



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
NESNEYE DAYALI PROGRAMLAMA DERSİ
1.ÖDEV DOKÜMANI

Adı: Ahmet Armutcu

Numara:G171210351

Sınıf(1C)

İçindekiler

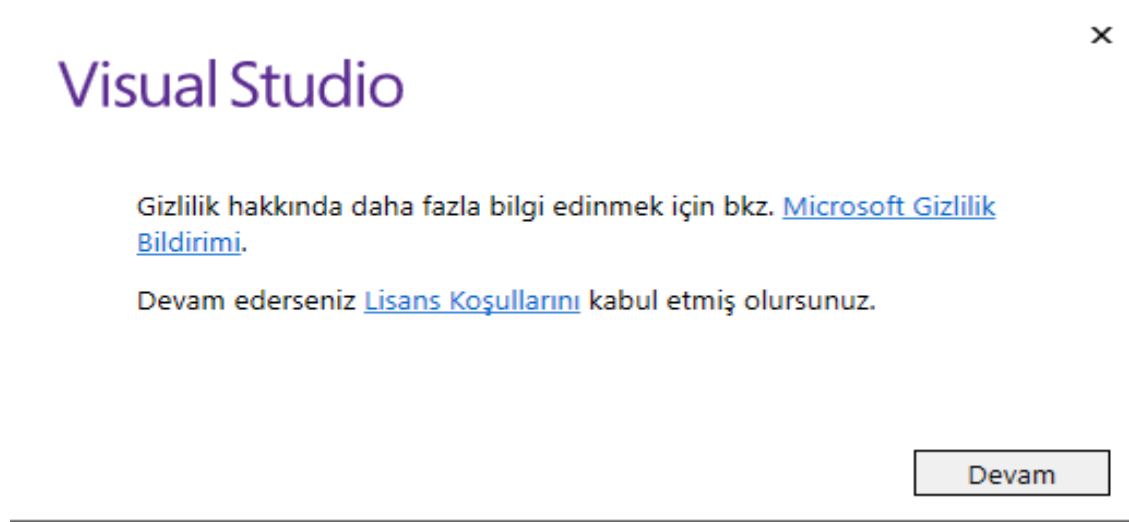
1. Microsoft Visual Studio 2017 Community Kurulumu	2
3.Aşama: Uygulama yüklendikten sonra Başlat düğmesiyle Visual Studio başarıyla çalıştırıyoruz.	3
2. Alternatif Derleyiciler ve Genel özellikleri	3
2.0 Derleyici(compiler) Nedir?	3
2.1 Microsoft Visual Studio	4
2.2 Netbeans	4
2.3 Javac	5
2.4 Komut Satırı Derleyicisi (csc.exe).....	5
2.5 C++ için Derleyici Çeşitleri ve Özellikleri.....	5
2.5.1 Dev-C++,	5
2.5.2 Intel Compiler	5
2.6 Digital Mars C/C++ Compiler	6
2.7 .Net Uyumlu Derleyiciler	6
3. Visual Studio Kullanımı ve Menü seçenekleri.....	7
3.1 Proje çeşitleri.....	8
3.2 UYGULAMA GELİŞTİRME ARAYÜZÜ.....	9
3.3 İlk Programımız.....	11
4. C,C++,C#, Java Veri Tipleri Tablosu	14
4.1 TEMEL VERİ TİPLERİ	16
4.1.1 C/C++	16
4.1.2 C#.....	17
4.1.3 JAVA.....	17
5. KAYNAKÇA	19

1. Microsoft Visual Studio 2017 Community Kurulumu

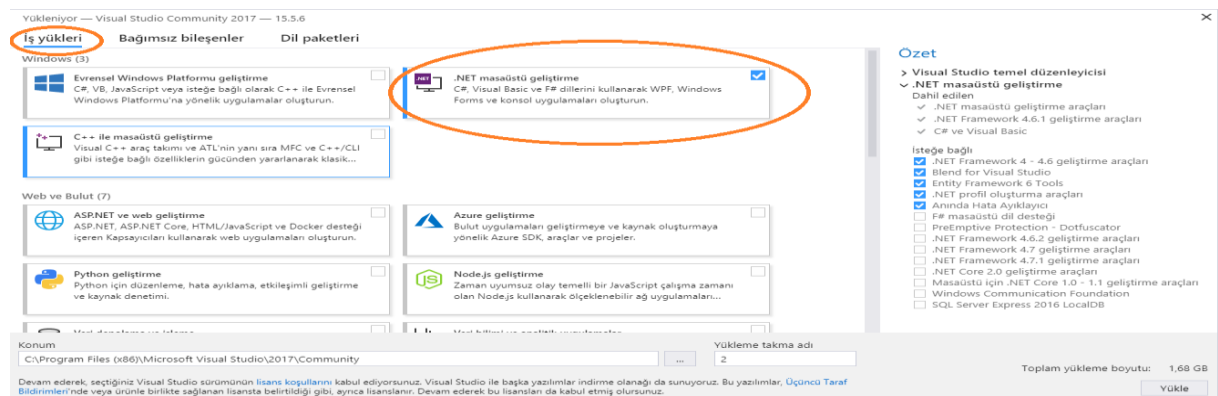
<https://www.visualstudio.com/tr/free-developer-offers/> adresine giriş yapalım ve buradan visual Studio 2017 Community indirme kısmına tıklayalım.

Kurulum Aşamaları

1.Aşama: İndirdiğiniz exe dosyasını açalım. Buradan devam tuşu ile ilerleyelim.

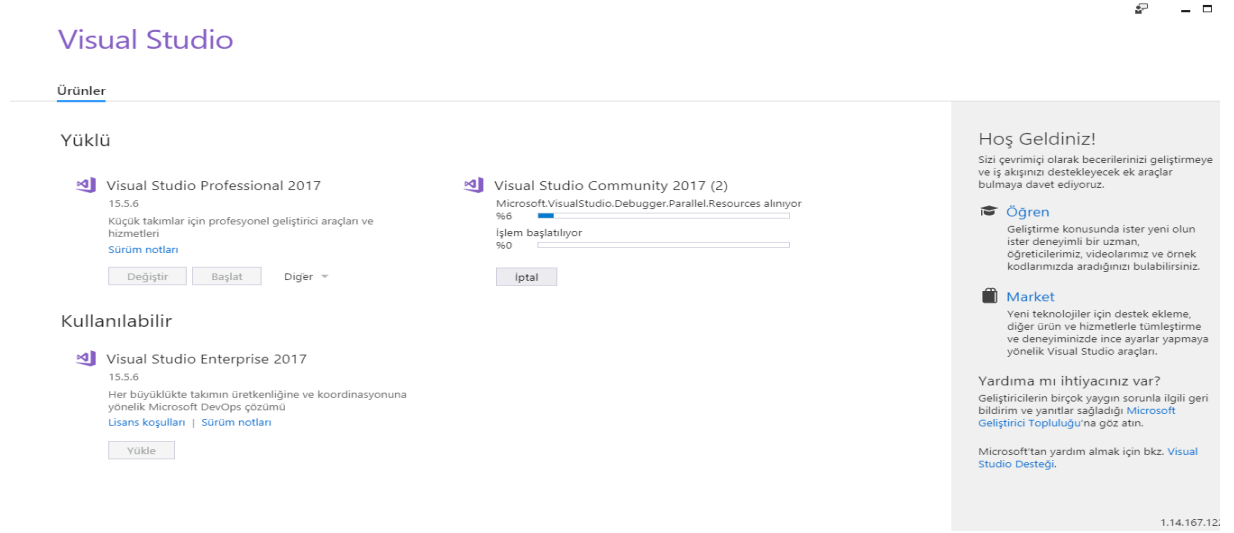


2.Aşama: İş yükleri sekmesinden yüklemek istediğimiz uygulamaya seçiyoruz. Windows için C#,C++,Visual Basic vb. seçenekler vardır. Web için ASP.NET,Azuru,Pyhton vb. geliştiriciler vardır. Mobil için Unity, C++ oyun geliştirme vb. programlama dilleri vardır. Buradan istediğim programlama dillerini seçebiliyoruz. Biz ise(C#) olduğu için Windows kısmında. NET masaüstü geliştirme seçiyoruz.



Bağımsız bileşenler sekmesinde yüklemek istediğimiz. Net Framework 'leri seçiyoruz. Dil Paketleri sekmesinde programa dilinin hangisini olacağını seçiyoruz. Konum kısmından Visual Studio nereye kurulacağını seçiyoruz. Tüm bu ayarlamaları yaptıktan sonra sağ

tarafıta özet kısmında yüklemek istediğimiz uygulamalar gözüküyor. Yükle tuşu ile ilerleyelim. Visual Studio kurulumunu başlatmış olduk, kurulum işleminin bitmesini bekleyelim.



3.Aşama: Uygulama yüklendikten sonra Başlat düğmesiyle Visual Studio başarıyla çalıştırıyoruz.

2. Alternatif Derleyiciler ve Genel özellikleri

2.0 Derleyici(compiler) Nedir?

Basitçe bir dilde yazılmış olan kodu (kaynak kodu ya da source code) istenilen başka bir kod haline dönüştüren programdır. Genelde üretilen bu kod ortama göre çalıştırılabilir kod (executable code) olarak üretilmektedir. Ancak bir derleyicinin daha doğru tanımı bir dildeki kodu başka dile çeviren program olarak yapılabilir. Örneğin C dilinde yazılan bir programı PASCAL diline çeviren programlara derleyici adı verilebilir. Derleyicinin diğer bir tanımı ise daha üst seviye bir dilden daha alt seviyeli bir dile tercüme olarak kabul edilebilir. Buna göre örneğin C dilinden Assembly veya makine dili gibi daha alt dillere tercüme ile derleyici kavramı daha da sınırlandırılmış olarak kabul edilebilir. Derleyiciler günümüzde daha çok bir dilde yazılmış koddan, işletim sistemi ve donanım bağımlı kodların üretilmesinde kullanılmaktadırlar.

Derleyiciler bu kod üretmesi sırasında, üretilen kodun en verimli şekilde üretilmesi için kod iyileştirmesi (optimisation) da yapmaktadırlar. Yani hedef dildeki çalışma süresi ve hafıza ihtiyacı en az olan kodu üretmek bir derleyicinin daha başarılı olma kriterlerinden birisidir.

2.1 Microsoft Visual Studio

Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır.

Visual Studio IntelliSense'in yanı sıra "code refactoring" destekleyen bir kod editörü içerir. Entegre hata ayıklayıcı, hem kaynak-seviyesinde hem de makine-seviyesinde çalışır. Diğer yerleşik araçlar, GUI uygulamaları, web tasarımcısı, sınıf tasarımcısı ve veritabanı şema tasarımcısı yaratabilmek için bir form tasarımcısı içerir. Hemen hemen her işlevsellik düzeyinde dahil olmak üzere, kaynak kontrol sistemleri için destek (Subversion ve Visual SourceSafe gibi) sunan eklentileri kabul eder.

Visual Studio, değişik programlama dillerini destekler, bu da kod editörü ve hata ayıklayıcısının neredeyse tüm programlama dillerini desteklemesini sağlamaktadır. Dahili diller C/C++ (Görsel yoluyla C++), VB.NET (Visual Basic .NET üzerinden), C# (Visual C# ile), ve F# (Visual Studio 2010 itibarıyla) içermektedir.

2.2 Netbeans

Başta Java olmak üzere PHP, C++ ve daha birçok programlama dilini kullanarak yazılımlar geliştirmeye imkân sunan, tümleşik geliştirme ortamı (IDE) yazılımıdır. Açık kaynaklı bir yazılım olan ve Oracle tarafından geliştirilen NetBeans, sıklıkla Java dilinde yazılım geliştiren yazılımcılar tarafından masaüstü uygulamaları geliştirmek amacıyla kullanılır.

- Ücretsizdir ve Türkçe dilini destekler
- Pencerelerin yerleşimi ve görünümü düzenlenebilir
- NetBeans Visual Library'e sahiptir
- Depolama özelliği mevcuttur
- Kullanıcı tabanlı ayarlar yapılabilir
- Tümleşik geliştirme araçlarına sahiptir
- Framework Wizard / Kütüphane Sihirbazı bulunur

2.3 Javac

- **javac** ("java si" şeklinde okunur) Oracle Corporation tarafından sağlanan Java Geliştirme Kiti'nda (JDK) bulunan Java derleyicisidir.
- Bu derleyici Java Dil Şartnamesi'ne uygun olan kaynak kodu alır ve Java Sanal Makinesi Şartnamesi'ne (JVMS) uygun bytecode üretir.
- GNU Classpath'e ait Java Sınıf Kütüphanesi'nin (JCL) özgür bir Java gerçeklemesi olan, bu classpath çalışma zamanı sistemi (Java için GNU yorumlayıcısı) ve derleyicisi (Java için GNU derleyicisi)'ni kullanarak Javac'ı derlemeyi ve çalıştırmayı desteklemekte ve aynı zamanda GNU Classpath sınıf kütüphanesini, kiti ve örnekleri derlemeye izin vermektedir.

2.4 Komut Satırı Derleyicisi (csc.exe)

- Bağımsız değişkenler, bir boşluk veya bir sekme olduğu boşluk tarafından sınırlandırılır.
 - Şapka (^) karakteri kaçış karakteri veya sınırlayıcı tanınmıyor. İçin geçmeden önce karakter işletim sisteminde komut satırı ayrıştırıcı tarafından işlenir `argv` program dizisinde.
 - Çift tırnak işaretleri ("dize") içine bir dize kapsamında yer alan boşluk bakılmaksızın tek bir bağımsız değişken olarak yorumlanır. Tırnak içine alınan bir dizeyi bir bağımsız değişken karıştırılabilir.
 - Çift tırnak işareti eğik çizgi işaretinden (\) değişmez değer çift tırnak işareti karakteri (") olarak yorumlanır.
 - Bunlar hemen çift tırnak işareti koyun sürece ters eğik çizgi tam anlamıyla, yorumlanır.
 - Ters eğik çizgi çift sayıda çift tırnak işareti izlediyseniz, bir ters eğik çizgi içine `argv` dizi her çift ters eğik çizgi ve çift tırnak işareti dize ayırıcı olarak yorumlanır.

2.5 C++ için Derleyici Çeşitleri ve Özellikleri

2.5.1 Dev-C++,

Windows platformlar için tasarlanmış, açık kaynak yazılım felsefesi ile hazırlanmış bir C++ derleyicisidir (**compiler**).

GCC/Mingw ve Cygwin gibi pek çok gelişmiş grafik kütüphanesini de destekleyen **Bloodshed Dev-C++ Compiler**/Derleyici, Türkçe dil desteğine de sahiptir. Ufak boyutlarda, gelişmiş ve grafik kütüphanelerine de sahip bir C++ **Compiler (Derleyici)** arıyorsanız, Bloodshed Dev-C++ işinizi görebilecek güçte bir derleyicidir.

2.5.2 Intel Compiler

Intel Compiler ile Windows işletim sistemleri için maksimum performansa sahip programlar **derleyebilirsiniz**.

Masaüstü, sunucular, cep telefonları gibi mobil cihazlar için yüksek hızlarda çalışan, iyi optimize edilmiş bir C++ **derleyicisi** arıyorsanız, **Intel C++ Compiler (derleyici)** tam aradığınız geliştirme ortamıdır.

Intel Compiler, derleyici programlarının en popülerlerinden olan **Microsoft Visual C++ 6,0** ve .NET gibi geliştirme ortamları ile binary uyumludur (compatible).

Programın Özellikleri

- Uygulamalardan maksimum performans almak için, Intel Compiler ile derlenen programlar en iyi şekilde optimize edilir.
- Daha önceden kullanmış olduğunuz Windows derleyiciler ile tamamen uyumludur, beraber kullanabilirsiniz.
- Teknoloji devri Intel tarafından hazırlanıp, güncellemeleri de Intel tarafından yapılmaktadır.

2.6 Digital Mars C/C++ Compiler

Kapsamlı bir C/C++ **derleyicisi (compiler)**, optimize bir linker, standart **win32 header**(başlık) dosyaları ve çalışma zamanı kütüphane dosyaları sağlayan komple bir C/C++ geliştirme ortamı derleyicisidir.

Win32 ve DOS sürümleri mevcut olup, C/C++ dillerinde uygulama geliştirmek ve çalışma yapmak için ideal.

2.7 .Net Uyumlu Derleyiciler

.NET platformunda istediğimiz programlama dili ile program yazabileceğimizi önceki yazımızda söylemiştik. Bunun için tek şart, kullandığımız dilin. .NET için yazılmış olan bir derleyicisine ihtiyacımız olduğudur. .NET uyumlu programlama dili oluştururken belirli standartlara uyulması gerekir. Bu standartlar CLS (Common Language Specifications - Dillerin ortak özellikleri) ile belirlenmiştir. CTS(Common Type System) ise veri tipleri, nesneler, arayüzler ve programlama dillerine ait özellikleri tanımlar ve CLS'in bir parçası olarak karşımıza çıkar. CLS'de tanımlanmış kurallara uymak şartı ile istersek kendi programlama dilimizi dahi geliştirebiliriz veya herhangi bir dili. .NET platformunda uygulama geliştirmek üzere değiştirebiliriz.

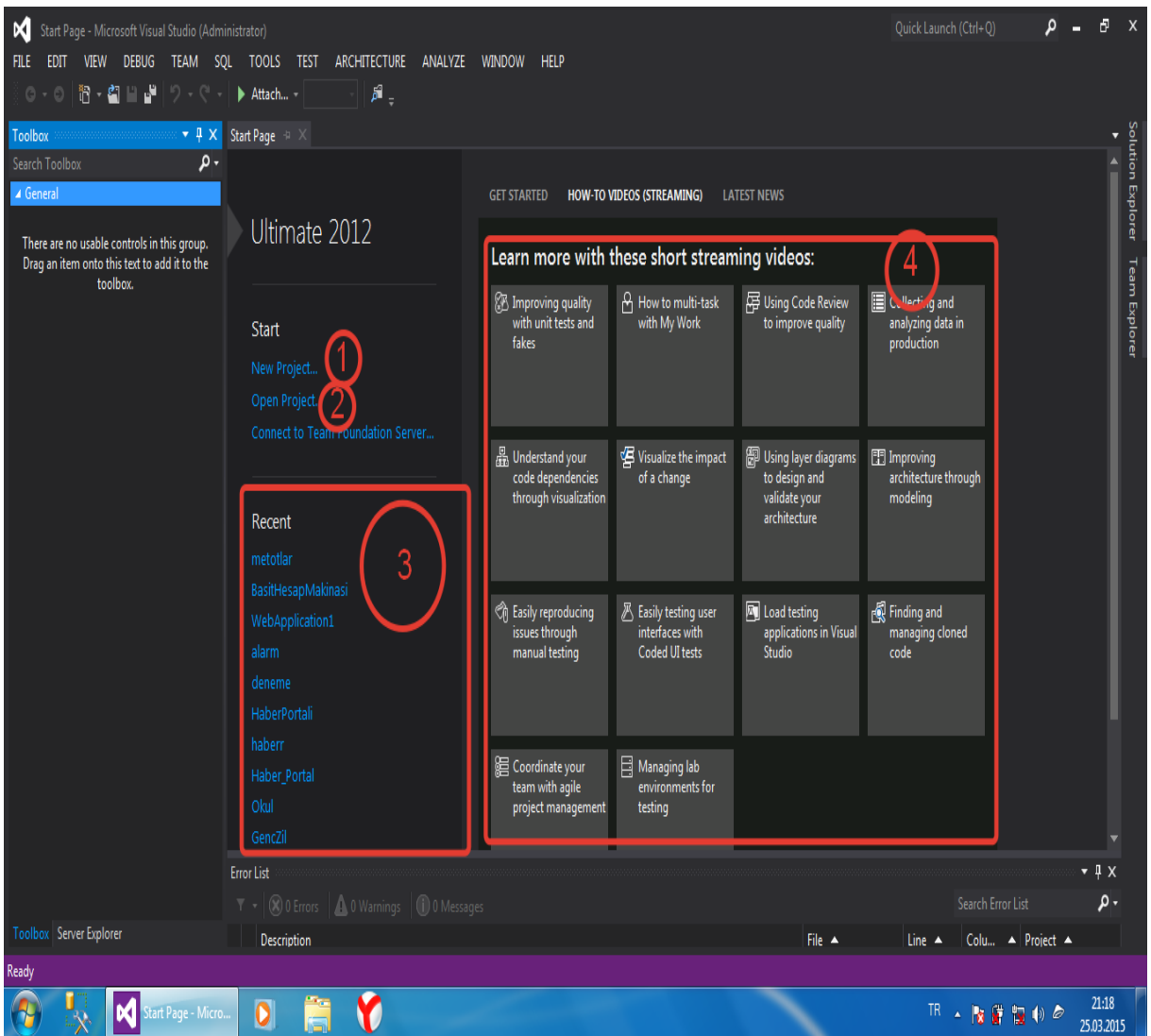
CLR, programlarımızı değişik şekilde derleyebilir. Varsayılan derleme türü JIT(Just IN TIME- çalışma anında derleme) 'dır. Program çalışırken daha önce derlenmemiş bir parçasına gelince hemen o kısmı da derler ve bunu hafızda chach'e koyar. Tekrar aynı program parçasını çalıştırmak gerekirse burayı hafızadan çalıştırır. Eğer RAM 'imizi yeteri kadar büyükse, programın tamamı derlenmiş ve hafızada depolanmış durumda olabilir. Bu durumda programımız çok hızlı çalışır.

Hafızamızın yeteri kadar büyük olmadığı durumlarda EconoJIT (Ekonomik JIT) derleyicisini kullanabiliriz. Bu derleyici ile programın derlenmiş kısımları hafızada depolanmaz ve her seferinde aynı program parçası derlenir. Tabi ki bu derleyici normal JIT'e göre programlarımızı daha yavaş çalıştırır. Ama RAM 'imizi çok daha az kullanır.

CLR ile gelen üçüncü derleyicimiz PreJIT(ön JIT derleyicisi) ise derleme işini program çalışmadan önce yapar ve tüm makine kodlarını bir yerde saklar. Çalışma anında çok hızlı olan programımız diğer JIT derleyicileriyle derlenmiş olanlara nazaran çok hızlı çalışır.

Kolayca görebileceğimiz birkaç noktaya da parmak basmak istiyorum. .NET ile yazdığımız programlar diğerlerine göre yavaş çalışır. Çünkü iki defa derleme aşamasından geçerler, program kodu MSIL'ye, MSIL ise makine koduna çevrilir. Diğer taraftan. NET ile programlarımız platform bağımsız olacak, .NET uyumlu herhangi bir dil ile program geliştirebileceğiz ve programımız CLR altında daha güvenli bir şekilde çalışacaktır.

3. Visual Studio Kullanımı ve Menü seçenekleri

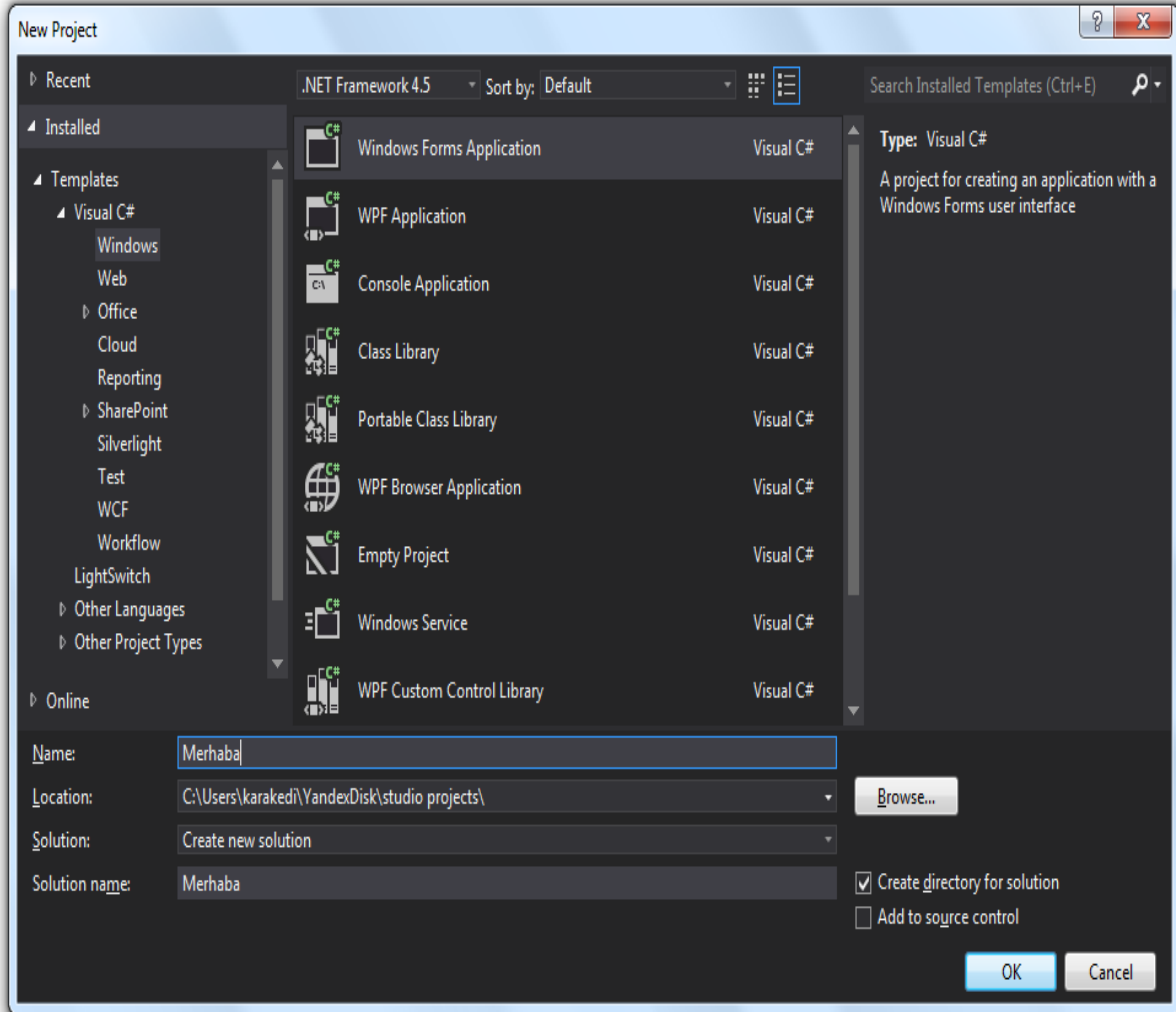


Visual Studio programını başlattığımızda karşımıza resimdeki gibi bir görüntü gelir. Benim kullandığım programda tema bulunduğundan dolayı ilk kurulumdan farklı olarak siyah görünüyor. Bu arayüz ihtiyacımız olan olabilecek işlemlere doğrudan erişmemize olanak sağlar.

Bu ekrandan yeni bir proje oluşturabilir, daha önceden oluşturulmuş bir projeyi açabilir ya da eğitim içeriklerine erişebilirsiniz.

1 numaralı bağlantı yeni proje oluşturmamızı, 2 numaralı bağlantı mevcut olan bir projeyi açmamızı sağlar. 3 numaralı bölümden en son çalışılan projelere erişebiliriz. 4 numaralı bölüm ise eğitim içeriklerinin bulunduğu bölümdür.

3.1 Proje çeşitleri



Visual Studio başlangıç sayfasında New Project (yeni proje) linkini ya da menü kısmında bulunan File -> New Project yolunu izleyerek yeni bir proje başlatabilirsiniz. Bu durumda karşınıza şöyle bir pencere gelecektir.

Burada bulunan projelerin hepsinden tek tek bahsetmek yerine en çok kullanacağımız daha doğrusu bizim işimizi şimdilik görecektir, bize temel hazırlayacak olanlardan bahsetmek istiyorum.

Windows Form Application (Windows Form Uygulaması)

Geleneksel bağımsız (Standalone) Windows uygulamaları ve Web ya da İnternet üzerinden çalışabilen dağıtık (distributed) ağ uygulamaları geliştirmek için kullanılan menüdür. Örneğin, Windows Hesap Makinesi, Wordpad, Skype, Visual Studio vb...

Bizlerde girişte konuları daha iyi anlamak için Windows Form Uygulamaları oluşturarak gideceğiz.

WPF (Windows Presentation Foundation) Application (Windows Sunum Temelli Uygulama)

WPF altyapısını kullanan bağımsız Windows uygulamaları geliştirmek için bu projeyi kullanırız. WPF, Microsoft DirectX teknolojisini kullanarak işletim sisteminin kullanıcı arayüz özelliklerine ilave olarak daha zengin bir kullanıcı arayüzü sunar. WPF projelerinde kullanıcı arayüzü elemanlarına erişim için XML tabanlı bir dil olan XAML dili kullanılmaktadır.

Console Application (Konsol Uygulaması)

Grafik bir ara yüzden son derece uzak olan uygulamalardır. Kullanıcı ile olan etkileşimi tamamen metin tabanlıdır ve bu etkileşim komut satırı üzerinden gerçekleştirilir. Windows komut satırı üzerinden çalıştırılan programlar buna örnek olarak gösterilebilir.

Class Library (Sınıf Kütüphanesi)

Birçok uygulama tarafından ortak olarak kullanılabilecek sınıf ve bileşenler oluşturmak için kullanılır. .dll dosyaları sınıf kütüphanesi tipinde olan dosyalara örnek olarak gösterilebilir.

WPF Browser Application (WPF Web Tarayıcı Uygulaması)

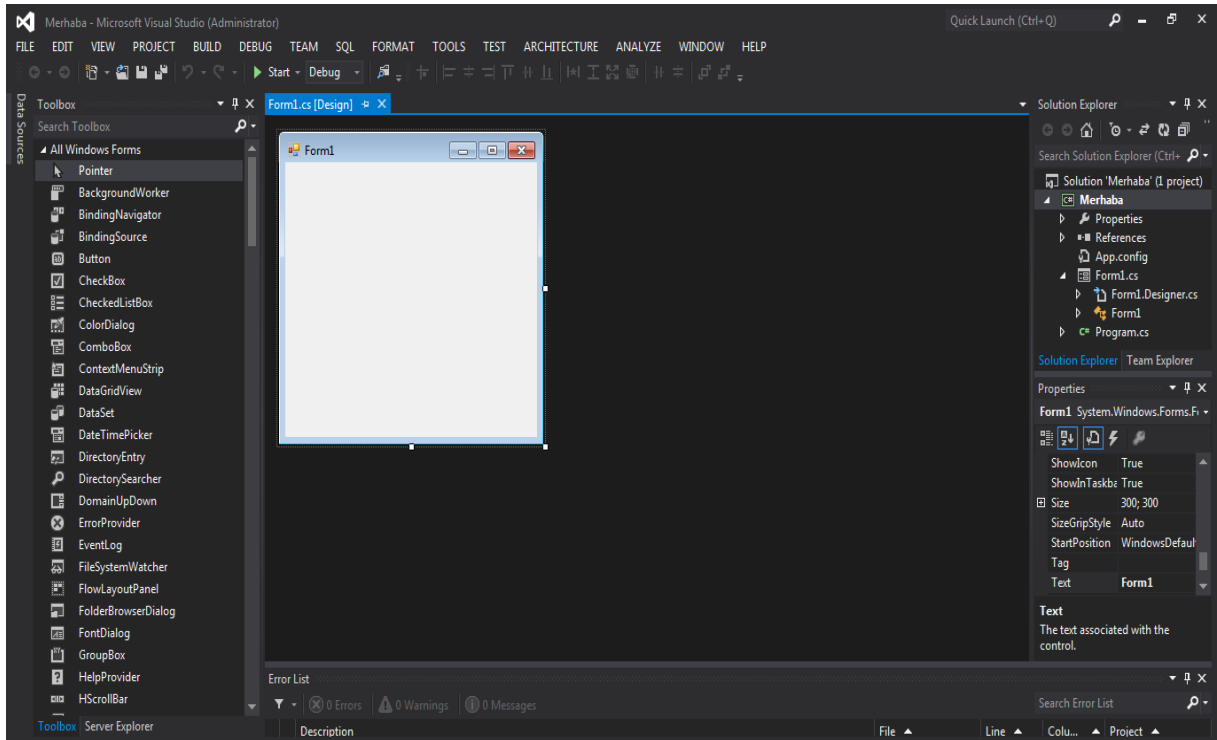
Teknoloji açısından karşılaştırıldığı zaman WPF uygulamalarıyla aynıdır. WPF uygulamaları doğrudan işletim sisteminde çalıştırılır. WPF Web uygulamaları ise Web Tarayıcısı ile çalıştırılır. Aradaki fark budur. Bu sayede Web Tarayıcısına gömülü uygulamalar geliştirebilirsiniz.

Empty Project (Boş Proje)

Kendi proje tipimizi oluşturmak istediğimiz zaman bunu kullanabiliriz. Bu proje seçildiğinde sadece uygulama bilgilerini saklamak için gereken dosya yapısı oluşturulur. Gerisi size kalmıştır. Sınıf ve referansları sizin eklemeniz gerekir.

3.2 UYGULAMA GELİŞTİRME ARAYÜZÜ

New Project penceresinden oluşturulmak istenen proje türü seçilip, OK düğmesi tıklandığında, Visual Studio seçilen proje neyse onunla ilgili olan dosya yapısının ve gerekli dosyaların bulunduğu bir arayüzü karşınıza getirir. Bu ara yüzde program geliştirmek için ihtiyacınız olan menüler, araçlar, editörler, bilgilendirme kutuları gibi birçok panel bulunmaktadır. Bu ara yüzü kullandıkça hem programı daha çok sevecek hem de bu elemanlara daha çok aşina olacaksınız.



Şimdi de ekranımızda bulunan öğeleri kısaca tanıyalım.

Menu (Tepedeki kısım) -> Bu kısım ihtiyacımız olan her türlü komuta ulaşmamızı sağlar. Menü seçenekleri seçtiğimiz projeye ve işlemimize göre değişebilir.

Toolbar (Araç Çubuğu) -> Yukarıda bulunan menüden ulaşabileceğimiz bazı elemanlara daha hızlı ve kolay ulaşmamızı sağlar. Menünün hemen altında bulunan ikonlar toolbara aittir.

Editing Area (Düzenleme Alanı) -> Bu alanda üzerinde çalıştığımız projemize istediğimiz şekli verebiliriz. Yani çalışma masamız diyebiliriz.

Form Object (Form Nesnesi) -> Projenin ana formudur. Yukarıdaki resimde bulunan beyaz alan form nesnesidir. Bu form nesnesi kapatılırsa program sonlandırılmış olur.

Toolbox (Araç Kutusu) -> Formun üzerine yerleştirmek için tasarlanmış araçların bulunduğu alandır. İlerleyen kısımlarda bunları sıkça kullanacağız. Ekranın sol tarafında bulunur. Ama bu Allah'ın emri değildir, yeri değiştirilebilir.

Solution Explorer (Çözüm Gezini) -> Projemize ait dosyalarımızın gösterildiği bölümdür. Bu bölümden projemize yeni dosyalar ekleyebilir, üzerine çift tıklayarak çalışma alanında açabilir ya da bu dosyaları silebiliriz.

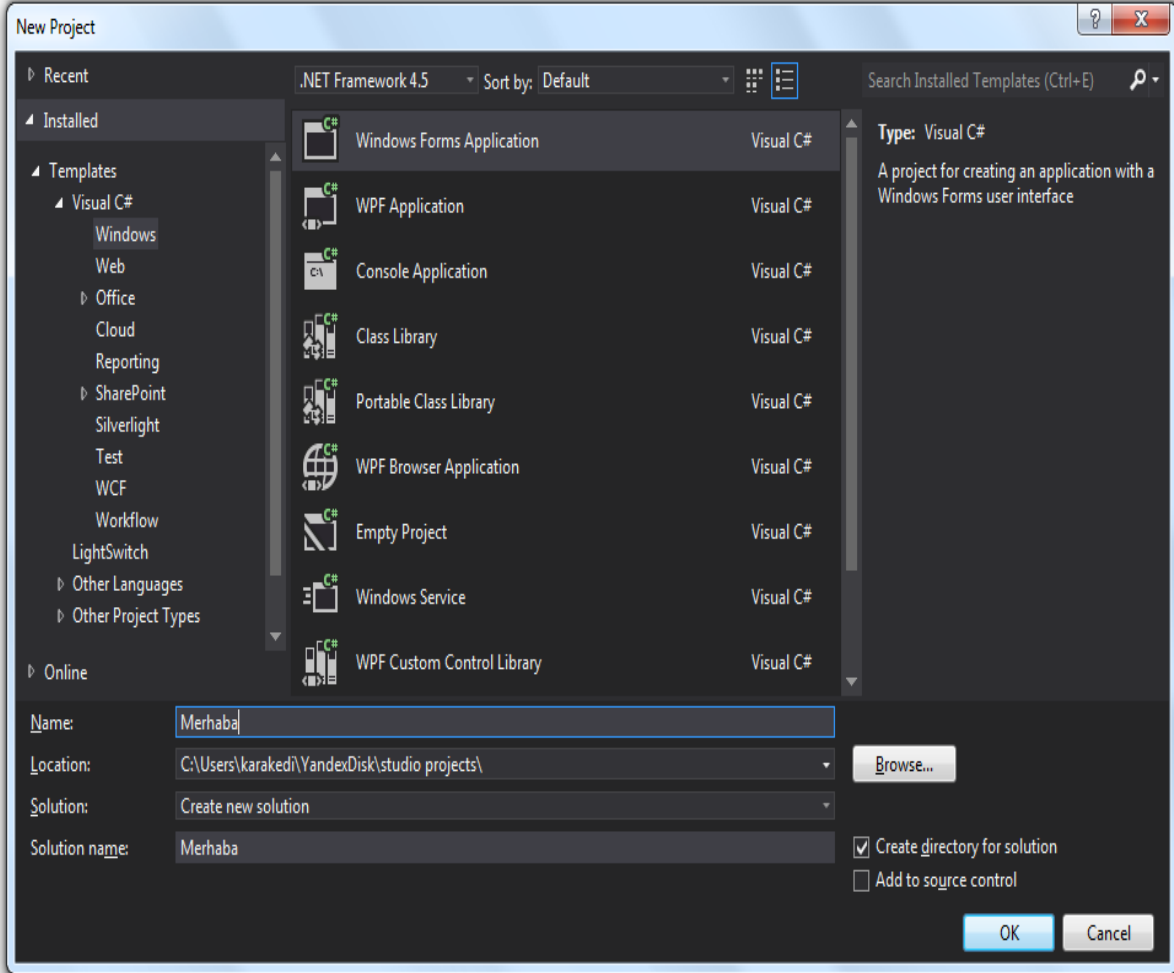
Properties Window (Özellikler Penceresi) -> Çalışma alanındaki seçili olan bir kontrolün özelliklerini görmek ve değiştirmek için kullanılır.

Property Description (Özellik Tanımlama) -> Properties penceresinde seçili olan özelliğin tanımlandığı alandır.

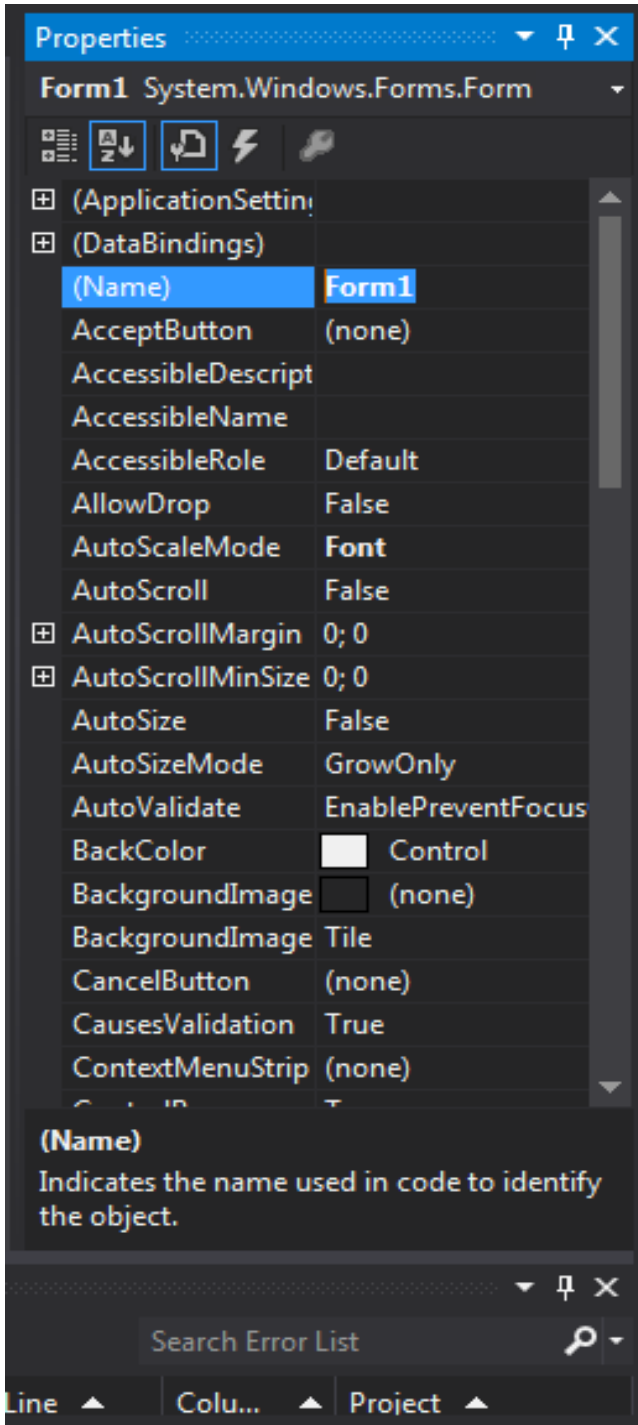
3.3 İlk Programımız

Bu tanıtımdan sonra isterseniz ilk uygulamamızı yazalım.

New Project penceresinden Windows Forms Application türünde bir proje oluşturalım. Projemize isim olarak da Merhaba ismini verelim ve OK düğmesine basalım.



OK düğmesine bastığımız zaman Visual Studio bizim için yeni bir proje oluşturur ve ana form olarak da Form1 adında bir form oluşturur. Çalışma alanındaki Formun isminin Form1 olması bana itici geldi. Eğer size de itici geldiyse değiştirelim. Bunun için sağ tarafta bulunan properties penceresini kullanacağız. Eğer sizin ekranınızda sağdaki properties penceresi görünmüyorsa Formun üzerinde sağ klik -> properties yolunu izleyerek açabilirsiniz.



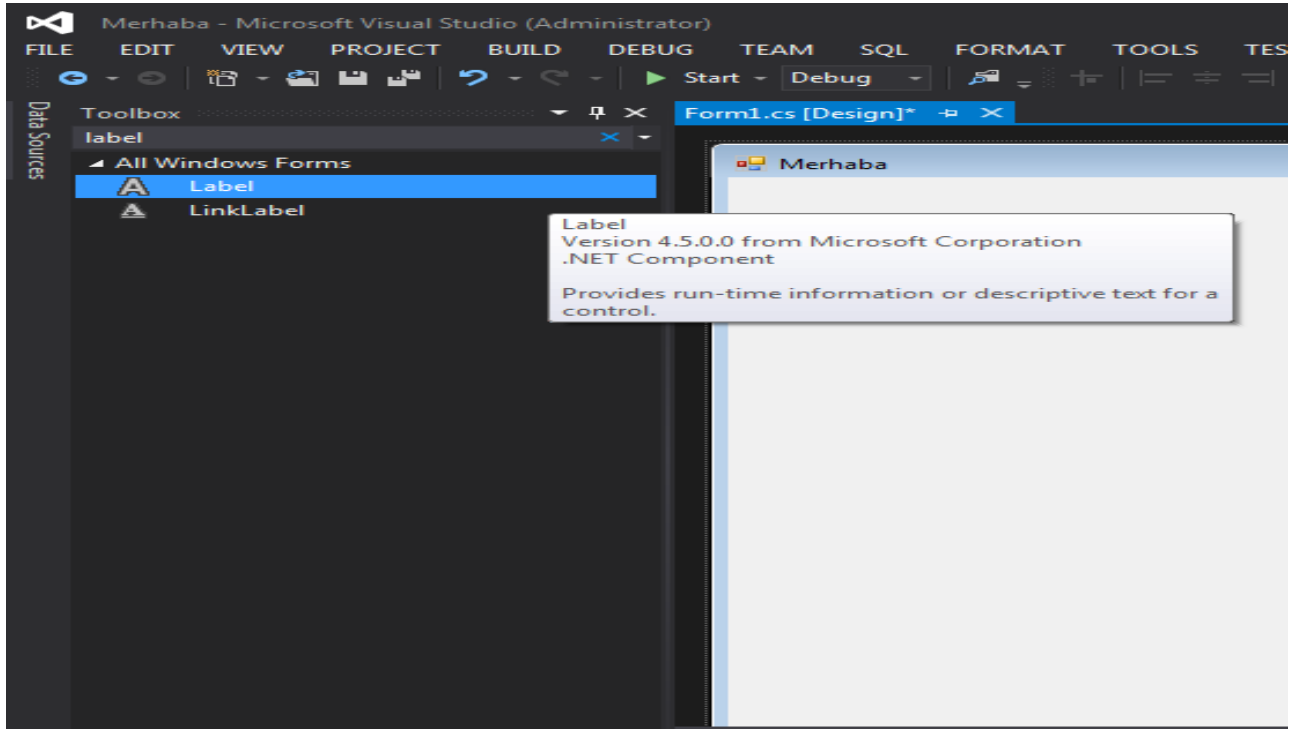
Yanda gördüğünüz name alanı formumuzun adıdır. Ancak bu ad ön kısımda yani formumuzda gördüğümüz Form1 yazısını değiştirmez. Name alanında yazılan yazı bizim kod kısmında kullanacağımız isimdir.

Türkçe karakter içermez, rakamla başlayamaz. Kod tarafında çalışırken bize bu kontrolü yani formu hatırlatacak bir isim verilebilir.

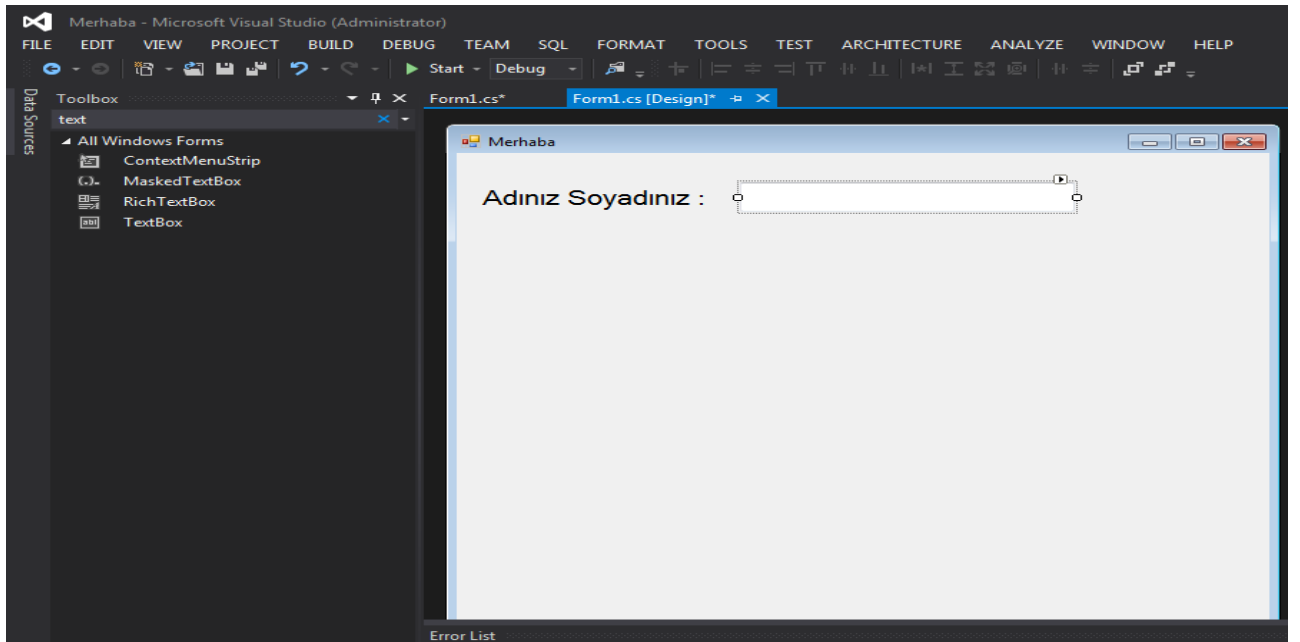
Gelelim pencerenin üzerindeki yazıyı değiştirmeye. Özellikler penceresinde aşağılara doğru indiğimiz zaman text yazısını görürüz. Buradan pencerenin üzerinde yazan yazıyı değiştirebiliriz.

Aynı zamanda bu özellikler sayfasını kullanarak sayfanın görünümünü de değiştirebilirsiniz. Biraz İngilizceniz varsa anlarsınız bunları.

Şimdi de toolbox üzerinden Label yazan aracı sürükleyerek formun üzerine bırakınız ve bunun da text özelliğini 'Adınız ve Soyadınız' olarak değiştiriniz.



Şimdi de formumuza Toolbox üzerinden sürükleyip bırak yöntemiyle bir buton atalım. Butonun text özelliğine giderek 'Tıklayınız' yazalım.



Formu sadece yerleştirdiğimiz kontrolleri alacak şekilde küçütelim. Bunun için formun boş bir alanını tıklayarak kenarlarda çıkan beyaz noktalardan çekebilirsiniz.

Eğer yanlışlıkla bir kontrole çift tıkladıysanız endişelenmeyin.

Shift+F7 tuşuna basarak ya da çalışma alanının hemen üstünde bulunan form1.cs[Design] yazan sekmeye geçerek çalışmaya devam edebilirsiniz.

Formun FormBorderStyle özelliğini Fixed3D (Sabit 3 boyutlu) olarak değiştiriniz.

Bu özellik kullanıcıların formun boyutunu değiştirmelerini engeller.

MaximizeBox (Kutuyu büyüt) özelliğini False yapınız. Bu da kullanıcıların formu büyütmesini engeller.

Start Debugging (Hata ayıklamaya başla) düğmesine tıklayarak (yukarıda bulunan yeşil start düğmesi) ya da hemen hemen her programda olduğu gibi **F5** tuşuna basarak programı çalıştırabilirsiniz.

Uygulamayı çarpı işaretinden kapattığınızda otomatik olarak Studio ekranına geri dönersiniz. Bazı durumlarda programı Studio ekranından da kapatabilirsiniz. Bunun için Start Debugging düğmesinin sağında bulunan Stop butonunu kullanabilirsiniz.

4. C,C++,C#, Java Veri Tipleri Tablosu

Veri Tipleri		C++		C	
Tür(Type)	Tanım	Depolama alanı boyutu	Değer Aralığı	Depolama alanı boyutu	Değer Aralığı
İşaretili Türler					
int	Tamsayı	4 byte	2147483648 ile 2147483647 arası	2 veya 4 byte	-2,147,483,648 ile 2,147,483,647 arası
short	Kısa Tamsayı	2byte	-32768 ile 32767	2 byte	32,768 ile 32,767
long	Uzun Tamsayı	4byte	2,147,483,648 ile 2,147,483,647	4 byte	-2.147.483.648 ile 2.147.483.647
float	Kayan Ondalık Sayı	4byte	+/- 3.4e +/- 38 (~7 haneli)	4 byte	3.4E +/-38 (7 haneli)
double	Çift Ondalık Sayı	8byte	+/- 1.7e +/- 308 (~15 haneli)	8 byte	3.4E +/- 308 (15 haneli)
long double	Uzun Ondalık Sayı	8byte	+/- 1.7e +/- 308 (~15 haneli)	10 byte	1.7E +/- 4932 (19 haneli)
char	Karakter	1byte	127 ile 127 veya 0 ile 255	1 byte	-128' den 127' ye kadar
İşaretsiz Türler					
unsigned int	İşaretsiz Tamsayı	4byte	0 (sıfır) ile +4 294 967 295	4 byte	0 ile 65,535 veya 0 ile 4,294,967,295
unsigned short	İşaretsiz Kısa Tamsayı	2byte	0 ile 65,535	2 byte	0 ile 65,535
unsigned long	İşaretsiz Uzun Tamsayı	4byte	0 ile 4,294,967,295	4 byte	0 ile 4,294,967,295
unsigned char	İşaretsiz Karakter	1byte	0 ile 255	1 byte	0 ile 255

Signed(işaretli) hem pozitif hemde negatif değerler barındıran tiptir.

Unsigned(işaretsiz) ise sadece 0 ile alabildiği maksimum pozitif değer barındıran tiptir.

Veri Tipleri			C#	Java	
Tür(Type)	Tanım	Depolama alanı boyutu	Değer Aralığı	Depolama alanı boyutu	Değer Aralığı
İşaretli Türler					
Byte	Tam sayı	1byte	0.....255	1byte	-128 ile +127
int	Tamsayı	4 byte	-2,147,483,648 ile 2,147,483,647 arası	2byte	-2,147,483,648 ile 2,147,483,647 arası
short	Kısa Tamsayı	2byte	-32768 ile 32767	4byte	32,768 ile 32,767
long	Uzun Tamsayı	8byte	9,223,372,036,854,775,808 ile 9,223,372,036,854,775,807	8byte	-2.147.483.648 ile 2.147.483.647
float	Kayan Ondalık Sayı	4byte	+/- 3.4e +/- 38	4 byte	1.4e-045 ile 3.4e+038
double	Çift Ondalık Sayı	8byte	(+/-)5.0 x 10-324 ile (+/-)1.7 x 10308	8 byte	4.9e-324 ile 1.8e+308
decimal	Uzun Ondalık Sayı	16byte	(-7.9 x 1028 ile 7.9 x 1028) / 100 ile 28		yok
char	Karakter	2byte	0.....65,535	2byte	-128' den 127' ye kadar
String(Referans tipli)	Metin	2byte	0.....2 milyar karakter	2byte	alfnumerik değişkenler tanımlamak için kullanılır
Bool	Lojik	2byte	True veya False(False anında 0 döndürür)	1 bit	128 ile 127
Object(Referans tipli)	Genel	4byte	Bütün veri tiplerinin türediği temel sınıf		
İşaretsiz Türler					
unsigned int	İşaretsiz Tamsayı	4byte	0 ile +4 294 967 295	4 byte	0 ile 65,535 veya 0 ile 4,294,967,295
unsigned short	İşaretsiz Kısa Tamsayı	2byte	0 ile 65,535	2 byte	0 ile 65,535
unsigned long	İşaretsiz Uzun Tamsayı	4byte	0 ile 18,446,744,073,709,551,615	4 byte	0 ile 4,294,967,295

4.1 TEMEL VERİ TİPLERİ

Bir verinin bellekte nasıl tutulacağını, değerinin nasıl yorumlanacağını ve veri üzerinde hangi işlemlerin yapılabileceğini belirleyen bilgiye **veri tipi** denir.

Bazı diller veri tiplerini önceden ayırır, bazıları ayırmaz.

Teknik olarak, veri tipini önceden belirleyen diller, ana belleğe yerleşecek verileri, orada kapsayacakları alanlara göre gruplara ayırır. Bu grupları pratikteki kullanımlarına göre adlandırır. Örneğin, karakterler, tamsayılar, kesirli sayılar vb gibi yalın tipler yanında dizim (array) vb gibi birleşik yapılar bu ayrımı yapan her dilde vardır.

C/C++, Java vb diller veri tiplerini önceden ayıran dillerdir.

Bu iki yöntemin iyi ve kötü sayılabilecek yanları vardır. Verileri önceden tiplerine ayırmanın avantajları şöyle sıralanabilir:

- Her veri tipine yetecek ve ancak o kadar anabellek alanı ayrılır.
- Programcı, ana bellekte ayrılacak adresin büyüklüğünü bilerek değişkenlerini tanımlar.
- Her değişkene, ana bellekte bir yer ayrılır. Değişken etkin olduğu sürece, o adrese bir başka veri yazılamaz. Programcı her istediğinde değişkenin değerini değiştirebilir, yeni değer atayabilir. Bu işlem değişkenin adresini değiştirmez.
- Her veri tipi üzerinde yapılacak işlemler farklıdır. Örneğin sayılarda yapılan işlemlerle metinlerde yapılan işlemler farklıdır. Bu fark gözetilerek veri tipleri üzerinde işlem tanımlanır. Bu olgu programcıya kolaylık sağlar.

Buna karşın, Verileri önceden tiplerine ayırmayan dillerin avantajları şöyle sıralanabilir:

- Programcı veri tipleriyle uğraşmaz; gerekli verileri girer ve kullanır.
- Aynı adlı değişkene her istediğinde aynı ya da başka tipten veri atayabilir.
- Bir veri ana belleğe bir kez konulur. O değeri kullanan değişkenler o değeri ortak kullanır. Böylece ana belleğe bir veri ancak bir kez girer. Bu olgu, anabelleğin çok ekonomik kullanılmasını sağlar.

4.1.1 C/C++

C/C++ dili verileri önceden tiplere ayıran dillerdendir.

Tamsayılar Karakterler Mantıksal Kesirli Sayılar Yapısal Tipler	İnt Char False=0,true!=0 Float, double array, string, functions, struct, files, pointers
---	--

Ama C/C++ dilinde bazı ilkel tipleri standart kütüphaneden çağırmak gerekebilir. İlkel veri tipleri için bir standart yoktur. Ama her dilde karakterler, tamsayılar (integer), kesirli sayılar (float) yalın tiplerdir. Dizim (array), string, fonksiyon, dosya, işaretçi vb birleşik (yapısal) tiplerdir. Bazı diller metin (string) tipleri de ilkel veri olarak alır. C dili string tipini karakter dizimi olarak tanımlar

4.1.2 C#

Dilinde 2 türde değişkenler vardır.

Bunlar **Değer Tipli** Değişkenler ve **Referans Tipli** Değişkenlerdir.

Değer Tipli Değişkenler: Taşıdıkları ve atandıkları değerlere göre belleğin ‘stack’ bölgesinde yer alırlar. Adında da anlaşılacağı üzere alabildiği değer aralığı sabittir. Örnek: “int”, “long”, “float”, “double”, “decimal”, “char”, “bool”, “byte”, “short”, “struct”, “enum

Referans Tipli Değişkenler: Alacağı değer aralığı belli olmayan ve belleğin ‘heap’ bölgesinde saklanan değişkenlerdir. Örnek: “string” ve “object”

Bool

Koşullu yapılarda kullanılır. Bool türünden değerlere true, false gibi ifadeler örnek verilebilir.

object

Bu değişken türüne her türden veri atanabilir.

Var

C# 3.0 da artık tür belirtmeksizin değişken tanımlamamıza olanak sağlayan yenilikler mevcut. Ancak bu değişkenlerin özelliği object değişkenler gibi referans tipli değişkenler değildir. Bunun yerine değeri atanırken tipinin belirlendiği değişkenlerdir.

Datetime

İçinde zaman barındıran değişken tipidir.

4.1.3 JAVA

Nesne yönelimli programlama dillerinde, genel olarak her sınıf bir veri tipidir. Kullanılan bu tipler verileri tutmak, saklamak için kullanılır. Kullanılan veri tipine göre bellekte yer ayrılır. Java’da veri tipleri 3 gruba ayrılır.

- İlkel (Primitive) Veri Tipleri
- Referans Tipleri
- Null Tipi

Bu tipler arasında ilk ikisi kendi aralarında alt gruplara ayrılır.

İlkel Veri Tipleri

Tam sayılar: Byte, Short, Integer, Long

Ondalık Sayılar: Float, Double

Char: Java da karakterleri temsil eder. 16 bit yer kaplar.

Boolean: Bu tip sadece True-False değerini alabilir.

Referans Tipleri

- Class
- Interface
- Array

Null Tipi

Null, herhangi bir değere sahip olmama durumuna verilen isimdir.

5. KAYNAKÇA

<http://ozenozkaya.com/blog/?p=186>

<http://www.codeargs.com/icerik/cplusplusda-degiskenler-ve-veri-tipleri-21>

<https://social.msdn.microsoft.com/Forums/tr-TR/379a88e0-ebd5-4666-9b73-e16f77698a38/c-degiskenler-ve-veri-tipleri?forum=csharptr>

<http://selimkaratas.com.tr/wp/veni-basl原因anlar-icin-java-ders-notlari-1.html>