

1. GENEL BİLGİLER VE GİRİŞ

1.1. .NET Framework Nedir?

.NET Framework, Microsoft firması tarafından tasarlanmış, açık internet protokolleri ve standartları üzerine kurulmuş bir uygulama geliştirme platformudur. Bu geliştirme platformu kullanılarak çok çeşitli uygulamalar geliştirilebilir. Buradaki platform işletim sistemi üzerinden ve donanımdan daha üst seviyede geliştirilmiştir. Bu platformun mimarisi ortak bir yürütme ortamı, ortak bir parametre tür sistemi ve birbiriyle bağlantılı kütüphanelerden oluşur. .Net platformu birçok farklı uygulama geliştirme ortamı ile masaüstü, mobil, internet, uzak kontrol, Windows servisi gibi çok çeşitli programlar oluşturulmasına imkan tanır. Ayrıca bu platformu kullanan yazılımların bellek yönetimini otomatik yapar ve bellekte kullanılmayan değişken varsa onları temizler. Kapsamlı *sınıf/class* kütüphanelerine sahiptir ve alt sürüme ait programlar üst sürümlerde rahatlıkla çalışır.

1.2. C# Programlama Dili

C# programlama dili, .Net platformu üzerinde çalışan bir dildir. Yani Windows işletim sisteminin kurulumu sonrası .Net platformunun ilgili versiyonu kurulmaz ise C# ile yazılan programlar çalıştırılmayacaktır. Bununla birlikte, C# ile yazılan programın .Net Framework versiyonu kaç ise işletim sisteminde kurulu olan versiyon; ya aynı olmalıdır ya da daha üst versiyon olmalıdır. Delphi veya C++ gibi bazı programlar direkt makine dili üzerinde çalışabilme yeteneğine sahip olduğundan ekstra herhangi bir platforma ihtiyaç duymazlar. Fakat C# ve Java gibi programlar .Net Framework platformu veya Java platformunun kurulumuna ihtiyaç duyarlar.

C++ dili üzerine geliştirilmiş olan C# programlama dili çok uzun uğraşlar ile geliştirilebilecek programları daha basit mantıklar ile geliştirmeye imkân tanır. Nesnel bir dil olması nedeniyle sınıf mantıkları kullanılır ve gereksiz kod tekrarları oluşmaz. Tüm bu sebepler sonucu günümüzde en çok kullanılan ve tercih edilen programlama dili C# olmuştur.

Bu programlama dili kullanılarak, masaüstü yazılımların yanı sıra, web siteleri, web servisleri, dll benzeri birçok farklı türde proje geliştirilebilir.

Microsoft firması tarafından 1999 yılı sonu itibari ile geliştirilmeye başlanan Visual C# programlama dili 2002 yılında .Net dili olarak ortaya çıkmıştır. İlk sürümü 1.0 olarak çıktıktan sonra her sürümde yeni eklemeler sonucu birçok üst düzey özelliği kendisinde barındırır olmuştur. Anders Hejlsberg'in başında bulunduğu bir ekip tarafından geliştirilen C# programlama dili .NET dili ailesi içerisindeki en güçlü dillerden birisidir.

1.3. Visual Studio Kod Geliştirme Ortamı

C# programlama dili istenirse hiçbir kod geliştirme ortamı kullanılmadan da yazılabilir ve .Net framework ile derlenebilir. Hatta bir notepad içerisinde dahi kod yazıp bunları `csc.exe` ile derlemek mümkündür. Fakat programcılar farklı dosya ve klasör yapıları ile çalıştıkları için daha kolay kod yazabilecekleri ortamlara ihtiyaç duyarlar. Microsoft tarafından geliştirilen Visual Studio kod geliştirme ortamı daha kolay, etkin ve verimli bir şekilde kod geliştirme imkânı sunmaktadır. Fakat yalnızca Visual Studio'da .NET uygulamaları

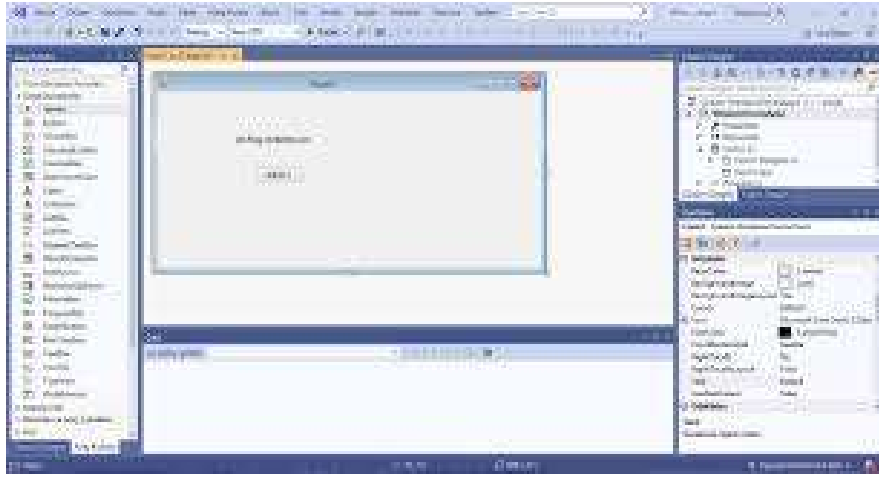
geliştirilebilir demek pek de doğru olmayacaktır. MonoDevelop ve SharpDevelop gibi farklı kod geliştirme ortamları da vardır.

2. VISUAL STUDIO VE C#

2.1. Visual Studio Pencereleri

2.1.1. Pencere Düzenlerini Özelleştirme

Visual Studio kod geliştirme ortamı, aşağıdaki şekilde gösterildiği gibi dört farklı noktada pencerelerin konumlanmasına izin verir. 1 numara ile gösterilen kısım en geniş yer olması nedeniyle çoğunlukla kod yazma ve tasarım yapma amacıyla kullanılır. Diğer pencereler ise daha az sıklıkla kullanılmasından dolayı 2, 3 ve 4 numara ile gösterilen yerlere konumlandırılır. Pencerelerin yükseklik ve genişlikleri istenildiği gibi ayarlanabilir.

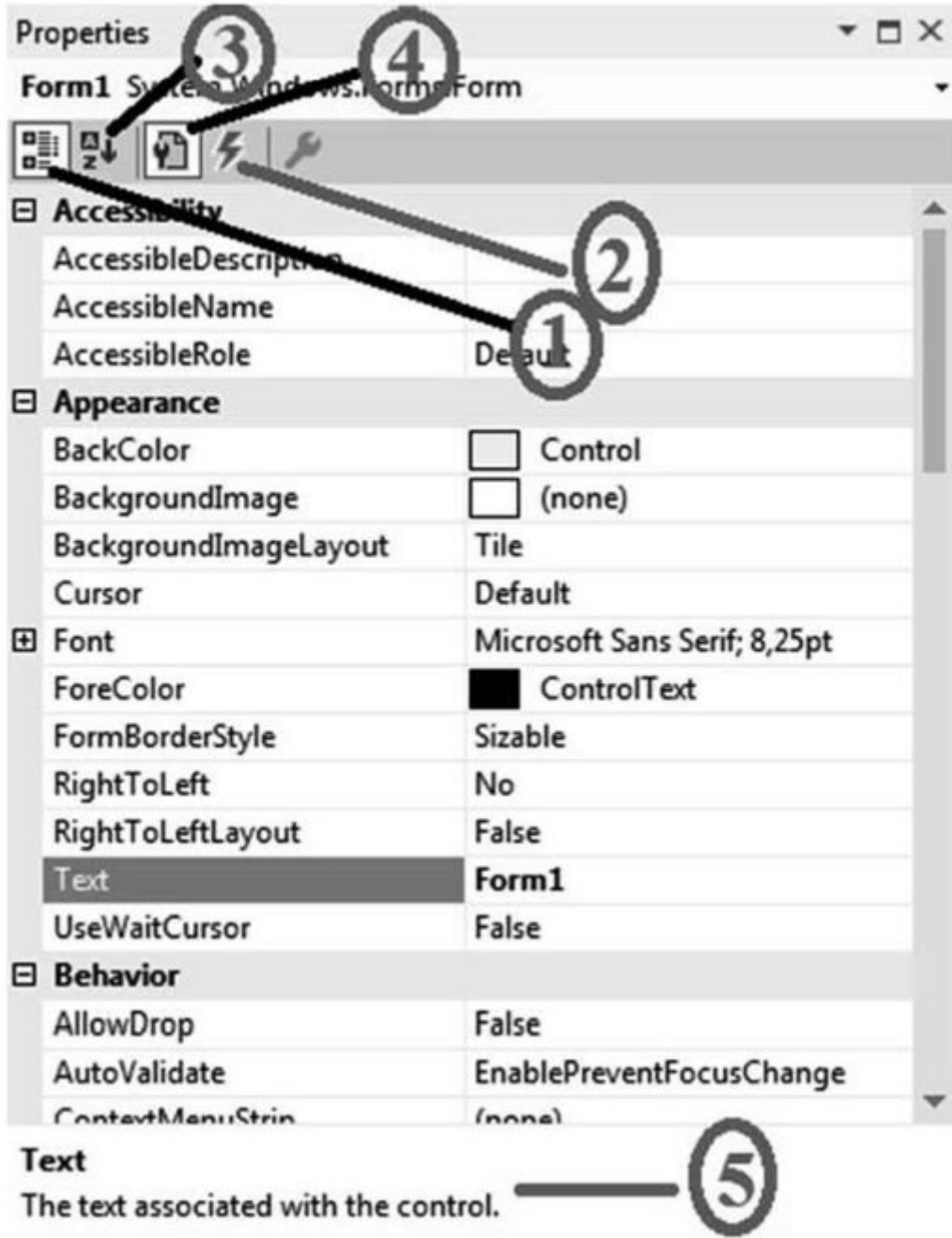


📷 Şekil 2.1. Ekran Pencerelerinin Konumları

Yukarıdaki şekilde gösterilen pencerelerin konumları, pencerenin üst başlık kısmından tutulup sürüklenerek ekranda çıkan konumlandırma işareti üzerine bırakılır ve oraya yerleşir. Ya da daha önceki pencerelerin üzerine bırakılarak onlar için yeni bir sekme olarak konumlanması sağlanır. Bir pencerenin üst başlık kısmı tutulup sürüklenince ortaya çıkan ekran görüntüsü aşağıdaki şekilde gösterilmiştir.

2.1.2. Properties (Özellikler)

Seçili olan kontrolün özelliklerini kod yazmadan kolayca değiştirmek amacıyla kullanılan penceredir. Buradaki yapılan değişiklikler kalıcı olarak değiştirilir ve form yüklenince ayarlanan şekilde gösterilir. Aşağıdaki şekilde 1 numara ile gösterilen buton aracılığıyla buradaki özellikler benzerlerine göre gruplandırılır. 2 numaralı buton ise seçili kontrolün olaylarının (events) listelendiği ekranın görünmesini sağlar. 3 numara ise özelliklerin A'dan Z'ye sıralı olarak gösterilmesine imkân verir. 4 numaralı buton ise olayların gösterilmesinden özelliklerin görüntülenmesine geçişi sağlar. 1 ve 3 numaralı butonlar olaylar ekranında da aynı işlemleri yerine getirir. Seçili özellik hakkında kısa bilgi ise ekranın sol alt köşesindeki 5 numaralı yerde gösterilir.



Şekil 2.6. Properties Penceresi

- **ImageList:** Görüntüleri depolamak için kullanılır ve daha sonra kontroller aracılığıyla görüntülenebilir. Bir resim listesi, tek ve tutarlı bir resim kataloğu için kod yazmaya izin verir.
- **Label:** Kullanıcı tarafından düzenlenemeyen metin veya resimleri görüntülemek için kullanılır. Bir formdaki nesneleri tanımlamak için kullanılırlar; örneğin, belirli bir kontrolün tıklandığında ne işe yapacağını ifade eden kısa bir etiket veya başlık olarak kullanılır.

- **LinkLabel:** Windows Forms uygulamalarına web sayfası stili özelliklerine sahip bağlantılar eklemeye olanak tanır. Label kontrolünde kullanılabilen her şey bu kontrolde de kullanılabilir. Metnin bir bölümü bir nesnenin veya web sayfasının bir bağlantısı olarak da ayarlanabilir.
- **ListBox:** Kullanıcının bir veya daha fazla seçebileceği öğelerin listesini görüntüler.
- **ListView:** Simgeleri içeren öğelerin listesini görüntüler. Windows Gezgini'nin sağ bölümü gibi bir kullanıcı arayüzü oluşturmak için liste görünümü kullanılır.
- **MainMenu:** Form üzerinde bir menü görüntüler.
- **MaskedTextBox:** Metin kutuları, kullanıcıdan giriş yapmak veya metni görüntülemek için kullanılır. TextBox denetimi genellikle salt okunur hale getirilebilmesine rağmen, düzenlenebilir metin için kullanılır. Metin kutuları birden çok satır görüntüleyebilir, metni kontrolün boyutuna sığdırır ve temel biçimlendirmeyi ekler. MaskedTextBox kontrolü, görüntülenene veya formdan girilen metin için tek bir format sağlar. Böylelikle doğru ve yanlış kullanıcı girişi kullanılan bir maske yardımıyla ayırt edilir.
- **TextBox:** Metin kutuları, kullanıcıdan giriş yapmak veya metni görüntülemek için kullanılır. TextBox denetimi genellikle salt okunur hale getirilebilmesine rağmen, düzenlenebilir metin için kullanılır. Metin kutuları birden çok satır görüntüleyebilir, metni kontrolün boyutuna sığdırır ve temel biçimlendirmeyi ekler. TextBox denetimi, görüntülenene veya kontrolde girilen metin için tek bir format sağlar.
- **MenuStrip:** Bu kontrol, uygulama komutlarını ve işlevlerini gruplayarak kolayca erişilebilir hale getirir.
- **AutoCompleteSource:** Otomatik tamamlama için kullanılan dizi kaynağını belirten bir değer alır.
- **AutoSize:** Bu özellik `false` yapılırsa, Label öğesinin yüksekliği ve genişliği manuel olarak ayarlanabilir. `true` olması durumunda ise otomatik olarak metin kadar uzunluk ve yüksekliğe sahip olur.
- **BackColor:** Kontrolün arka plan rengini seçilen renk olarak değiştirir.
- **BackgroundImage:** Kontrolün arka planında seçilen resmin görüntülenmesini sağlar.
- **BackgroundImageLayout:** Kontrolün arka planında seçilen resmin nasıl görüntüleneceğini (ortalı, yayılmış, gerçek boyut vb.) ayarlar.
- **BorderStyle:** Label gibi kontrollerin çerçevesinin nasıl görüneceğini ayarlar.
- **Bottom:** Kontrole ait alt kenar ile formun alt kenarı arasındaki uzaklığı piksel cinsinden ayarlar.
- **Bounds:** Üst kontrolün boyutunu ve konumunu, ana kontrole göre piksel olarak ayarlar.

- **CanFocus:** İmlecin kontrol üzerinde olup olmadığını belirler.
- **CanRaiseEvents:** Kontrolde olayların artırılıp artırılamayacağını belirler.
- **CanSelect:** Kontrolün seçilip seçilemeyeceğini gösterir.
- **Capture:** Kontrolün fareyi yakalayıp yakalamadığını belirtir.
- **CausesValidation:** Kontrole imleç odaklandığında, doğrulama gerektiren herhangi bir denetimin gerçekleştirilmesine izin verilir verilmeyeceğini belirten bir değer ayarlar.
- **CharacterCasing:** TextBox denetiminin, yazılan karakterlerin sürekli büyük harf veya sürekli küçük harf olup olmayacağını ayarlar.
- **CheckForIllegalCrossThreadCalls:** Bir işlem yapılırken aynı anda bir başka işlemin çalışmasını hata olarak gösterip göstermeyeceğini ayarlar.
- **CausesValidationChanged:** İlgili kontrolün `CausesValidation` özelliği değiştirildiğinde çalışan metottur.
- **ChangeUICues:** İmleç odaklandığında veya klavye kullanıcı arayüzündeki işaretler değiştirildiğinde çalışan metottur.
- **Click:** İlgili kontrolün fare sol tuşu ile tıklanması ya da imleç üzerinde aktif iken Enter tuşu ile etkinleştirilmesi sonucu çalışan metottur.
- **ClientSizeChanged:** İlgili kontrolün `ClientSize` özelliği değiştirildiğinde çalışan metottur.
- **ContextMenuChanged:** İlgili kontrolün `ContextMenu` özelliği değiştirildiğinde çalışan metottur.
- **ContextMenuStripChanged:** İlgili kontrolün `ContextMenuStrip` özelliği değiştirildiğinde çalışan metottur.
- **ControlAdded:** İlgili kontrolün altına yeni bir kontrol eklendiğinde çalışan metottur.
- **ControlRemoved:** İlgili kontrolün altındaki bir kontrol silindiğinde çalışan metottur.
- **CursorChanged:** İlgili kontrolün `Cursor` özelliği değiştirildiğinde çalışan metottur.
- **DockChanged:** İlgili kontrolün `Dock` özelliği değiştirildiğinde çalışan metottur.
- **DoubleClick:** İlgili kontrolün fare sol tuşu ile çift tıklanması ya da imleç üzerinde aktif iken Enter tuşu ile iki kez arka arkaya etkinleştirilmesi sonucu çalışan metottur.
- **DpiChangedAfterParent:** Bir kontrol için DPI ayarı, ana kontrolün veya formun DPI'sının değişmesinden sonra programsal olarak değiştirildiğinde çalışan metottur.

- **DpiChangedBeforeParent:** Bir kontrol için DPI ayarı, ana kontrolün veya formun DPI'sının değişmesinden önce programsal olarak değiştirildiğinde çalışan metottur.
- **DragDrop:** Sürükle ve bırak işlemi tamamlandığında çalışan metottur.
- **DragEnter:** Herhangi bir kontrol, bu kontrolün sınırlarına sürüklendiğinde çalışan metottur.
- **DragLeave:** Herhangi bir kontrol, bu kontrolün sınırlarına sürüklendikten sonra sınırlarından ayrıldığında çalışan metottur.

Değişkenler ve Veri Türleri aşağıda ki tabloda verilmiştir.

Ad	Boyut(Byte)	Aralık
byte	1	0 - 255
short	2	"-32768" - "32767"
int	4	"-2.147.483.628" - "2.147.483.647"
long	8	"-9.223.372.036.854.775.808" - "9.223.372.036.854.775.807"
sbyte	1	"-128" - "127"
ushort	2	"0" - "65.535"
uint	4	"0" - "4.294.967.295"
ulong	8	"0" - "18.446.744.073.709.551.615"
float	4	"+1.5x10 ⁻⁴⁵ " - "+3.4x10 ³⁸ "
double	8	"+ 5.0x10 ⁻³²⁴ " - "+1.7x10 ³⁰⁸ "
object		
char	2	Bir Unicode Karakter
string		Karakterlerin Tümü
decimal	16	"+1.0x10 ⁻²⁸ " - "+7.9x10 ²⁸ "
bool	2	True veya False