin bir TabIndex'i vardır. TabIndex kullanıcının Tab tuşuyla kontroller arasında dolaşırken bu kontrollerin sıralamasını belirler. Örneğin Tab indexi 5 olan bir nesne de iken Tab tuşuna basılırsa kontrol Tab indexi 6 olan nesneye geçer.

Normalde her yerleştirilen kontrole otomatik olarak bir Tab sırası verilir. Form tasarımınız bittikten sonra programı çalıştırıp Tab tuşuyla bütün kontrolleri gezerek Tab tuşunun sırayla gidip gitmediğini kontrol etmeniz gerekir. Çünkü araya yerleştirdiğiniz kontrollerin Tab sıraları bozulacaktır.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Form tasarım penceresine araç kutusundan bir nesneyi ekleyiniz. Sürükle bırak yöntemini kullanınız.</p> 	<p>Araç kutusunda bulunan nesnelere ve görevleri iyi bilinmelidir.</p>
<p>2. Aşağıda verilen form tasarımını yapınız.</p> 	<p>Araç kutusunda bulunan nesnelere ve görevleri iyi bilinmelidir.</p>
<p>3. Listbox kontrolüne liste elemanı ekleyiniz.</p> 	<p>Items özelliğini kullanabilirsiniz.</p>

ÖLÇME VE DEĞERLENDİRME





ÖLÇME SORULARI

Aşağıdaki sorulardan; sonunda parantez olanlar doğru / yanlış sorulardır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız. Şıklı sorularda uygun şıkkı işaretleyiniz.

1. Birden fazla kontrolü aynı anda seçebilmek için kullanılan yöntemlerden hangisi yanlıştır?

- A) Kontrolleri fare ile çerçeve içine alıp seçebilirsiniz.
B) Ctrl + A tuş kombinasyonu ile seçebilirsiniz.
C) Ctrl + Shift tuş kombinasyonu ile seçebilirsiniz.
D) Ctrl + Alt tuş kombinasyonu ile seçebilirsiniz.





2. Tüm kontrol elemanlarını form üzerinde sola dayalı olarak yerleştirebilmek için kullanılan komut düğmesi aşağıdakilerden hangisidir?

- A) 
B) 
C) 
D) 

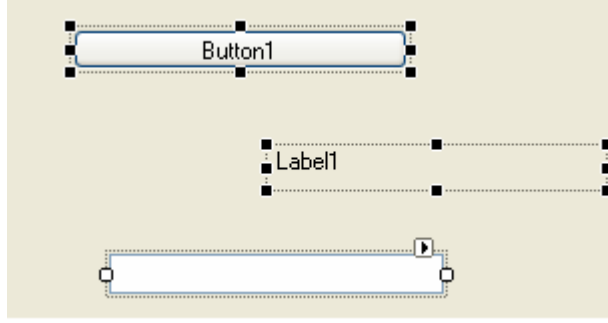
3. AcceptButton özelliğinin aktif olabilmesi için forma eklenmesi gereken kontrol nesnesi aşağıdakilerden hangisidir?

- A) Combo Box
B) Check Box
C) Button
D) Radio Button

4. Properties penceresindeki özellikleri gruplamak için kullanılan komut düğmesi aşağıdakilerden hangisidir?

- A) 
B) 
C) 
D) 

5. Aşağıdaki formda olduğu gibi referans alınan kontrol elemanına göre diğer elemanların hem genişlik hem de yüksekliğini ayarlamak için hangi komutu kullanırsınız?



- A) Make Horizontal Spacing Equal
- B) Make Same Width
- C) Make Same Height
- D) Make Same Size


DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığınız sorularla ilgili konuları tekrar tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

MODÜL DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki sorulardan; sonunda parantez olanlar doğru / yanlış sorulardır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız. Şıklı sorularda uygun şıklı işaretleyiniz.

1. Giriş sayfasındaki Recent Projects kısmında bulunan Open Project ile daha önce kaydedilmiş bir projeyi açabiliriz. ()
2. **View Menüsü:** Proje açmak, kapatmak, kaydetmek, yazdırmak gibi dosyalama işlemlerinin yapıldığı menüdür. ()
3. Tasarım Penceresine Solution Explorer penceresinde yer alan  ikonuna basarak geçilebilir. ()
4. Properties penceresinde bulunan Visible özelliği nesnenin görülüp görülmemesi özelliğini ayarlar. ()
5. Properties penceresinde bulunan BorderStyle özelliği nesnenin ekran üzerindeki yerini belirler. ()
6. Properties penceresinde bulunan Name özelliği, seçili kontrol elemanınıza program çalışırken kullanılacak ismi vermek için kullanılır. ()
7. Çalıştığımız projeye **ToolBox**’ ta bulunmayan bir referans eklemek istersek Solution Explorer penceresinde **References** seçeneğine sağ tıklayarak **Add Reference** komutunu verebilirsiniz. ()

PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığımız yeterliği aşağıdaki kriterlere göre değerlendiriniz.

DEĞERLENDİRME ÖLÇÜTLERİ	Evet	Hayır
Visual Studio .NET' bilgisayarımıza kurmak için gerekli olan sistem gerekliliklerini tespit ettiniz mi?		
Visual Studio .NET' bilgisayara kurabildiniz mi?		
Projelere Windows Form eklediniz mi*		
*.vbproj, *.vb ve *.sln dosya uzantılarını ayırt edebildiniz mi?		
Projeleri açabilme ve kaydedebildiniz mi?		
Solution Explorer penceresini açabilme ve kullanabildiniz mi?		
Properties Penceresini açabilme ve değerlerini değiştirebildiniz mi?		
Kod penceresine geçebildiniz mi*		
Genel olarak Visual Basic .NET'in menüleri tanıyabildiniz mi?		
ToolBox penceresindeki sık kullanılan kontrolleri kullandınız mı?		
Tools / Options penceresi ile genel ayarları yapabildiniz mi?		
Araç kutusundan kontrol ekleme yöntemlerini bildiniz mi?		
Birden çok kontrolü seçebildiniz mi?		
Form ve nesnelere boyutlandırabildiniz mi?		
Layout araç çubuğu ile nesnelere form üzerindeki yerleşimlerini düzenleyebildiniz mi?		
Form ve nesnelere ortak özellikleri bilme ve değiştirdiniz mi?		
Butonların Accept Button ve Cancel Button özelliklerini kullanabildiniz mi?		

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Programlamada önemli bir konu olduğundan belli zamanlarda bu modülü tekrar gözden geçiriniz.

Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır. Öğretmeninizle iletişime geçiniz.



CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	D
2	Y
3	Y
4	D
5	Y
6	Y
7	D
8	D
9	D
10	B

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	D
2	Y
3	D
4	D
5	C
6	A

ÖĞRENME FAALİYETİ-3 CEVAP ANAHTARI

1	D
2	B
3	C
4	A
5	D

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	D
2	Y
3	Y
4	D
5	Y
6	Y
7	D

KAYNAKÇA

- HAYDAR Yalçın, **Adım Adım Microsoft Visual Basic .NET**, Arkadaş Yayınları, Ankara, 2002.
- HALVORSON Michael, **Microsoft Visual Basic .NET Step By Step**, Microsoft Pres, A Divicion of Microsoft Corporation One Microsoft Way Redmond 2002.
- <http://www.ceturk.com/default.asp>
- <http://www.msakademik.net/makaleler.aspx?grup=VBN>
- <http://www.mutasyon.net/dersoku.asp?id=86>
- <http://www.bmssoftware.net/programlama/vbnet/vbnet02.aspx>
- Mastering, Visual Basic .NET
Evangelos Petroustos, SYBEX, Inc., Alameda, CA, 2002