



## Hedefler

Bu üniteyi çalıştıktan sonra;

- 👉 Visual Basic ve Visual Studio 2010 arayüzü hakkında bilgi sahibi olmak,
- 👉 Bir Visual Basic Uygulamasının yapısını öğrenmek,
- 👉 Visual Basic Program Geliştirme Arabirimi hakkında bilgi sahibi olmak,
- 👉 Visual Basic'in Özelleştirilmesi hakkında bilgi sahibi olmak,
- 👉 Visual Studio Kullanarak Visual Basic ile Program Yazmaya Giriş ve Çalıştırmayı öğrenmek.

### *İçindekiler*

- Visual Basic Nedir?
- Bir Visual Basic Uygulamasının Yapısı
- Visual Basic Geliştirme Arabirimi
- Visual Basic'in Özelleştirilmesi
- Visual Studio Kullanarak Visual Basic ile Program Yazmak ve Çalıştırmak

## VISUAL BASIC VE VISUAL STUDIO 2010 ARAYÜZÜ

### Visual Basic Nedir?

**Visual Basic**, Microsoft tarafından, **BASIC** programlama dili üzerinde geliştirilmiş, olay yönlendirmeli, üst seviye, nesne tabanlı ve görsel bir programlama dilidir. Öğrenilmesi de kullanılması da oldukça kolaydır. Çok kısa sürede değerleri 100'lerce doları aşabilen programlar oluşturabilirsiniz. En güzel özelliklerinden birisi ise **Windows** ile tam uyumlu olmasıdır. **Windows** bileşenlerine (API, COM gibi) erişmek çok kolaydır ayrıca ActiveX Component gibi bir özelliği kullanma yeteneğine sahiptir. Yani bir işi yapabilecek bir programcık için saatlerce uğraşıp yaptığınızda bunu ActiveX olarak yaparsanız daha sonraları aynı özelliği kullanmanız çok kolaylaşır.

Görsel programlama için basit bir tanım yapmak gerekirse "*göz göre göre program yapmak*" diyebiliriz. Bu programlama anlayışı; önceki yıllarda kullanılan temel programlama dillerindeki bir düzenleyici (editör) ekran üzerine satır satır yazılan programlamanın yerine, Grafiksel Kullanıcı Arabirimi (GUI)'nin ve bu arabirim içerisinde program geliştirme amacıyla bulunan Entegre Geliştirme Ortamı (IDE) adlı bir araçlar koleksiyonunun almasıyla ortaya çıkmıştır.

Visual Basic'e gelince, öncelikle **BASIC** programlama dili temel alınarak ortaya çıkarılmış olduğunu söylemek gerekir. **BASIC** (Beginners-All Purpose Symbolic Instruction Code) kelime anlamı olarak "*Yeni Başlayanlar İçin Çok Amaçlı Sembolik Talimat Kodu*" anlamına gelmektedir. **BASIC** metin tabanlı düzenleyiciler (editörler) arasında en çok rağbet göreni olmuştur. Başlangıçta çok basit programların geliştirilmesi amacıyla kullanılırken, bu ilgi dolayısıyla gitgide gelişerek son halini almıştır.

### Bir Visual BASIC Uygulamasının Yapısı

Visual BASIC'de bir Windows uygulaması Solution (Çözüm) olarak tanımlanır. Bir Solution, bir veya daha fazla Proje'den oluşabilmektedir. Projeler, Form veya Formlardan ve Modüllerden oluşabilirler.

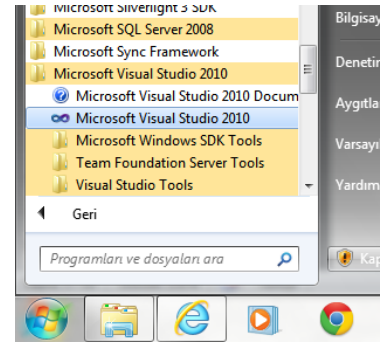
- **Formlar** : Kullanıcı ara birimi oluşturmak için kullanılan pencereler
- **Kontroller** : Formlar üzerine yerleştirilen kontrol nesneleri (text kutusu, etiketler, komut butonları vb. gibi)
- **Özellikler** : Bir form veya kontrolün her bir karakteri bir özellik ile belirlenir. Özelliklere örnek olarak; renk, pozisyon, başlık, içerik sayılabilir.
- **Metodlar** : Nesneler üzerinde işlem yapmada kullanılan hazır prosedürlerdir.

- **Olay Metotları :** Herhangi bir olay algılanır algılanmaz, o olaya uygun BASIC kodu (olay prosedürleri) işletilir.
- **Genel Metotlar :** Herhangi bir nesneye ilişkin olmayan ve uygulama tarafından çağrılan kodlardır.

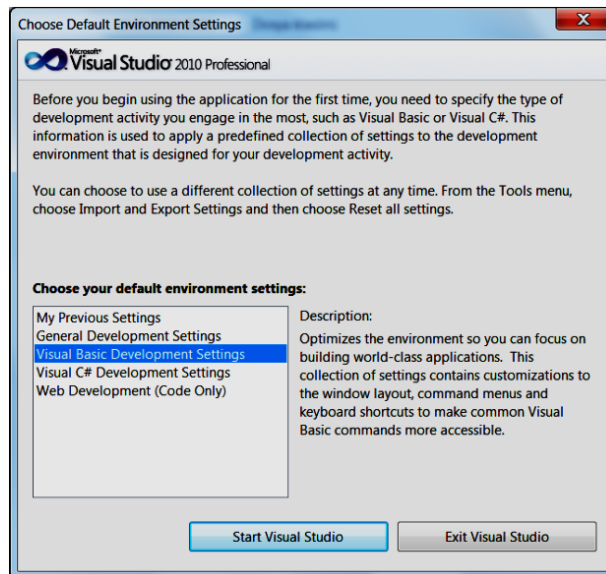
## Visual BASIC Program Geliştirme Arabirimi

Visual Studio'nun Bütünleşik Ortamı olan Program Geliştirme Arabirimi (Integrated Development Environment - IDE), çeşitli .NET programlama dilleriyle (Visual Basic, C#, C++, F#) programların oluşturulması, çalıştırılması, geliştirilmesi ve hatalarının düzeltilmesinde kullanılan oldukça güçlü bir araçtır.

Visual BASIC 2010'u çalıştırmak için, **Start (Başlat)** butonu tıklanır. **All Programs (Tüm Programlar)** seçeneğinin altındaki **Microsoft Visual Studio 2010** seçeneği bulunur ve burada açılan seçeneklerden **Microsoft Visual Studio 2010** seçeneği tıklanır.



Program ilk defa çalıştırıldığında **Choose Default Environment Settings** diyalog kutusu çıkar. Burada en çok çalışılacak dile göre ayarlamalar yapılır (Visual Basic, Visual C# gibi). Visual Studio dillerinden hangisi ile çalışılacaksa o dille ilgili seçeneği seçmek uygundur. Ama bazen bir dille bazen diğer dillerle çalışılacaksa, o zaman **General Development Settings** seçeneği seçilmelidir. Tabi ki burada yaptığınız ayarları daha sonra değiştirmeniz mümkün. Bu ayarları değiştirmek için **Tools** menüsünden **Import and Export Settings** seçildikten sonra **Reset all settings** seçilebilir.

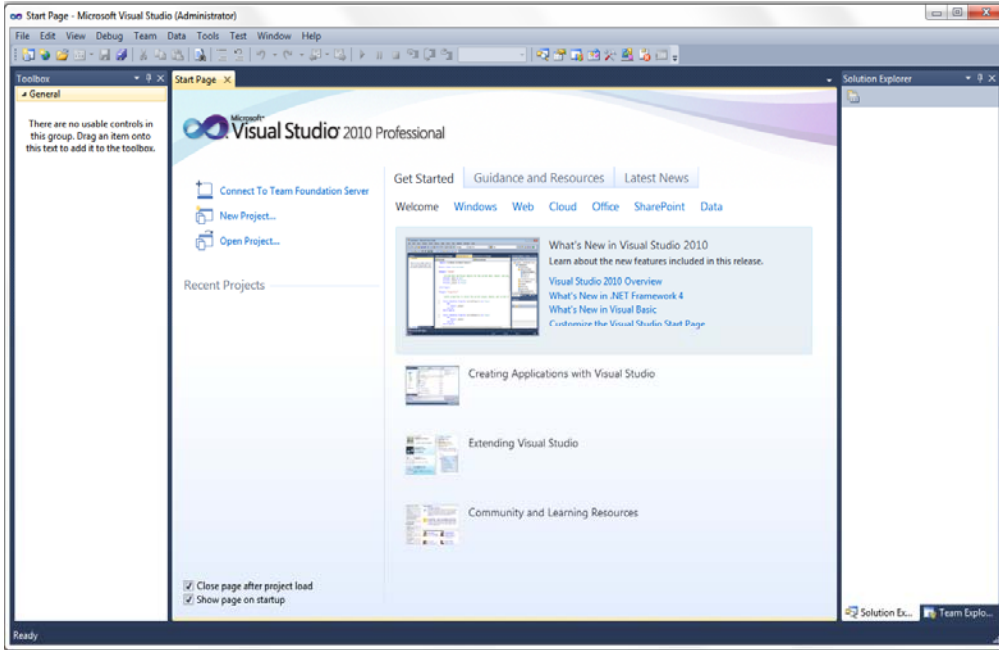


Program çalıştığında ekranda **Start Page** (Başlangıç Sayfası) sayfası görüntülenecektir. Bu sayfa, projelerle çalışmada kullanıcıya hızlı ve kullanışlı bir alternatif sunar. Ayrıca buradan bazı faydalı bağlantılara da kolayca ulaşmak mümkündür. **Start Page** sayfasının sol tarafında **Recent Projects** kısmında son çalışılan dosyalar tarih sırasına göre listelenecektir. Ayrıca **Open Project** (Önceden Kaydedilen Projeyi Açma) ve **New Project** (Yeni Proje) bağlantıları bulunmaktadır.

**Get Started** kısmında ise kullanıcı için yararlı bağlantılar sıralanmıştır. Bu linkler yardımıyla, .NET 'teki yenilikleri görmek, internetteki ilgili kaynaklara bağlanmak, örnekler indirmek mümkün olabilmektedir.

**Guidance and Resources** kısmında yardımcı kılavuz ve kaynaklar yer almaktadır. Kod geliştirme süreçleri ile ilgili Microsoft'un ilgili web adresleri açılır. Visual Studio 2010 ile gerçekleştirilebilecek unit testing (birim testi) işlemleri hakkında geniş bilgi alabileceğiniz Microsoft'un ilgili web adresi açılır.

**Latest News** kısmında ise abone olunan sitelerin RSS bilgileri görüntülenir, buradan kullanıcılar Visual Studio ile ilgili son haberlere ulaşabilir.



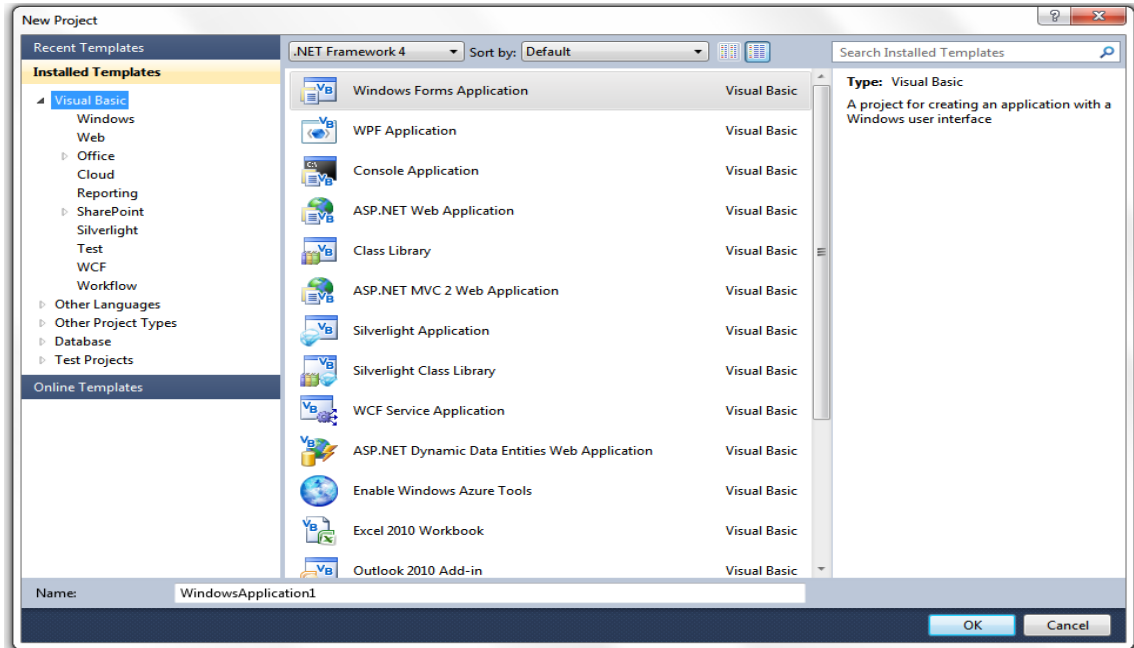
Yeni bir projeye başlamak için, **New Project** butonu veya **File** menüsünden **New → Project** seçeneği tıklanır. Yeni bir projeye başlamanın bir diğer yolu da Ctrl+N kısayol tuşlarını kullanmaktır. Ekranda **New Project** diyalog kutusu görülecektir.

Bu pencerede bulunan diller (Visual BASIC veya Other Languages seçeneğinin altındaki diğer diller) ile sağ taraftaki mevcut şablonlardan biri seçilerek proje geliştirilebilir.

Sol taraftaki *Recent Templates* seçeneği ile son kullanılan proje şablonları seçilebilir.

*Installed Templates* ise yüklü olan proje şablonları listelemektedir.

*Online Templates* ile internet ortamında bulunan proje şablonlarını listeleyebilirsiniz.

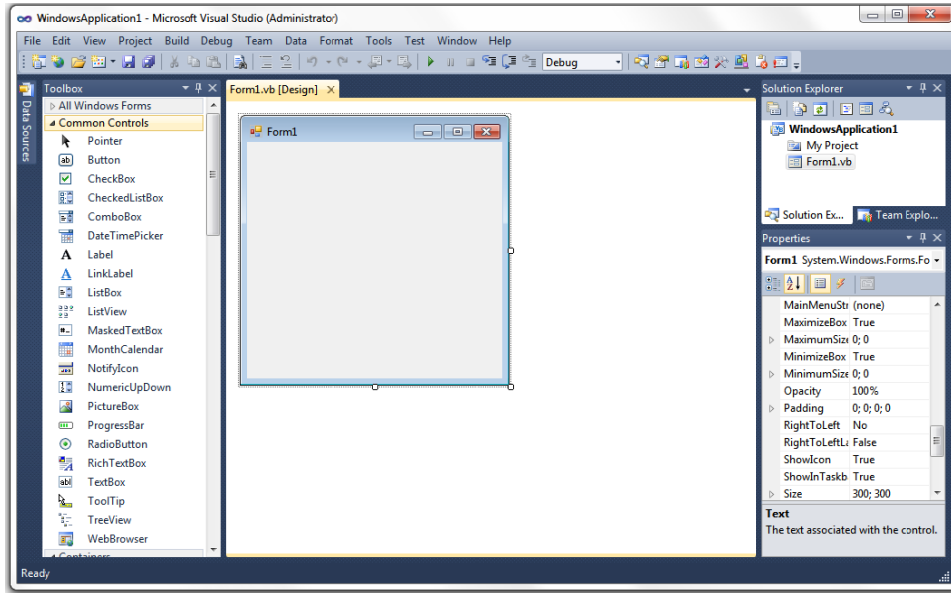


Proje geliştirirken *Templates* bölgesinde aşağıdaki şablonlardan biri seçilebilir. Şablonlar seçilen Framework versiyonuna göre değişiklik göstermektedir.

Template	Açıklama
Windows Forms Application	Windows platformunda çalışacak masaüstü uygulamalar geliştirmek için kullanılan şablondur
WPF Application	XAML kullanarak görsel açıdan zengin masaüstü uygulamalar geliştirilebilecek şablondur
Console Application	Komut İstem Satırında çalışacak uygulamalar geliştirmek için kullanılacak şablondur
ASP.NET Web Application	Web ortamında çalışacak ASP.NET uygulamaları geliştirmek için kullanılacak şablondur
Class Library	Bir başka uygulamada kullanım için class (dll) kütüphanesi oluşturmada kullanılacak şablondur
ASP.NET MVC 2 Web Application	MVC (Model-View-Control) kullanarak web uygulamalarını geliştirmek için kullanılacak şablondur
Silverlight Application	Görsel açıdan zengin özelliklere sahip Silverlight internet uygulamaları geliştirmek için kullanılacak şablondur
Windows Forms Control Library	Windows uygulamalarında kullanmak için bir kontrol oluşturma şablondur

Visual BASIC 2010, uygulamaları kaydetmek için, özel bir klasör yapısı kullanmaktadır. Yeni bir projeye (Solution) başlanırken, **Name** (İsim) istenecektir. Başlangıçta yapılan **Choose Default Environment Settings** ayarlarına bağlı olarak, **Name** (İsim), **Location** (Konum), **Solution name** (Proje ismi) (Proje için **Create directory for Solution** kutucuğu seçilmiş ise) bilgileri istenebilir. **Name** kutusuna yazılan isimle, belirtilen Konum'da (**Location**) bir klasör oluşturulacaktır. Bu klasör tüm solution (çözüm) dosyalarını, proje dosyalarını, form dosyalarını ve proje için gerekli tüm dosyaları saklamak için kullanılacaktır. Burada yeni başlayacağınız projelere otomatik isimler verilir (WindowsApplication1 gibi). İsterseniz siz farklı bir isim vererek projenize başlayabilirsiniz.

**OK** butonuna bastığımızda Uygulama Geliştirme Arabirimi ekrana gelecektir.



Ekranın üst kısmında, Ana Pencere (Main Window) bulunmaktadır. Bu pencere başlık çubuğu, menü çubuğu ve standart araç çubuğundan oluşur.

### Menü Çubuğu

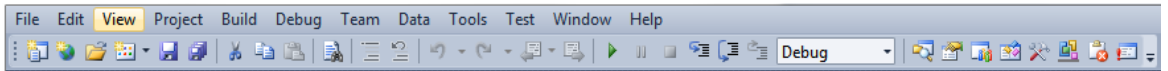


Başlık çubuğu, proje ismini, o andaki çalışma modunu ve aktif durumdaki formun ismini gösterir. Menü çubuğunda, diğer Windows uygulamalarında olduğu gibi, Visual BASIC ortamında işlemleri kontrol etmek için gerekli komutların olduğu açılan menü seçenekleri sıralanmıştır.

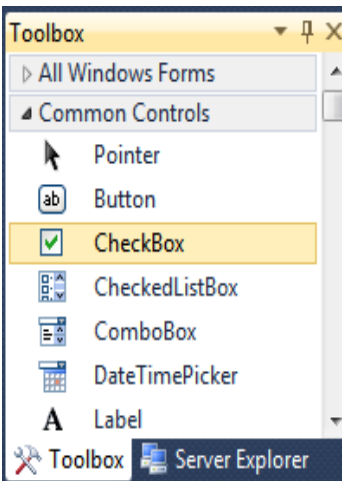
Menü	Açıklama
File	Projelerin açılması, kapatılması, yazdırılması gibi komutlar
Edit	Kopyala, yapıştır, kes, geri al gibi menü seçenekleri
View	Program Geliştirme Arabiriminin nasıl görüntüleneceği ve araç çubuklarının düzenlenmesi ile ilgili seçenekler
Project	Bir projenin ve dosyalarının düzenlenmesini sağlayan seçenekler
Build	Programın derlenmesini sağlayan seçenekler
Debug	Programdaki hataların belirlenmesi, düzeltilmesi ve programın çalıştırılması ile ilgili menü seçenekleri
Data	Veritabanları ile etkileşimi sağlayan komut seçenekleri
Tools	Program Geliştirme Arabirimine ilave araçlar ve seçenekler eklemeyi sağlayan seçenekler
Windows	Pencerelerin görüntülenmesi ve düzenlenmesi için komutlar
Help	Yardım seçeneklerine erişme komutları

### Araç Çubuğu

Araç çubuğunda, diğer Windows uygulamalarında olduğu gibi sıklıkla yapılan işlemlerin bazılarının kolayca ve hızlı bir şekilde yapılmasını sağlayan kısayol butonları vardır. Bu butonlara tıklanarak istenilen işlemler daha kolay ve daha hızlı gerçekleştirilebilir. Eğer Araç Çubuğunu göremiyorsanız, View → Toolbars → Standart komutunu verin. Mouse' unuzu araç çubuğu üzerinde gezdirdiğinizde düğmelerin kısa açıklamalarını gösteren ipuçlarını görebilirsiniz.



### Server Explorer - Toolbox



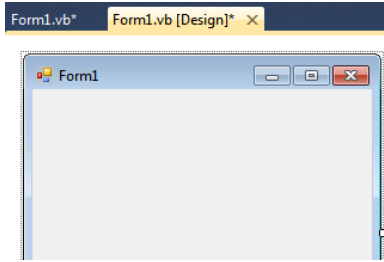
Ekranın sol tarafında **Toolbox** ile birlikte yer alan **Server Explorer** ile ağ üzerindeki serverlara erişmek mümkündür. Ayrıca buradan yeni veritabanı bağlantıları oluşturulabilir.

**Toolbox** (Araç kutusu), adından da anlaşılacağı üzere, uygulama arabirimlerinizi oluşturmak için gereksiniminiz olan parçaları içeren, uygulamada kullanılan kontrollerin oluşturulmasında yardımcı olan çok önemli bir bölümdür. Paint programında resim yapar gibi, bu araç kutuları kullanılarak Kullanıcı Arabirimi olan formlar oluşturulabilmektedir. Kontroller gruplara ayrılarak isimlendirilmiştir. **Common Controls, Containers,**



**Menus&Toolbars, Data, Dialogs** gibi. Hangi grup ismi tıklanırsa, onunla ilgili kontroller listelenecektir. **All Windows Forms** seçeneği ile tüm kontroller, bir kerede listelenmektedir. Toolbox Penceresini göremiyorsanız, View → Toolbox komutunu verin.

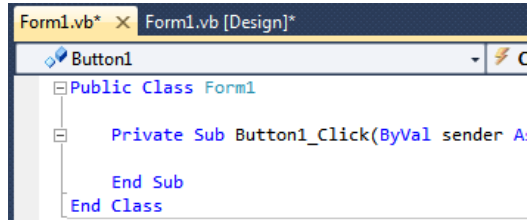
### Form Penceresi (Form)



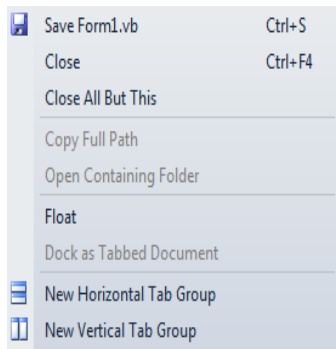
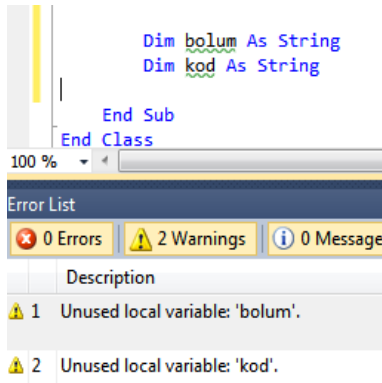
Ekranın ortasında gözüken **Form Penceresi** Visual BASIC uygulaması geliştirmenin merkezidir. Formlar kullanıcıyla etkileşimi sağlayacak olan ara yüzü geliştirmek için kullanılır. Toolbox ile birlikte, etkileşimli olarak çalışan formlarda, her kullanıcı kendi ihtiyaçlarına göre farklı tasarımlı ve farklı renk seçenekleriyle arayüzler geliştirebilir. Bu pencerenin üstünde çeşitli sekmeler yer almaktadır. Bu sekmelere (tab) tıklayarak kod sayfasına (örneğin Form1.vb\*) geçmek mümkündür. Hatta Yardım (Help) konuları seçildiğinde, bu konular burada gösterilmektedir. İlgili tab'lara basarak, form tasarımına veya kod sayfasına dönülebilir.

### Code View (Kod Editörü)

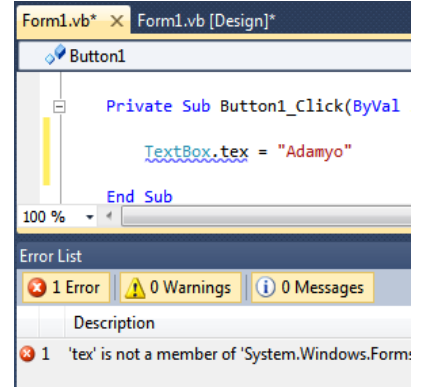
Kod Editörü Visual Studio içerisinde program bloklarının bulunduğu ve program kodumuzu yazacağımız ekrandır. Kod editörünü, Form Penceresinde bir form yada herhangi bir denetimi çift tıklayarak açabilirsiniz (View



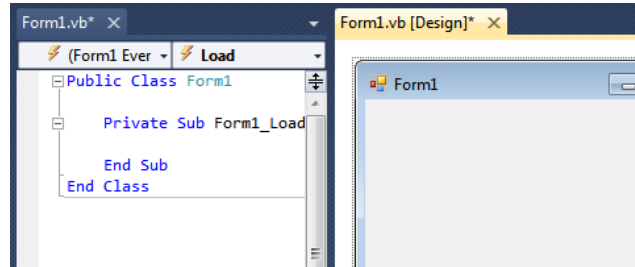
→ Code yada F7 ile de açabilirsiniz). Bir denetimi çift tıklarsanız, Kod Editörü içinde bu denetimin bir yordamına ait kod parçacığı çıkacaktır. Visual Studio Kod Editörü, basit bir metin düzenleyiciden çok daha fazla işlevlere sahiptir. Otomatik girintileme sistemi, kodların daha kolay okunabilmesini, hata aramayı kolaylaştırması açısından çok faydalıdır. Ayrıca Kod Düzenleyici, akıllı kodlar (code snippets) yardımıyla bir çok kod kalıbını otomatik olarak eklememizi sağlar. Kod editörü başlığı **Form1.vb**, Form Penceresi başlığı ise **Form1.vb [Design]** şeklinde gözükecektir.



Diğer taraftan Visual Basic.NET' in hata ve uyarı mesajları da çok işimize yarayacaktır. Metnin altına eklene mavi dalgali çizgiler, bu kelimelerde yazım hatası bulunduğu anlamına gelir. Bu durumda Visual Basic.NET kodu derlemeyecektir. Aynı zamanda hatanın ne hatası olduğu aşağıdaki

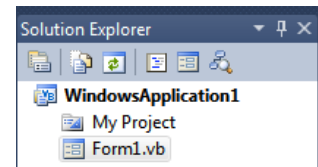


**Error List** Penceresinde gösterilecektir. Metnin altına eklenen yeşil dalgali çizgiler ise uyarıdır. Kod derlenecek ve çalıştırılacaktır, ama yine de yeşil dalgali çizgilerin üstünde yer alan kodları tekrar gözden geçirmekte fayda vardır. Aynı anda farklı ekranları yatay veya düşey olarak görüntülemekte mümkündür. Örneğin **Form1.vb [Design]** tabının üzerinde iken mouse'nun sağ tuşuna basılınca bir menü çıkacaktır. Bu menüde **New Vertical Tab Group** seçilirse, aynı anda hem kod sayfası hem tasarım sayfası düşey olarak görüntülenecektir.



## Solution Explorer



Windows Gezini' ne benzeyen ekranın sağ tarafında yer alan, Visual Studio.NET içerisindeki projeleri gruplandırarak bir çatı altında toplayan **Solution Explorer Window**, geliştirilen uygulamadaki tüm formları, modülleri ve ilgili diğer dosyaları hiyerarşik olarak gösterir.

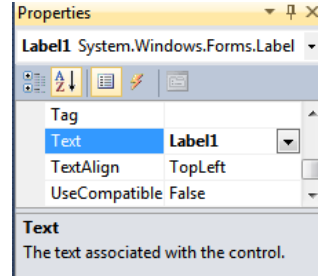


Başlangıçta yalnız gerekli dosyalar listelenir. Tüm dosyaların ve referansların listelenmesi isteniyorsa, **Solution Explorer** penceresindeki **Show All Files** butonu tıklanır. Buradaki listeden istenilen bir form veya modül, üzerine mouse ile çift tıklama yapılarak çağrılabilir.

## Properties (Özellikler) Penceresi

Zengin kullanıcı arabirimi uygulamaları geliştirirken, kullandığınız kontrollerin özelliklerini, ekranın sağ alt köşesinde yer alan **Properties** (Özellikler) Penceresi aracılığıyla görebilir ve değiştirebilirsiniz. Bu kavramı netleştirebilmek için bir uygulamadaki bütün formların ve formlar üzerindeki bütün denetimlerin birer *nesne* olduklarını düşünün. Her nesnenin renk, boy, konum, yazı tipi gibi özellikleri vardır. Bir nesnenin bütün bu karakteristiklerine *özellik* denir. Bu özelliklerin çoğu kendi kendini açıklar (Width, Height, Name gibi). Herhangi bir özelliğe tıkladığınızda, pencerenin alt tarafında onunla ilgili kısa bir açıklama çıkacaktır. Pencerenin en üstünde formdaki tüm nesnelerin listesi aşağı açılır pencerede gösterilir. Özellikler pencerede iki şekilde sıralanmıştır.

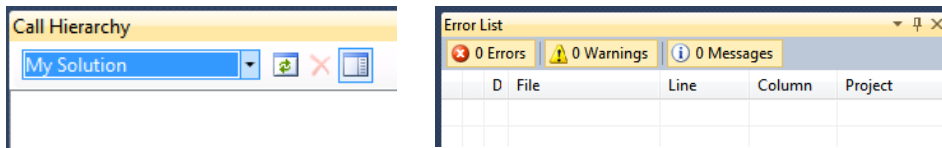
Alfabetik (Alphabetic) ve  Kategorize edilmiş (Categorized).  Herhangi bir özellik seçili iken <F1> tuşuna basıldığında, ekrana o özellik ile ilgili yardım gelecektir.



## Error List – Call Hierarchy

Programın çalıştırılması sırasında kullanılan ve ekranın alt kısmında gözükken 2 faydalı pencere daha vardır. **Error List**, proje süreci içerisinde oluşan hata ve uyarı mesajlarını görüntüler. Programın çalışmasını kesintiye uğratabacak hatalar (x) işareti ile gösterilir.

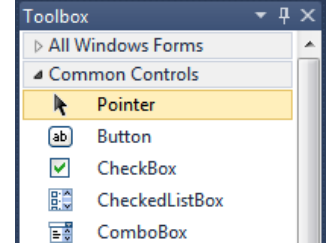
**Call Hierarchy**, bir sınıfın, bir tipin, bir alt programın metodlarını, özelliklerini, parametrelerini görüntülemeye yarar.



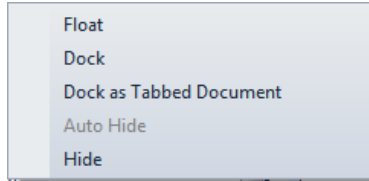
Help (yardım) özellikleri oldukça geniş ve faydalıdır. Help (yardım) menüsü 3 farklı şekilde kullanabilirsiniz.

### Visual Basic' in Özelleştirilmesi

Microsoft Visual Basic çok esnek bir özelleştirme seçeneğine sahiptir. Ekrandaki tüm pencereleri dilediğiniz gibi açabilir, kapayabilir, yerlerini ve boyutlarını değiştirebilir, sabit yada otomatik gizlenip/görünebilir hale getirebilirsiniz. Ayrıca kodlama renklerini, yazı tipini, arka planı istediğiniz gibi değiştirebilirsiniz.

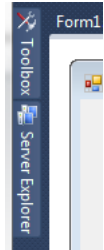


Program geliştirirken kullandığımız ortamda, çok fazla sayıda pencere ekran üzerinde bulunabilmektedir. Bazen bu ekranların bir kısmının gizlenmesi, gerektiğinde ekrana çağırılması, program yazarken karışıklığı engellemesi açısından faydalı olacaktır. Program geliştirirken en çok ihtiyaç duyabileceğiniz pencereler olan Toolbox, Solution Explorer ve Properties Pencereleri herhangi bir sebeple kapatıldıktan sonra, bu pencereleri tekrar geri açmak isterseniz, Menü çubuğundaki View menüsünden açabilirsiniz.

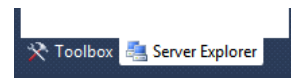


Bir panelin/pencerenin Visual Basic ilk açıldığında gibi kenarlara yapışabilir olması için “dockable” olması gerekir. Floating özelliği alan bir panel/pencere, kenarlara yapışma özelliğini kaybeder ve ekranın üzerinde bağımsız olarak konumlanır. Bu Pencereleri çalışma sayfanıza sabitlemek için Pencerenin başlık çubuğuna mouse’ un sağ tuşu ile tıkladıktan sonra “**Dock**” seçeneğini seçerek sabitleyebilirsiniz. Eğer bir panele/pencereye **Dock as Tabbed Document** özelliği vererseniz, kod ve tasarım yöneticisi ekranlarının olduğu bölümde (Form’un yanında) yer alacaktır. Toolbox, Solution Explorer ve Properties pencerelerinin başlık kısmında Auto Hide simgesi bulunmaktadır. Bu simge yardımıyla, ekranın solundaki ve sağındaki bu yardımcı pencereler gizlenebilir.

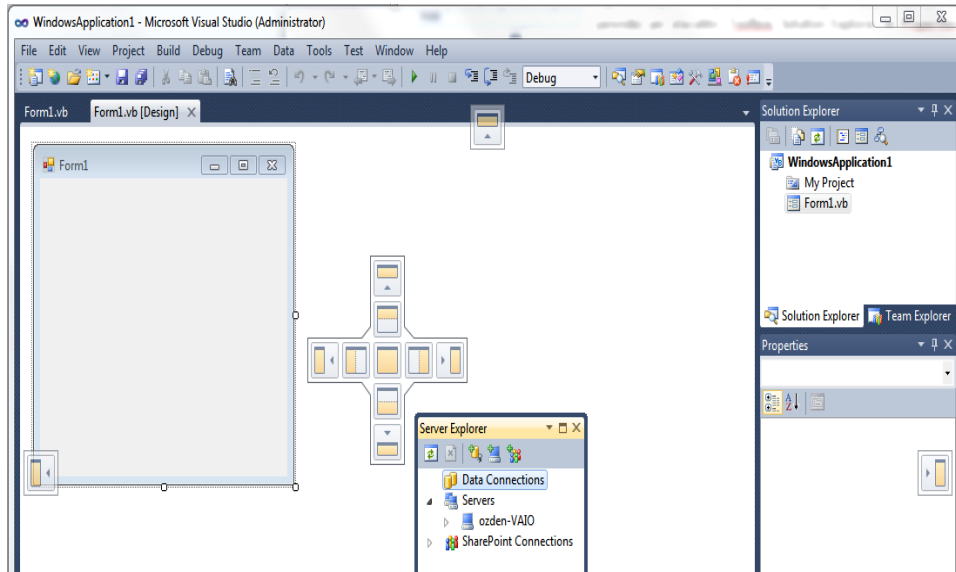
Örneğin Toolbox penceresinin başlık kısmındaki Auto Hide simgesi tıkladığında, Toolbox penceresi gizlenecek ve ekranın solunda düşey olarak Toolbox yazısı gözükecektir, yani simge durumuna düşecektir. Toolbox penceresi gerektiğinde, mouse ile Toolbox simgesinin üzerine gelindiğinde, Toolbox penceresi çıkacaktır. Diğer pencereler içinde aynı şeyler yapılabilir.



Visual Studio 2010 program geliştirme ortamında, aynı bölgede birden fazla pencere bulunabilmektedir. Mesela Toolbox bölgesinde, ayrıca Server Explorer penceresi de var. Solution Explorer Bölgesinde, Class View ve Yardım pencereleri bulunabilmektedir. Properties bölgesinde, Dynamic Help’de olabilir. Bu bölgelerin en altında yer alan tab’lar yardımıyla istenilen pencere aktif hale getirilebilir. Örneğin yandaki şekilde Toolbox bölgesinde Server Explorer aktif durumdadır. Toolbox penceresini aktif hale getirmek için, pencerenin alt tarafındaki Toolbox tab’ı tıklanmalıdır.



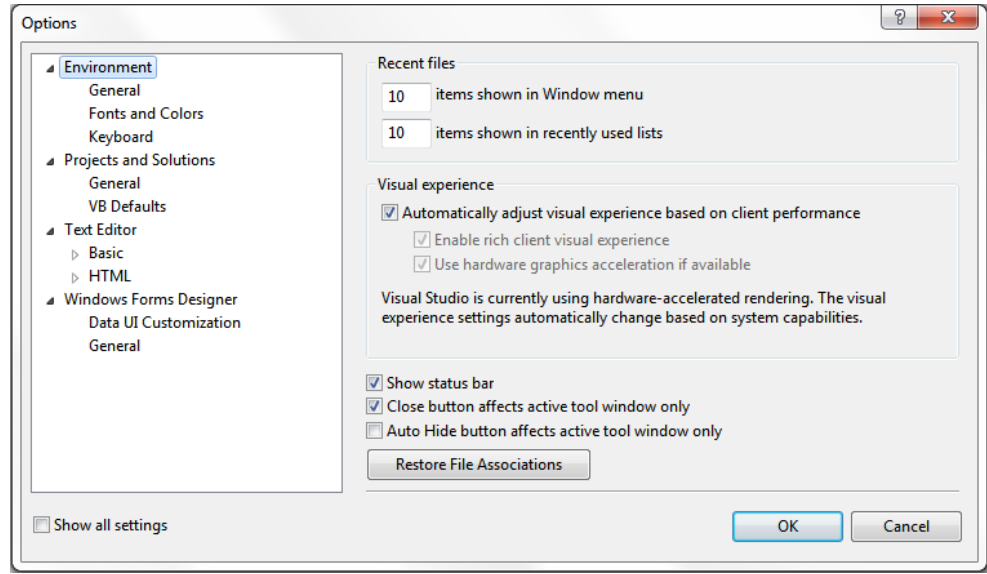
Pencerelerin yerlerini değiştirmek için başlık çubuğu kullanılır. Bir panelin/pencerenin başlık çubuğundan tutup sürüklediğinizde, panel/pencere hareket eder ve ekranda hareket ettirilen bu pencerenin konumlandırılacağı yerler gösterilir.



Program Geliştirme Arabirimindeki pencereleri başlık kısımlarından tutup istediğimiz yere taşıyabiliriz. Fakat bazen çok fazla taşımadan sonra ekrandaki düzenimiz bozulabilir ve bunları taşıma ile tekrar eski yerlerine getirmede zorlanabiliriz, hatta getiremeyebiliriz de. Tekrar default (varsayılan) ekran düzenini getirmek için, **Windows** menüsünden **Reset Window Layout** seçeneği tıklanır. Çıkan diyalog **kutusunda OK** butonuna basılırsa, varsayılan ekran düzeni tekrar sağlanacaktır.

Visual Basic' in program geliştirme ortamı ile ilgili özelleştirmeler **Tools** menüsündeki **Options** komutu ile ekrana getirilen Options diyalog kutusundan yapılabilir. Bu pencereden, menülerdeki son dosya sayılarının belirlenmesinden, kod yazı biçimlerine, derleme seçeneklerinden, açılış penceresinin hangi rss feed'i kullanacağına kadar birçok seçeneği özelleştirebilirsiniz.

**Options** diyalog kutusunda yapılabilen seçim ve ayarlamalar **Environment, Projects and Solutions, Text Editor ve Windows Forms Designer** adında 4 gruba ayrılmıştır. Diyalog kutusunun sol tarafında listelenen bu seçenekler ayrıca alt seçeneklere sahiptir. Ana seçenek grubunun üzerine tıkladığınızda, alt seçeneklerini görebilirsiniz.



## Visual Studio Kullanarak Visual BASIC ile Program Yazmak ve Çalıştırmak

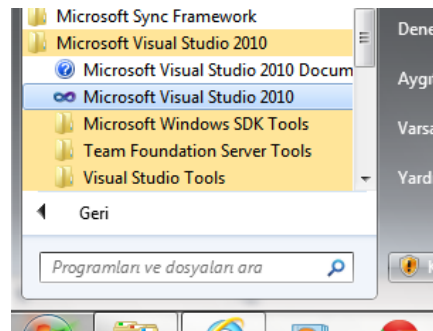
Visual Basic'te program yazmak ve çalıştırmak için 5 temel adım vardır :

- Adım 1 :** Visual Studio 2010'u Çalıştırmak
- Adım 2 :** Kullanıcı Arabirimini Tasarlamak
- Adım 3 :** Kontrollere Özellikleri Atamak
- Adım 4 :** Kontrollere Kodu Ekleme
- Adım 5 :** Programın Çalıştırılması

### İlk Uygulamamızı Oluşturalım

Yukarıdaki adımları uygulayarak, şu ana kadar öğrendiğimiz bilgilerle basit bir uygulama geliştirelim.

**Adım 1 :** Microsoft Visual Studio 2010 çalıştırılır.



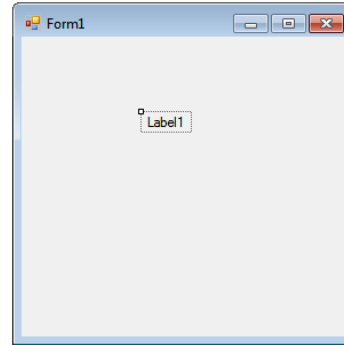


Açılan Visual Studio.NET programının ortadaki penceresinden **New Project** veya **File** menüsünden **New Projects** tıklanır. **New Projects** Penceresinde, **Recent Template** kısmından **Visual Basic** ortadaki pencereden de **Windows Application Forms** şablonu seçilir. **Name** kutusuna **Uygulama1** yazıyoruz (veya seçiyoruz). Ve **OK** butonuna basarak yeni projemize başlıyoruz.

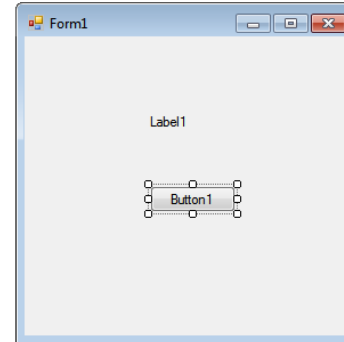
**Adım 2 : Kullanıcı arabirimini (Form) tasarlamak ;** Visual Basic'te bir form, programın kullanıcı arabirimini yaratmak için özelleştirilen bir penceredir. Bir form text kutuları, etiketler, butonlar, düğmeler, liste kutuları, kaydırma çubukları gibi öğeleri içerir. Bu öğeleri kullanıcı arabirimine eklemek için, araç kutusundaki araçlar ve kontroller kullanılır.

a) Ekrandaki Form1 'i kullanarak arabirimi geliştirmeye başlayalım. Bu Form'un varsayılan genişlik (Width) ve yüksekliği (Height) 300 x 300'dur.

b) İlk önce **A Label** (**Toolbox Penceresinden**) kontrolü çift tıklanır yada üzerine tıklanarak, **Form** üzerine sürüklenir. **Label1** etiketinin ortasına mouse ile basılı tutarak etiketi, form üzerinde istediğimiz yere taşıyabiliriz.

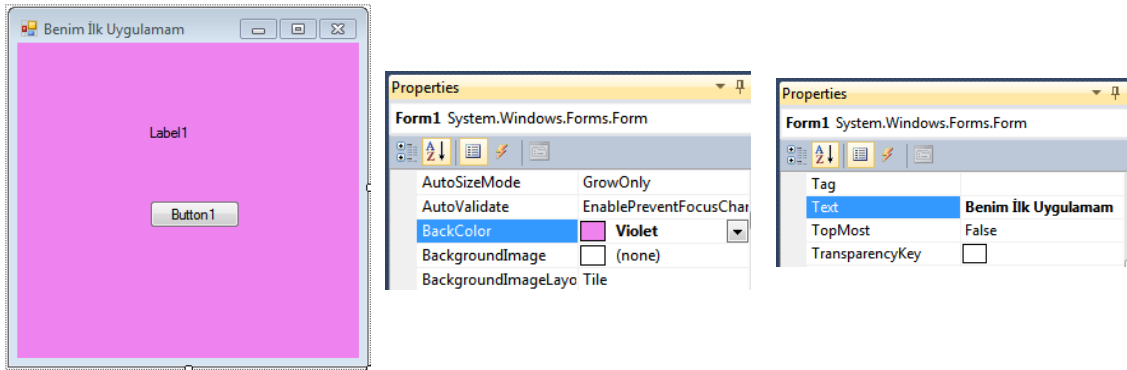


c) Şimdi Button (buton) kontrolünü çift tıklayarak yada sürükleyerek **Label1** etiketinin altına yerleştirelim. Bunun için **Button1** butonunu fare ile sol tuşuyla basılı tutarak taşıyabilirsiniz.



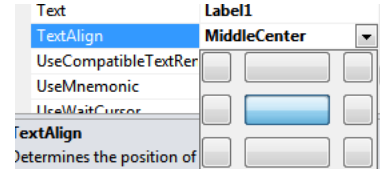
**Adım 3 : Kontrollere Özellikleri Atamak ; Properties** (Özellikler) penceresi bir form üzerindeki kullanıcı arabirimi öğelerinin özellik ayarlarını değiştirmeyi sağlar. Form üzerindeki öğeler seçilerek, nesne özellikleri **Properties** penceresinden değiştirilir.

**a) Form 'un özelliklerinin değiştirilmesi** : Form üzerine bir kez tıklanır. Sağ taraftaki **Properties** kutusunda **BackColor** (Zemin Rengi) seçeneği açılan bir listedir. Bu listeden **Custom** sekmesi tıklanarak, **Violet (menekşe)** renk seçilir. Form1 etiketinin yerine “Benim İlk Uygulamam” yazdırmak için, **Text** kutusuna **Benim İlk Uygulamam** yazılır.

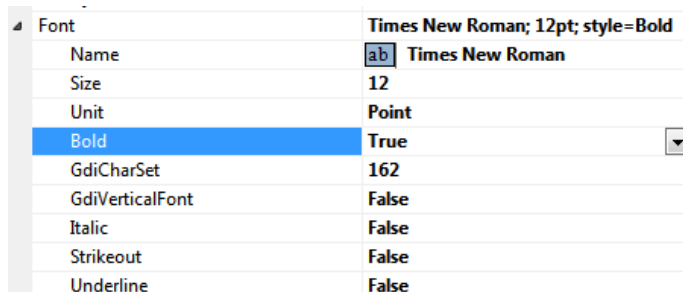


**b) Label etiketlerinin ortak özelliklerinin değiştirilmesi** : **Label1** etiketi tıklanır.

- **Properties** penceresinde **TextAlign** (Hizalamak) açılan listesinden **MiddleCenter** seçeneği tıklanır. Etiketler, kutuların yatay ve dikeyde ortasında olacaktır.



- **BackColor** (Zemin Rengi) **Custom** sekmesindeki açılan listeden **White (Beyaz)** seçilir.
- **BorderStyle** (Çerçeve stili) açılan listeden **FixedSingle** seçilir. Etiketlerin etrafına ince bir kenarlık eklenir.
- **Font** kutusuna tıklanınca sağ  tarafta bir buton çıkacaktır. Buna tıklanınca, Font (yazı Tipi), Font style (yazı tipi biçemi) ve Size (boyutunu) ayarlamak için diyalog kutusu çıkar. Yada Font seçeneğinin sol tarafındaki ok tıklandığında, aşağıya doğru yazı tipi ile ilgili seçenekler çıkacaktır. Burada **Times New Roman, Bold (Kalın)** ve **12 punto** değerleri seçilir.

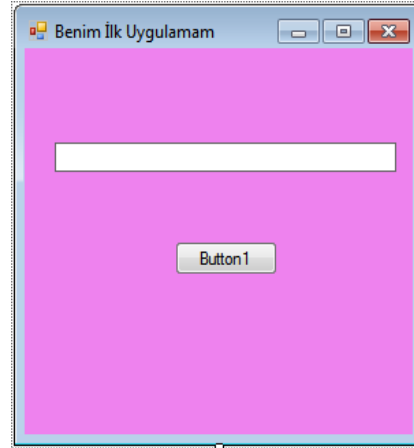




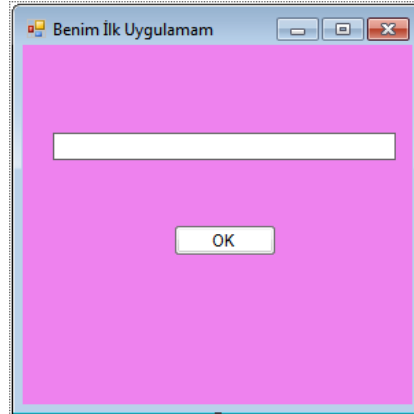
- **ForeColor (Yazı rengi)** : Etiketlerin yazı rengini ayarlamak için kullanılan bu açılan listede **Red (Kırmızı)** renk seçilir.

- **Size (Etiket Kutusunun Genişliği;Yüksekliği)** : Bu örnek için biz **250;20** değerini alıyoruz (Not : bu değer piksel denilen ölçü birimindedir Ve **AutoSize** özelliği **False** olmalıdır.)

- **Label1 etiketini silelim.** Label1 etiketi tıklanır. **Properties** penceresinde **Text** kutusundaki yazı silinir.



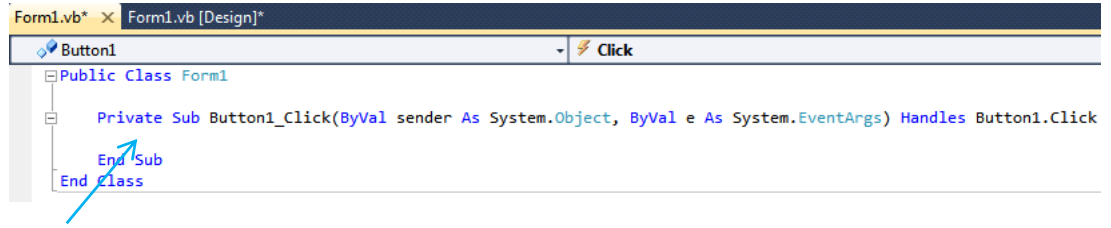
c) Şimdi de **Button1** butonunun etiketini değiştirelim. **Button1** tıklanır. **Properties** penceresinde **Text** kutusuna **OK** yazılır. **BackColor (Zemin Rengi)** de **White (Beyaz)** yapılır.



**Adım 4 : Kontrollere Kod Ekleme** : **OK** butonuna tıklandığında ne yapılması gerektiğini kod olarak yazmalıyız. Henüz programcılıkla ilgili bilgiler bilinmediği için, buraya yazılanların anlaşılması şu an önemli değil.

**OK** butonuna basıldığında ne yapılması gerektiğini kod olarak yazalım : **OK** butonuna basıldığında, Label1 kutucuğuna "Hoş Geldin Visual Basic" yazılmasını istiyoruz. Yazacağımız Kod' da bunu ifade edeceğiz.

**OK** butonuna çift tıklanır. Aşağıdaki pencere açılacaktır :



Bu satırlar arasına;

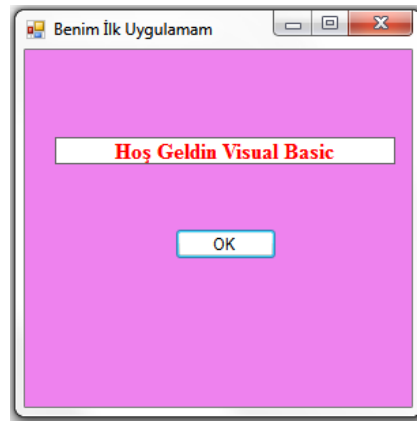
```
Label1.Text = "Hoş Geldin Visual
```


Satırını yazarız.

Bu satırın anlamı şudur : **OK** butonuna tıklandığında **Label1** kutusundaki **Text** kısmına “Hoş Geldin Visual Basic” yazısını yaz.

**Adım 5 : Programı çalıştırmak :** Hazırlanan programı çalıştırmak için aşağıdaki yöntemlerden birisini kullanabilirsiniz.

- **Debug** menüsünden **Start Debugging** seçeneği,
- **F5** tuşu,
- Standard araç çubuğundaki **Start Debugging**  tıklanarak



**Adım 6 : Son olarak program kaydedilir :** Standart araç  çubuğunda **Save All** butonuna tıklanır. Uygulama projeye başlangıçta verdiğimiz isimle klasöre kaydedilir.