

WINDOWS FORMLARI VE ÖZELLİKLERİ



İÇİNDEKİLER

- Windows Formlar
- Form Özellikleri



HEDEFLER

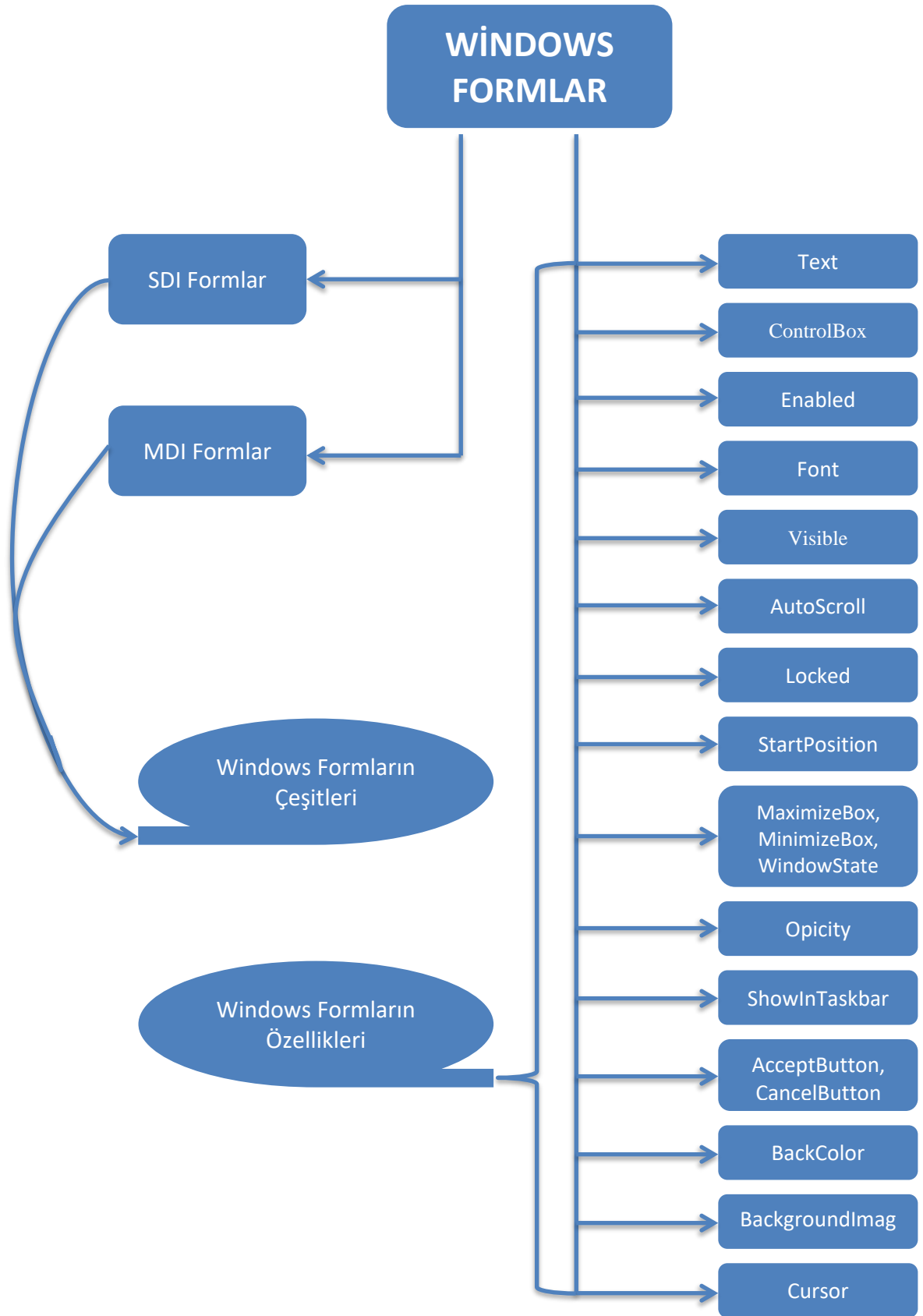
- Bu üniteyi çalıştıktan sonra;
- SDI form ve MDI form arasındaki farklılıkları ayırt edebilecek,
- Projenize yeni formlar ekleyebilecek,
- Formlarınızın özelliklerini istediğiniz ölçüde değiştirebileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

**GÖRSEL
PROGRAMLAMA I**
Öğr. Gör.
Daha ORHAN

**ÜNİTE
3**



GİRİŞ

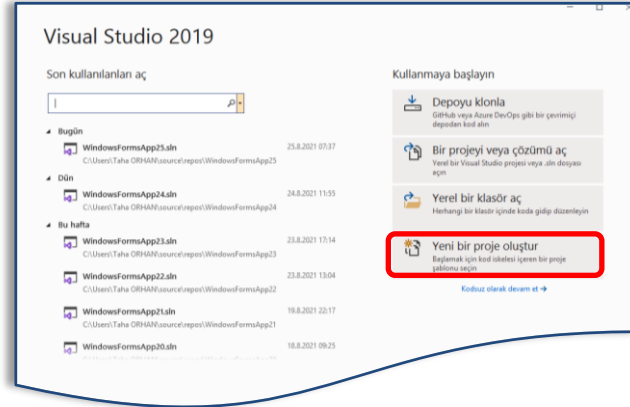
Bilgisayar donanımı ile kullanıcı arasındaki iletişim, yazılımlar sayesinde gerçekleştirilmektedir. Yazılımlar programcılar tarafından oluşturulurken görsel bir arayüze sahip olmadan da tasarlanabilmektedir. Ancak programların bilgisayar kullanıcılarının tamamına yönelik olabilmesi için görsel bir arayüze ihtiyaç duyulmaktadır. Bu doğrultuda görsel programlama dillerini kullanmak programcıların projelerini geliştirmesi noktasında kolaylıklar sağlamaktadır. Ayrıca bilgisayarda hazırlanan programların daha kolay bir şekilde tasarlanabilmesi ve anlaşılabilir olması açısından görsel programlama dilleri tercih edilmektedir.



Formlar birer kontrol nesnesidir.

Bilgisayar donanımı ile programcı arasındaki iletişimi sağlayan birçok programlama dili bulunmaktadır. C# bu programlama dillerinden yalnızca bir tanesidir. Microsoft firması tarafından geliştirilen C#, temel seviyedeki programcılara yönelik olduğu gibi ileri düzey uygulamalar geliştirmeye de imkan sağlayabilecek potansiyele sahip nesne yönelimli programlama dilidir.

Visual Studio 2019 programı başlatıldığında Şekil 3.1.'de gösterilen pencere ekrana gelecektir. Bu pencerede **Yeni Bir Proje Oluştur** seçeneğini işaretledikten sonra **C#**, JavaScript, Python gibi farklı programlama dilleri ile geliştirebileceğimiz farklı türdeki (Konsol uygulamaları, Web uygulamaları, Windows form uygulamaları vb.) seçeneklerin bulunduğu bir alan bizi karşılamaktadır.



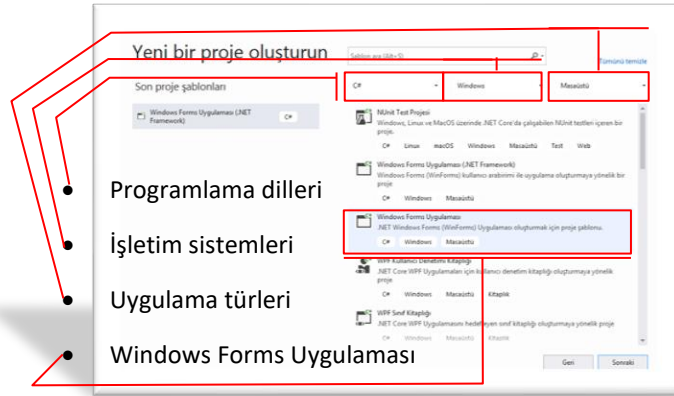
Şekil 3.1. Visual Studio 2019 Programının Açılış Sayfası

Windows tabanlı C# projeleri genel itibariyle formlar, bu formlar üzerine araç kutusundan eklenen kontrol nesneleri ve nesnelere ait olay prosedürlerinden oluşmaktadır. Kendisi de bir nesne olan **formlar**, Windows tabanlı uygulamaların en önemli seçenekleri arasında yer almaktadır. Form nesnesi ve araç kutusundan formlar üzerine eklediğimiz diğer kontrol nesneleri `System.Windows.Forms.Control` sınıfından türetilmişlerdir.

Bu ünite de Visual Studio tümleşik geliştirme (IDE) ortamında sıklıkla kullanacağımız **form** yapısı üzerinde durulmaktadır. Konu akış sıralamasında ağırlıklı olarak: formların projeye nasıl eklendiği, Windows formların çeşitleri ve formların sıklıkla kullanılan özelliklerinden bahsedilecektir.

WINDOWS FORMLAR

C# programlama dili ile Windows tabanlı bir uygulama gerçekleştirmek için Visual Studio 2019 programını başlattıktan sonra sırasıyla *Yeni Bir Proje Oluştur* seçeneğini işaretleyerek karşımıza gelen ekrandan *Windows Forms Uygulaması (.Net Framework)*'ü seçmemiz gerekmektedir. Yeni bir proje oluşturur penceresinde geliştirmek istediğiniz uygulama türlerini ararken, Şekil 3.2.'de gösterilen filtreleme alanlarını kullanmak programcılara kolaylıklar sağlamaktadır. Geliştirmek istediğimiz uygulama türünü işaretledikten sonra sağ alt köşede bulunan *Sonraki* butonuna tıklamamız gerekmektedir. Bu işlemin ardından projemizin *Proje adı*, *Konum*, *Çözüm adı* ve *Altyapı* özelliklerini ayarlayacağımız bir alan bulunmaktadır. Bu alan üzerinde gerekli değişiklikleri yaparak *Oluştur* butonuna tıkladığımızda *Form1* varsayılan ismi ile hazır bir Windows form yapısı projemizde düzenlemeye hazır olarak ekrana gelecektir.

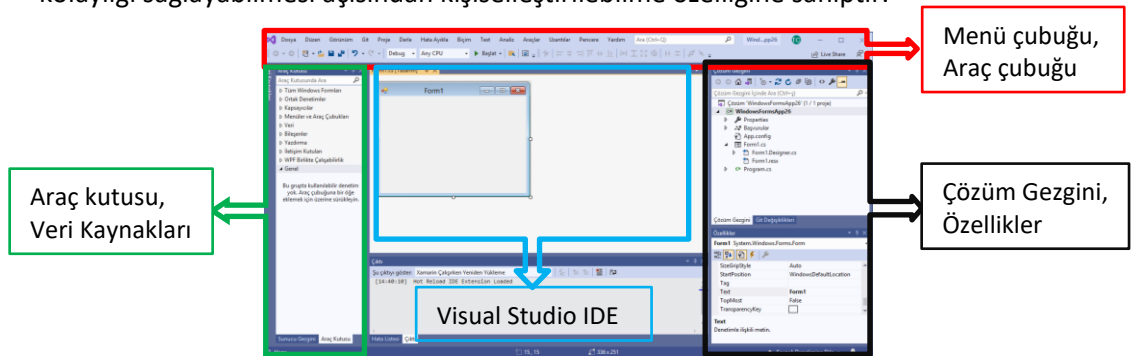


Şekil 3.2. Yeni Proje Oluşturun Penceresi



Windows tabanlı uygulamalar SDI ve MDI olmak üzere iki farklı form çeşidini kullanabilir.

Kısaca programda kullanacağımız alanları tanıttık olursak Şekil 3.3.'de gösterildiği gibi ekranımızın üst kısmında *menü çubuğu* ve *araç çubuğu* bulunmaktadır. Ekranın sol kısmında projemize ekleyebileceğimiz ve ilgili bağlantıları yapabileceğimiz seçeneklerin bulunduğu *Araç Kutusu* ve *Veri Kaynakları* bulunmaktadır. Ekranın sağ tarafından *Çözüm Gezgini* ve *Özellikler* kayan panelleri yer almaktadır. Kullanıcı arayüzünü oluşturacağımız ve bu ünite üzerinde duracağımız Windows form çalışma alanımız ekranın orta kısmında konumlanmaktadır. Programda belirtilen bu alanlar programcı tarafından kullanım kolaylığı sağlayabilmesi açısından kişiselleştirilebilir özelliğe sahiptir.



Şekil 3.3. Visual Studio 2019 Uygulama Geliştirme Ortamı

Windows formlar genel olarak iki ana başlık altında incelenmektedir. Bunlar SDI (Single Document Interface) formlar ve MDI (Multiple Document Interface) formlardır. Yani tek formlu arayüzler ve çok formlu arayüzlerdir. Bunlar “ana” ve “çocuk form” olarak da adlandırılmaktadırlar. Bu terimlere SDI ve MDI formların anlatıldığı başlıklarda yer verilecek olup neden bu şekilde adlandırıldıkları daha iyi anlaşılacaktır.

SDI Formlar



SDI formlar tek belgeli arayüze sahip formlardır.

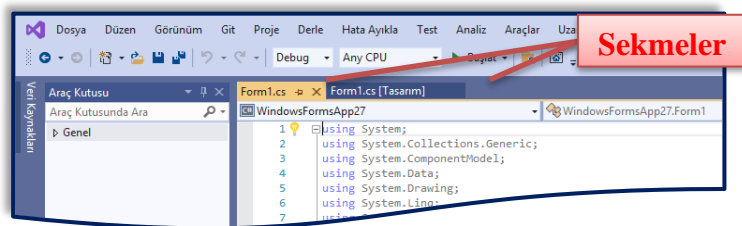
SDI formlar İngilizce “Single Document Interface” kelimelerinin baş harfleri kullanılarak isimlendirilen tek belgeli arayüze sahip formlardır. Başka bir forma ihtiyaç duymaksızın tek başına çalışabilen bir yapıdadırlar. Yeni başlattığımız bir projede ilk olarak ekrana gelen formlar SDI formlardır. Şekil 3.3.’de de proje başlatıldığında ekrana gelen form SDI formudur.

Formlar üzerinde çalışırken sadece tasarım alanında işlem yapmak, istediğimiz uygulamaları geliştirmek için yetersiz kalabilir. Bu nedenle projemizde açık olan formlardan o anda aktif olan formun kod kısmını görüntülemek için **görünüm** menüsünden Şekil 3.4.’de gösterildiği gibi **kod** seçeneğini seçebiliriz. Bu menü yolunu takip ettikten sonra projemizdeki her bir form için ayrı ayrı hazırlanan sınıf bloğunun bulunduğu alanı görebiliriz.



Şekil 3.4. Kod Editörüne Geçiş İçin Menü Yolu

Kod editörüne geçiş yapabilmek için farklı seçenekler de bulunmaktadır. Kullanıcı tercihinine bağlı olarak kısayol tuşu olan **F7** fonksiyon tuşu da kullanılabilir. Ayrıca ilgili tasarım ekranındayken form üzerinde farein sol tuşu ile çift tıklama işleminin gerçekleştirilmesiyle formun yüklenme olayına ait kod bloğunun açılması sağlanmaktadır. Ayrıca tasarım alanı ve kod editörü arasındaki geçişler için Şekil 3.5.’de olduğu gibi sekmelere tıklayarak da bu işlemi gerçekleştirebilirsiniz. Bu tarz işlemlerde programcının hızlı ve kolay bir şekilde işlemlerini gerçekleştirebilmesi programı iyi tanımasına bağlı olarak değişiklik gösterebilmektedir.

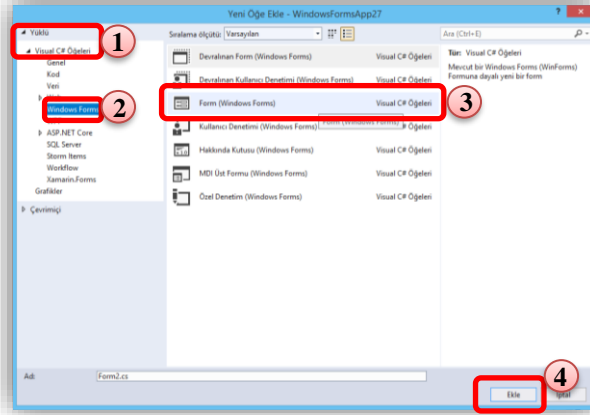


Şekil 3.5. Sekmelerin Kullanımı



Yeni öge ekle
penceresinin kısayol
tuşu: Ctrl+Shift+A

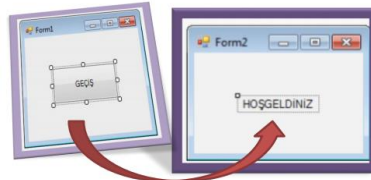
Projelerimizi başlattığımızda bizim karşımıza gelen SDI form ve kod editöründe yaptığımız değişikliklerin yeterli olmadığı durumlarda yeni formlar eklenebilmektedir. Bu durumda projemize yeni bir form ekleyebilmek için **yeni öge ekle** penceresini açmamız gerekmektedir. Bu işlem **ctrl+shift+a** tuş kombinasyonu kullanılarak da yapılabilir. Tasarım ekranından kod editörüne geçiş işleminde olduğu gibi bu işlemde de farklı yöntemler mevcuttur. Bu tarz alternatifli seçeneklerden artık bahsedilmeyecek olup hız kazanma noktasında araştırma yapma tercihi sizlere bırakılmıştır.



Şekil 3.6. Yeni Öge Ekle Penceresi

Yeni öge ekle penceresini açtıktan sonra eklemek istediğimiz öğeyi bulmak için Şekil 3.6.'da gösterildiği gibi sol tarafta bulunan paneli kullanmak bize kolaylık sağlayacaktır. Bu panel üzerinde Windows Forms seçeneğine tıkladıktan sonra Form (Windows Forms) seçeneğini işaretleyerek ekle butonu ile projemize yeni bir SDI form ekleyebiliriz. Eklediğimiz yeni form program üzerinde görünmekle beraber Çözüm Gezini içinde de yer almaktadır. Programımızı başlattıktan sonra projemizde var olan formlar arasında geçiş yapmaya ihtiyaç duyabiliriz. Bu durumda aktif olan form içinde etkileşimin gerçekleşmesini istediğimiz alana, geçiş yapılması istenen formun **Show** metodu kullanılarak geçilebilmektedir. Bu şekilde hazırlanmış iki formun bulunduğu bir uygulamada ilk formun içinde bulunan butonun tıklanma olayına yazılan kod aşağıda verilmiştir. Buton tıklandıktan sonra elde edilen ekran görüntüsü de Şekil 3.7.'de gösterilmektedir.

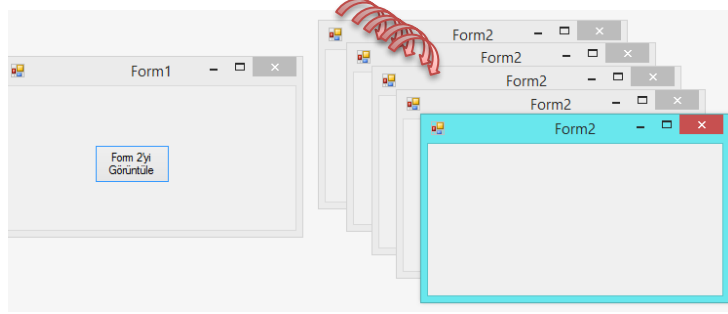
```
Form2 ikinci_form = new Form2();
ikinci_form.show();
```



Şekil 3.7. Show Metodu ile İkinci Bir Formu Ekranda Görüntüleme

Show metodu kullanılarak geçiş işleminin gerçekleştirildiği uygulamada dikkat edilecek olursa; iki formunda ekranda görüntülendiği durumda pencere

başlıkları tıklanarak istenilen form aktifleştirilebilir. Bununla beraber ilk form üzerine yerleştirilen butona tıklandığında Form2 yeniden açılacak ve birden çok form ekran üzerinde Şekil 3.8.'de gösterildiği gibi basamaklar halinde görüntülenecektir.



Şekil 3.8. Show Metodunun Olduğu Butona Birden Fazla Tıklama

Şekil 3.8.'deki ekran görüntüsüne dikkat edecek olursak formlar basamak hâlinde yerleşmişlerdir. Bu yerleşim hâline cascade adı verilmektedir.

Açılan form kapatılmadan diğer formun pencere başlığına tıklanarak geçilmesini engellemek istiyorsak *ShowDialog* metodunu kullanmamız gerekmektedir. Bu durumda buton içine yazılan kodu aşağıdaki gibi düzenlemeliyiz.

```
Form2 ikinci_form = new Form2();
```

```
ikinci_form.showdialog();
```

Çalışma anında bir formun görüntülenmesini sağlayabileceğimiz gibi gizli olmasına da imkan tanınmaktadır. Bu işlem *Hide* metodu kullanılarak gerçekleştirilmektedir. İki formun bulunduğu uygulamamızda ilk formda bulunan butona tıkladığımız zaman ikinci form açılmadan önce ilk formun gizli olması için kod editörünü aşağıdaki gibi düzenlememiz gerekmektedir.

```
this.Hide();
```

```
Form2 ikinci_form = new Form2();
```

```
ikinci_form.showdialog();
```

Formların görünür olup olmama durumu *Visible* metodu kullanılarak da ayarlanabilmektedir. Ancak burada karıştırılmaması gereken bir durum bulunmaktadır. *Kapatma* ve *gizleme* işlemleri aynı şeyi ifade etmezler. Gizlediğiniz formlar arka planda çalışırken sadece ekran üzerindeki görünürlüğü kaldırılır. Kapatma işlemini gerçekleştirdiğiniz formlar da ise, üzerinde yapılacak işlemleri sonlandırma ve hafızadan silme olayı gerçekleştirilmiş olur. Bu yüzden kapatma işlemini, uygulamamızın başlangıç formunda gerçekleştirirken dikkatli olmamız gerekmektedir. Başlangıç formunda *Close* metodu kullanıldığı zaman C# .Net uygulamayı sonlandıracaktır.

İki formun bulunduğu uygulamamızı, ikinci form üzerine yerleştireceğimiz butonla biraz daha genişletelim. İlk formda bulunan buton üzerinde değişiklik yapmadan ikinci formda bulunan butonun içine aşağıdaki kodları yazalım. Böylece



Gizleme Hide metodu ile, Kapatma Close metodu ile gerçekleştirilir.

ilk formda bulunan butona tıklayarak ilk form *gizlenip* ikinci forma geçilmiş olacaktır. İkinci formda bulunan butona tıklandığı zaman da ikinci form *kapatılıp* ilk form görünür hale getirilecektir.

```
this.Close();

Form1 ilk_form = new Form1();

ilk_form.visible = True;
```

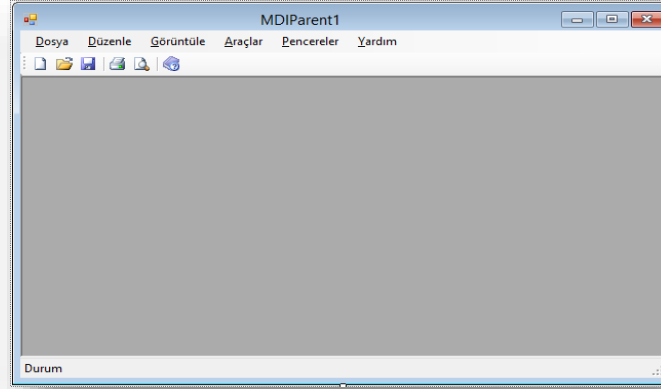
MDI Formlar



MDI formlar, çoklu doküman arayüzüne sahip formlardır.

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

Projemize MDI form eklemek istediğimiz zaman SDI formlarda olduğu gibi *Ctrl+Shift+A* tuş kombinasyonunu kullanarak *Yeni Öğe Ekle* penceresini açmamız gerekmektedir. Bu alan üzerinden *MDI Üst Formu (Windows Forms)* seçeneğini işaretleyerek ekle butonuna basmamız yeterli olacaktır. MDI Üst Formları projemize eklediğimiz zaman Şekil 3.9.'da olduğu gibi görünmektedir.



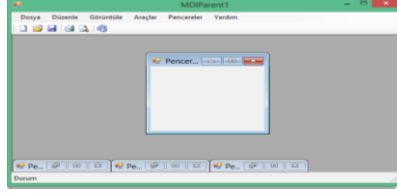
Şekil 3.9. MDI Üst Formu (Windows Forms) Ekran Görüntüsü

Ana pencere olan MDI formuna, parent (ebeveyn) form denir. Bu ana form içinde açılan formlara ise child (çocuk) form denilmektedir. Şayet projemizde parent form mevcut ve bununla beraber child formumuzda var ama birbirinden bağımsız çalışıyorsa kendi içinde olduğunu tanıtmamışızdır. Bu hususa dikkat etmenin faydalı olacağını düşünülmektedir. Birazdan inceleyeceğimiz örnekte bunu daha iyi kavrayacaksınız.

Windows tabanlı C# projelerinde Windows Forms Uygulaması (.Net Framework) türünde bir uygulama geliştirmek istediğimizde, projemiz başlangıç olarak SDI formla bizi karşılamaktadır. Bu durumda SDI formu MDI forma dönüştürmek için: Form seçildikten sonra sağ tarafta bulunan Özellikler panelindeki *IsMdiContainer* özelliğini *True* olarak değiştirmemiz yeterli olacaktır.

Örnek

- Şekil 3.10'daki MdiParent form içinde bulunan yeni butonuna tıklayarak child formların nasıl görüntülendiğinin örneği aşağıdaki gibidir.



Şekil 3.10. MDI Form ve İçinde Açılan Formlar

```
private void ShowNewForm(object sender, EventArgs e)
{
    Form childForm = new Form();
    childForm.MdiParent = this;
    childForm.Text = "Pencere " + childFormNumber++;
    childForm.Show();
}
```

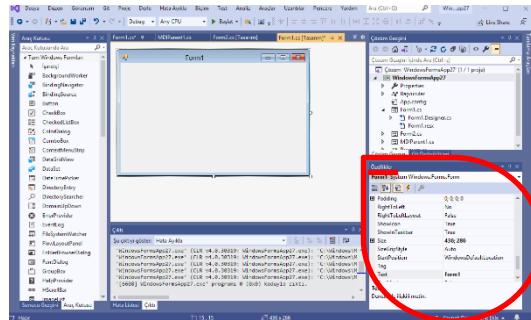
Örnekteki kod bloğunu incelediğimiz zaman ilk satır yeni açılacak olan formların tanımlanmasını sağlamaktadır. İkinci satır ise açılacak olan formların MDI form içinde hareket etmesini ve ona bağlı olarak çalışmasını sağlamaktadır. İkinci satırdaki kod yazılmadığında, açılacak olan formları simge durumunda küçülttüğümüzde görev çubuğuna yerleşirler. Üçüncü satırdaki kodlar açılacak olan formların pencere başlığını ayarlamaktadır. Son satırda bulunan kod ise formun görüntülenmesini sağlamaktadır.



Özellikler paneli, seçili olan nesnenin özelliklerini listelemektedir.

FORM ÖZELLİKLERİ

Visual Studio ortamında ekranın sağ alt tarafına yerleşen **Özellikler** paneli üzerinden, çalışma esnasında aktif olan nesneye ait özellikleri görüntüleyebilir ve değiştirebiliriz. Bu alan üzerinden formumuza ait özellikleri değiştirme imkânına sahibiz. Aynı zamanda forma ait özellikler kod editöründen de değiştirilebilmektedir. Özellikler panelinin ekran görüntüsü şekil 3.11.'de gösterilmektedir.

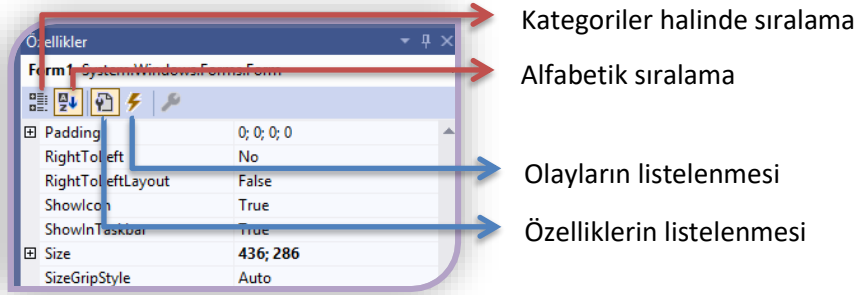


Şekil 3.11. Visual Studio Platformunda Özellikler Paneli



Özellikler paneli
ekranda görünmüyorsa
F4 fonksiyon tuşu
kullanılarak ekrana
getirilebilmektedir.

Özellikler paneli iki farklı görünümde nesnelere ait özellikleri listeleyebilir. Bunlardan biri **Alfabetik** seçeneğidir. Bu seçenek ile aktif olan kontrolün özellikleri panel içinde alfabetik olarak sıralanır. Diğer bir listeleme şekli ise **Kategorilere Ayrıldı** seçeneğidir. Bu seçenek ise özelliklerin benzer yönlerine göre ortak gruplar hâlinde (Davranış, Diğer, Erişilebilirlik vb.) birleştirilerek listelemek için kullanılır. Bu listeleme yöntemleri Şekil 3.12.'de gösterilen butonlar yardımı ile değiştirilebilmektedir. Özellikler panelinin görüntülenme tercihi programcıya ait bir durumdur. İstenilen özelliğe, listeleme seçeneklerinden hangisi ile daha hızlı ulaşılabilirse o yöntem tercih edilebilir.

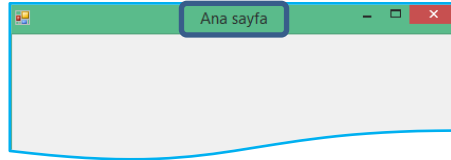


Şekil 3.12. Özellikler Paneli Listeleme Seçenekleri

Program çalıştırıldığında **Özellikler** paneli ekranda görünmüyorsa **F4** fonksiyon tuşu kullanılarak ekrana getirilebilmektedir. Bu üniteye Özellikler panelindeki bütün özellikler yerine, **formlarda** sıklıkla kullanılan özelliklere yer verilecektir.

Text Özelliği

Formun başlık çubuğunda görüntülenen isminin ayarlandığı özellik seçeneğidir. Bu özelliği kullanarak projemizde var olan bir formun isim değişikliği yapılabilir ve işlemin görseli Şekil 3.13.'de gösterilmektedir. Ancak burada dikkat etmemiz gereken nokta bu isim sadece projemizin tasarım kısmındaki ismidir. Text özelliği haricinde bir değişiklik yapılmadıysa kod editöründe çalışırken formumuzun ismi, programın varsayılan olarak atadığı (Form1, Form2 vb.) isimdir. Şayet kod alanında da sizin yazdığınız ismin olmasını istiyorsanız **"name"** özelliğini değiştirmeniz gerekmektedir.



Şekil 3.13. Text Özelliği Ana Sayfa Olarak Hazırlanmış Form

Name özelliğini değiştirirken isimlendirme kurallarına uygun olarak bu işlemi gerçekleştirmek, programınızın çalışma esnasında ve hataları ayıklama noktasında size kolaylık sağlayacaktır. İsimlendirme işlemi gerçekleştirilirken:

- Sayı ile başlamamaya,

- Boşluk ve özel karakterler (?,!,...) kullanmamaya,
- Programlama dilinde kullanılan (for, and, not,vb.) anahtar kelimeleri kullanmamaya,
- Türkçe karakterler (ç,ç,ğ, vb.) kullanmamaya,

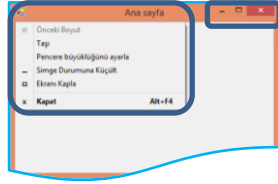
dikkat edilmelidir. Ayrıca isimlendirmeler yapılırken belirli bir yazım stili kullanmak programın okunabilirliğini de olumlu yönde etkileyecektir. Örneğin iki kelimeden oluşan isimlendirme işlemlerinde her kelimenin baş harfi büyük yazılarak anlamlı hale getirilebilir.

ControlBox Özelliği

Bu özellik program çalışırken anlaşılabilir bir yapıdadır. Mantıksal olarak değer alabilen iki seçenek söz konusudur. Bunlardan biri **True** (doğru veya 1) diğeri ise **False** (yanlış veya 0) değeridir. True değeri geçerli olduğu zaman programı çalıştırıp formun başlık çubuğunda sol üst köşeye tıklayarak bir kontrol menüsünün açılacağı şekil 3.14.'de görülmektedir. Ancak ControlBox özelliği False olarak ayarlanırsa bu kontrol menüsü açılmamaktadır. Bunun yanı sıra sağ üst köşede bulunan (simge durumuna, tam ekran, kapat) butonların olmadığı da Şekil 3.15.'de görülmektedir. Böyle bir durumda çalışan programı kapatmak için **görev çubuğu** üzerinde bulunan programını ikonun üstünde farenin sağ tuşu ile tıklandıktan sonra kapat komutunun verilmesi gerekmektedir.



Enabled özelliğinin True olması durumunda form ve üzerindeki nesneler aktif konumda olur.



Şekil 3.14. ControlBox = True



Şekil 3.15. ControlBox = False

Enabled Özelliği

Formların Enabled özelliği, Controlbox özelliğinde olduğu gibi True ve False şeklinde iki farklı değer almaktadır. **Enabled** özelliğinin **True** olması durumunda form aktif konumdadır ve formun üzerindeki nesneler ile işlem yapılması mümkündür. Formun enabled özelliğini false yapacak olursak bu defa form pasif konuma geçecektir. Bununla birlikte forma yerleştirdiğimiz nesneler de aynı şekilde pasif konumda olacaktır. Bu sonuçtan ötürü forma ait enabled özelliğine false değeri atanırken nesnelerin proje aşamasındaki işlevsellikleri gözden geçirilmelidir. Formun Enabled özelliğinin true ve false olma durumlarına yönelik ekran görüntüsü şekil 3.16.'te gösterilmektedir. Aynı zamanda enabled özelliğine göre yapılacak işlemler Tablo 3.1.'de verilmiştir.



Şekil 3.16. Enabled Özelliği

Tablo 3.1. Enabled Özelliği

Enabled = True	Enabled = False
Nesneleri kontrol edebiliriz.	Nesneleri kontrol edemeyiz.
Textbox içine yazı yazabiliriz.	Textbox içine veri girişi yapamayız.
Checkbox'ları işaretleyebiliriz.	Onay kutularını işaretleyemeyiz.

Font Özelliği

Font özelliği ile form üzerindeki nesnelere ait **text özellikleri**, form üzerindeki **yazı stilleri**, **yazı font özellikleri** ve **yazı boyutu** istenilen ölçüde ayarlanmaktadır. Böylelikle form ve form üzerindeki nesneler belirlenen parametreler doğrultusunda görüntülenmektedir. Bu değişiklikleri gerçekleştirmek için font özelliğinin sağ tarafında bulunan “...” butonuna tıklamak gerekmektedir. Bu işlemin ardından şekil 3.17.’de görünen **Yazı Tipi** penceresi ekrana gelmektedir. Bu pencerede istenilen değişiklikler yapılarak tamam butonuna tıklanır.



Projemizi çalıştırdığımız esnada formun ekran üzerinde görünür olup olmama durumunu visible özelliği ile ayarlayabiliriz.



Şekil 3.17. Yazı Tipi Penceresi

Yazı tipi penceresinde yaptığımız değişiklikler form ve form üzerinde bulunan nesnelerin boyutlarını etkilemektedir. Bu değişiklikler tasarımı etkileyeceği için kontrol edilmesi gerekmektedir.

Visible Özelliği

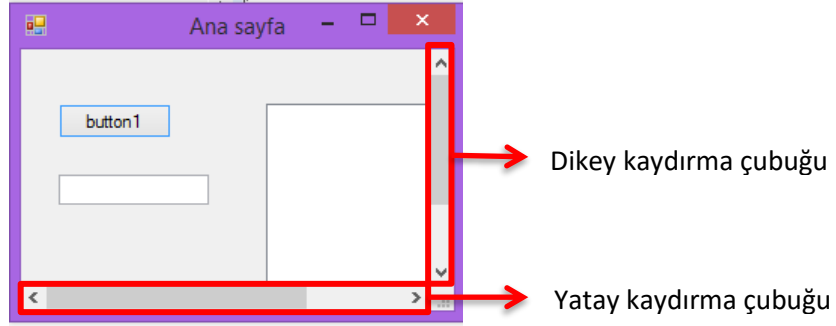
Projemizi çalıştırdığımız esnada formun ekran üzerinde görünür olup olmama durumunu belirleyen bir özelliktir. Mantıksal olarak iki farklı değer alabilir. Bu değerler daha önce aşına olduğumuz true ve false değerleridir.

Visible özelliği formlar için Visual Studio 2019’da özellikler panelinde bulunmamaktadır. Formun görünür olup olmama durumunu kod editöründen aşağıdaki kod satırını yazarak ayarlamamız gerekmektedir.

Fakat burada dikkat etmemiz gereken bir husus bulunmaktadır. Bu husus ise üzerinde çalıştığımız formun visible özelliğini değiştirmemektir. Örneğin form yüklenirken bu değeri false olarak değiştirmemiz çok makul olmayacaktır. Bundan dolayı program derlenirken hata verip çalışmayacaktır.

AutoScroll Özelliği

AutoScroll özelliği form üzerinde bulunan nesnelerin çok fazla olması durumunda veya kullanıcının form alanını tutamaçlar yardımıyla daralttığı durumda yatay ve dikey kaydırma çubuklarının aktif veya pasif olmasını sağlamaktadır. True veya false değerlerini alabilir. True olduğu durumda form üzerinde görünmeyen nesnelere kaydırma çubukları yardımıyla ulaşılabilir. Bu durum Şekil 3.18.'de gösterilmektedir.



Şekil 3.18. AutoScroll = True



StartPosition özelliği ile projeye ait başlangıç formunun ilk açılıştaki ekran üzerinde konumlanacağı alana yönelik ayarlamalar yapılmaktadır.

AutoScroll özelliğinin değeri false olarak ayarlandığında nesnelerin form üzerine yerleştirilmesine dikkat etmemiz gerekmektedir. Aksi takdirde görünmeyen nesneler üzerinde işlem yapılamayabilir. Böyle durumlarda form üzerinde bulunan nesnelerin **TabIndex** özelliklerine ait değer sıralaması da önem taşımaktadır.

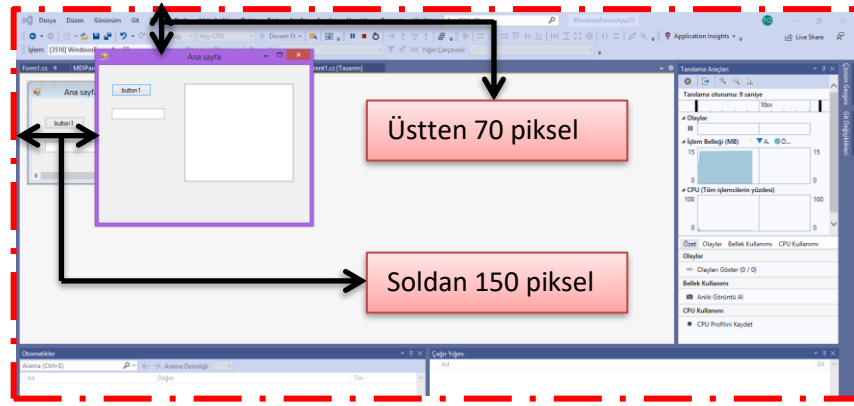
Locked Özelliği

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

StartPosition Özelliği

StartPosition özelliği ile projeye ait başlangıç formunun ilk açılıştaki ekran üzerinde konumlanacağı alana yönelik ayarlamalar yapılmaktadır. Bu işlem startPosition özelliğinin 5 farklı seçenek sunmasıyla gerçekleştirilmektedir. Bu seçenekler sırasıyla: Manual, CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds ve CenterParent seçenekleridir. Bu seçenekleri kullanarak projemizi çalıştırdığımız zaman formların nasıl konumlanacağını ve neler yapmamız gerektiğini sırayla inceleyelim.

Manuel: Bu seçeneği kullandığımız zaman özellikler paneli üzerinde bulunan location özelliğini de ayarlamamız gerekmektedir. Location kısmının sağ tarafında noktalı virgülle ayrılan iki adet sayı bulunmaktadır. Bunlardan ilki ekranımızda soldan uzaklığı, ikincisi ise üstten uzaklığı piksel cinsinden ifade etmektedir. Bu değerleri elimizle 150;70 girerek projemizi çalıştırdığımızda projenin başlangıç formu Şekil 3.19.'da olduğu gibi ekran üzerinde konumlanmaktadır.



Şekil 3.19. Formun Ekrandaki Sol Üst Köşeye Göre Uzaklıkları

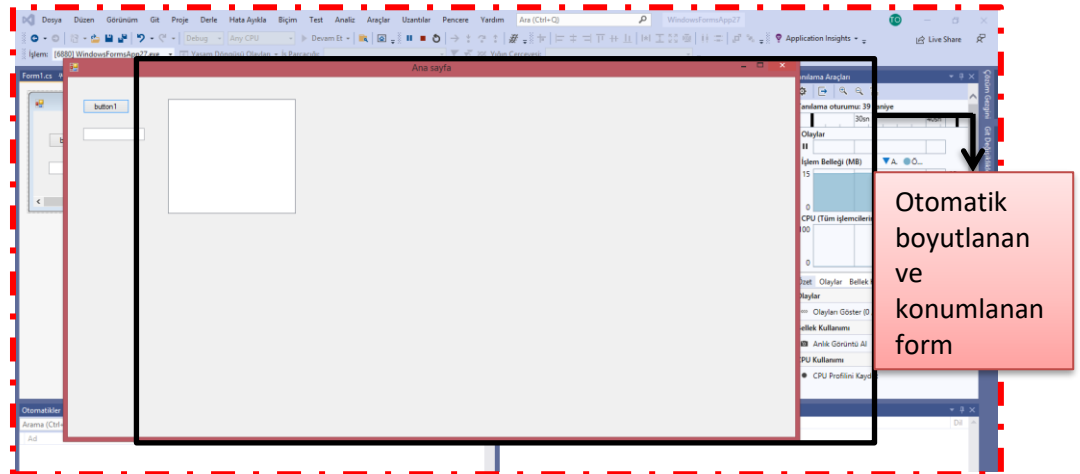


Başlangıç formunu ekranın ortasında konumlandırmak için Startposition özelliği centerscreen olarak ayarlanmalıdır.

CenterScreen: Bu seçeneği işaretleyerek projemizi başlatacak olursak formumuz ekranın ortasında konumlanmaktadır. CenterScreen seçeneği işaretlendikten sonra çalıştırılan proje ekran boyutlarına göre otomatik olarak ayarlanıp konumlanmaktadır.

WindowsDefaultLocation: Bu seçenek programın sunmuş olduğu varsayılan değerdir. Windows tarafından belirlen bir konumda proje başlatılmaktadır. Genellikle bu konum ekranın sol üst köşesidir.

WindowsDefaultBounds: Bu seçenek ile hazırlamış olduğumuz form Windows uygulamaları için standart bir boyutta varsayılan pozisyonda konumlanacaktır. WindowsDefaultBounds seçeneği işaretlendikten sonra çalıştırılan projenin ekran görüntüsü şekil 3.20.'de gösterilmektedir.



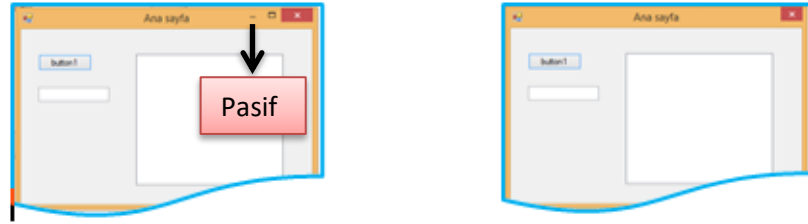
Şekil 3.20. Startposition Özelliği, Windowsdefaultbounds Seçeneği ile Ayarlanmıştır.

CenterParent: Bu seçenek işaretlendiğinde form, kendini çalıştıracak olan parent formun ortasında konumlanır. Birbirleriyle ilişkili olan MDI formlar için kullanışlıdır.

MaximizeBox, MinimizeBox ve WindowState Özellikleri

Projelerimizde hazırlanan formların çalışma esnasında tam ekranı kaplaması, simge durumunda küçülmesi veya görev çubuğunda görünmesiyle ilgili özellikler ayarlanabilmektedir. Bu durumda özellikler panelinde bulunan

MaximizeBox, MinimizeBox ve WindowState özelliklerini kullanmamız gerekmektedir. Bu üç farklı özelliğin tek başlık altında anlatılmasının sebebi arayüzdeki etkilerinden kaynaklanmaktadır. Örneğin MaximizeBox ve MinimizeBox özellikleri aynı anda False olarak değiştirilirse projemizi çalıştırdığımız zaman başlık çubuğunda bu butonlar görünmeyecektir. Bunlardan sadece birini false olarak ayarlarsak ekranda görünürler fakat false olarak ayarlanan özellik pasif olur ve üstünde herhangi bir işlem gerçekleştirilemez. Bu durum karşılaştırmalı olarak Şekil 3.21'da gösterilmiştir.



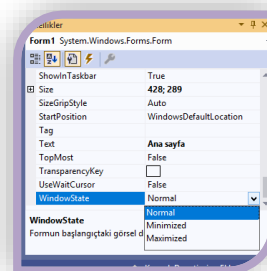
Şekil 3.21. Solda Sadece Minimizebox = False, Sağda Her İki Değer De False



MaximizeBox ve MinimizeBox özelliklerinin varsayılan değeri True olarak belirlenmiştir.

Visual Studio 2019 programında bir forma ait MaximizeBox ve MinimizeBox özellikler varsayılan değer olarak True seçeneği ile ayarlanmaktadır. Bu özellikler formun sağ üst köşesinde bulunan birçok Windows penceresinde olduğu gibi ekranı kapla ve simge durumuna küçült seçeneklerinin aktif olmasını sağlamaktadır. Projemizi çalıştırdığımız zaman formun sağ üst köşesinde bulunan ekranı kapla butonunun aktif olmasını istemiyorsak, Özellikler panelinde bulunan maximizebox özelliğini false olarak değiştirmeliyiz. Benzer şekilde minimizebox özelliğini false olarak ayarlarsak formun sağ üst köşesinde bulunan simge durumuna küçült butonu pasif olarak karşımıza gelecektir.

WindowState özelliği için üç farklı durum söz konusudur. Bu özelliğin sağ tarafında bulunan aşağı doğru ok tuşuna tıklandığında şekil 3.22.'de olduğu gibi seçenekler listelenmektedir. Bu seçeneklerden ilki olan *Normal* seçeneği varsayılan olarak atanan değerdir. Bunu seçtiğimiz zaman formumuz ayarlanan boyutlarda ekran üzerinde görünecektir. *Minimized* seçeneği ise form açıldığı zaman ilk olarak formun görev çubuğuna yerleşerek başlamasını ifade etmektedir. Görev çubuğundaki proje simgesine tıklayarak projenin başlangıç formunu ekrana getirebiliriz. Son seçenek olan *Maximized* seçildiğinde ise form, ekranı tam kaplayacak şekilde açılmaktadır.



Şekil 3.22. Windowstate Özelliği

Opacity Özelliği

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



Bireysel Etkinlik

- Şekil 3.23.'de bir formun iki farklı opacity değerine göre ekran görüntüsü paylaşılmıştır. Benzer şekilde word sayfasına "Bilgisayarı Çok Seviyorum..." yazarak formunuzu word üzerine taşıyın. Opacity değeri üzerinde değişiklikler yaparak yazının okunabilirliğini gözden geçirin.



Şekil 3.23. %40 ve %95 Opacity Değeri ile Ayarlanmış İki Form



Proje simgesinin çalışma esnasında görev çubuğu üzerinde bulunup bulunmayacağı ShowInTaskbar özelliği ile ayarlanabilmektedir.

ShowInTaskbar Özelliği

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

AcceptButton ve CancelButton Özellikleri

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

Onay işlemleri için genellikle klavye üzerinde bulunan enter tuşu kullanılmaktadır. Acceptbutton özelliği ile formumuz seçili iken form üzerinde bulunan butonlar görüntülenmektedir. Bu butonlardan hangisini onay işlemi için kullanmak istiyorsak onu seçeriz. Bu işlem yapıldıktan sonra seçilen butonumuz

enter tuşuna basıldıkça görevini icra edecektir. Acceptbutton için yaptığımız işlemlerin aynısını ilgili forma ait Cancelbutton için yaparak iptal işleminde kullanacağımız butonla ESC tuşunu eşleştirmiş oluruz. Aşağıdaki örneği inceleyerek daha iyi kavrayabiliriz.



Onay işlemleri için genellikle klavye üzerinde bulunan enter tuşu kullanılmaktadır.



Örnek

- Formumuz üzerine bir adet textbox ve iki adet buton yerleştirerek tasarım alanında ve kod editöründe yapmamız gerekenleri aşağıdaki adımları izleyerek gerçekleştirelim, enter ve ESC tuşları için ekran görüntüsünü inceleyelim.

- Formumuz üzerine ilk yerleştirdiğimiz butonun text özelliğini özellikler panelinden “onay” olarak değiştirelim.
- İkinci butonun text özelliğini “iptal” olarak değiştirelim.
- Onay butonuna çift tıklayarak kod editörüne aşağıdaki kod satırını yazalım.

```
MessageBox.Show("adınızı ESC tuşu ile iptal ettiniz");
```

- İptal butonuna çift tıklayarak aşağıdaki kod satırını yazalım.

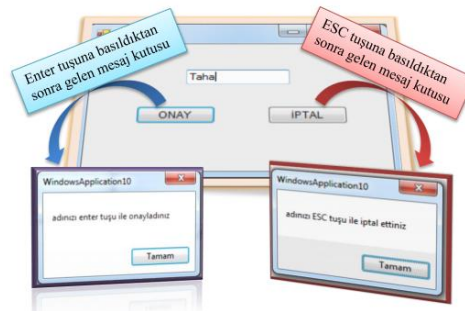
```
MessageBox.Show("adınızı enter tuşu ile onayladınız");
```

- Tasarım kısmında formumuzu seçtikten sonra özellikler panelinden sırasıyla acceptbutton özelliği için buton1’i cancelbutton özelliği için ise buton2’yi seçelim. Bu işlem Şekil 3.24.’de gösterilmiştir.



Şekil 3.24. Acceptbutton ve Cancelbuton Özelliklerini Ayarlama

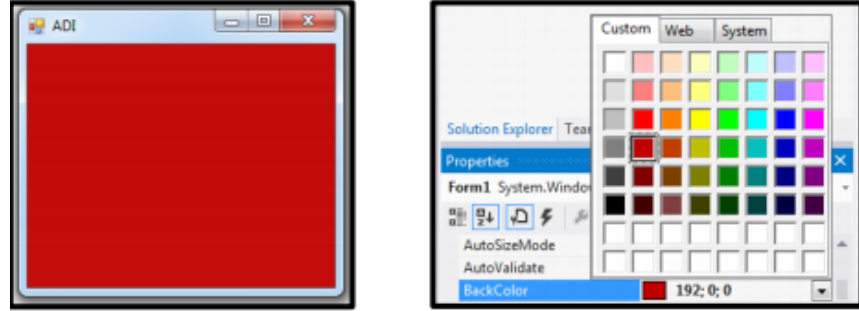
- Başlat butonuna basarak projemizi çalıştıralım. Bu işlemin ardından ENTER ve ESC tuşlarına basıldığında açılacak pencereler Şekil 3.25’de gösterilmiştir.



Şekil 3.25. Enter ve ESC Tuşlarının Kullanımı

BackColor Özelliği

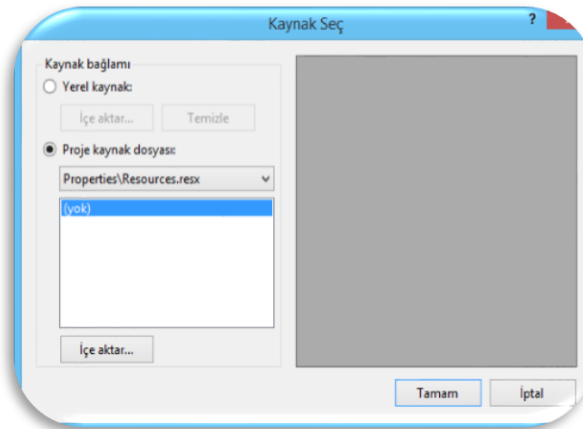
Formun arka plan rengini değiştirme gereği duyulduğunda kullanılan bir özelliktir. Bu özelliğin yanındaki ok tuşuna basıldığında *Custom*, *Web* ve *System* olmak üzere üç sekme halinde renk paleti sunulmaktadır. Şekil 3.26.'da custom sekmesinden bordo renginin seçilmesi ve formun çalışma esnasındaki görüntüsü verilmiştir.



Şekil 3.26. Özellikler Panelinden Bordo Renginin Seçilmesi ve Forma Uygulanması

BackgroundImage Özelliği

Bu özellik sayesinde hazırladığımız formlar istediğimiz görsel öğelerle özelleştirilebilmektedir. Böylelikle formun arka planına, var olan bir resim dosyası yükleyebiliriz. Özellikler panelinde bulunan BackgroundImage özelliğinin yanındaki “...” butonuna tıkladığımızda karşımıza Şekil 3.27’de olduğu gibi *Kaynak Seç* penceresi gelmektedir. Bu pencereden *İçe Aktar...* butonuna tıklayarak eklemek istediğimiz resmin bilgisayarımızdaki yolunu gösterir, ardından *Tamam* butonuna tıklarız.



Şekil 3.27. Kaynak Seç Penceresi

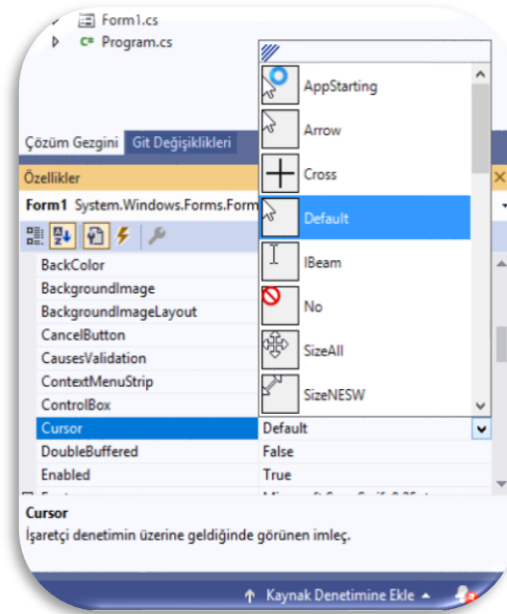
Burada dikkat etmemiz gereken bir husus bulunmaktadır. Resim dosyasını formumuza ekledikten sonra bu resmin form üzerinde nasıl konumlanacağını ayarlamamız gerekmektedir. Bu işlemi yapabilmek için *BackgroundImageLayout* özelliğini kullanabiliriz. Bu özellik içinde dört farklı hizalama seçeneği bulunmaktadır. Bunlar: *Tile*, *Center*, *Stretch* ve *Zoom* seçenekleridir. Projemize en uygun olan seçeneği kullanarak projemizi çalıştırabiliriz.



Formun arkaplan rengini değiştirmek için BackColor özelliği kullanılmalıdır.

Cursor Özelliği

Fare işareti, yapacağı göreve veya bulunduğu duruma göre farklı şekillerde olabilmektedir. Projemizin çalışması esnasında fare ile form üzerinde gezinirken farenin görünmesini istediğimiz şekli belirlemeye yarayan özellik cursor'dür. Bu özellik kullanılarak farenin alacağı şekil ile kullanıcıya formun durumu veya form ile yapılabilecek işlemler hakkında öngörü sağlanabilir. Özellikler panelinde bulunan Cursor özelliğinin ekran görüntüsü Şekil 3.28.'de gösterilmektedir.



Şekil 3.28. Cursor Özelliği



Özet

- C# programlama dili ile Windows tabanlı bir uygulama gerçekleştirmek için Visual Studio 2019 programını başlattıktan sonra sırasıyla Yeni Bir Proje Oluştur seçeneğini işaretleyerek karşımıza gelen ekrandan Windows Forms Uygulaması (.Net Framework)'ü seçmemiz gerekmektedir.
- Windows formlar genel olarak iki ana başlık altında incelenmektedir. Bunlar SDI (Single Document Interface) formlar ve MDI (Multiple Document Interface) formlardır.
- SDI formlar başka bir forma ihtiyaç duymaksızın tek başına çalışabilen bir yapıdadırlar.
- MDI formlar bir ana pencere altında bu ana pencereye bağlı olarak bir veya birden fazla çocuk pencere çalıştırabilen yapılardır.
- Özellikler paneli ile formumuza ait özellikleri görebilme ve değiştirebilme imkânına sahibiz.
- Text Özelliği formun başlık çubuğunda görüntülenen isminin ayarlandığı özellik seçeneğidir. Ancak burada dikkat etmemiz gereken nokta bu isim sadece projemizin tasarım kısmındaki isimdir. Text özelliği haricinde bir değişiklik yapılmadıysa kod editöründe çalışırken formumuzun ismi programın varsayılan olarak atadığı (Form1, Form2 vb.) isimdir.
- ControlBox özelliği program çalışırken anlaşılabilir bir yapıdadır. Mantıksal olarak değer alabilen iki seçenek söz konusudur. Bunlardan biri True (doğru veya 1) diğeri ise False (yanlış veya 0) değeridir.
- Enabled özelliği Controlbox özelliğinde olduğu gibi True ve False şeklinde iki farklı değer alabilen mantıksal bir büyüklüktür. Enabled özelliğinin True olması durumunda form aktif konumdadır ve formun üzerindeki nesneler ile işlem yapılması mümkündür. Formun enabled özelliğini false yapacak olursak bu defa form pasif konuma geçecektir. Bununla birlikte forma yerleştirdiğimiz nesneler de aynı şekilde pasif konumda olacaktır.
- Font özelliği ile form üzerindeki nesnelere ait text özellikleri, form üzerindeki yazı stilleri, yazı font özellikleri ve yazı boyutu istenilen ölçüde ayarlanmaktadır. Böylelikle form ve form üzerindeki nesneler belirlenen parametreler doğrultusunda görüntülenmektedir. Bu değişiklikleri gerçekleştirmek için font özelliğinin sağ tarafında bulunan ... butonuna tıklamak gerekmektedir.
- AutoScroll özelliği form üzerinde bulunan nesnelerin çok fazla olması durumunda veya kullanıcının form alanını tutamaçlar yardımıyla daralttığı durumda yatay ve dikey kaydırma çubuklarının aktif veya pasif olmasını sağlamaktadır.
- Form üzerindeki nesneler hizalanırken, tasarım yapıldığı süre içinde formun kilitli olması istenebilir. Böyle bir durumda locked özelliğine ait değer true olarak ayarlanmalıdır. Bu durumda formun konumu ve boyutu programcı tarafından değiştirilemez.
- StartPosition özelliği ile projeye ait başlangıç formunun ilk açılıştaki ekran üzerinde konumlanacağı alana yönelik ayarlamalar yapılmaktadır. Bu işlem startPosition özelliğinin 5 farklı seçenek sunmasıyla gerçekleştirilmektedir. Bu seçenekler sırasıyla: Manual, CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds ve CenterParent seçenekleridir.
- Projelerimizde hazırlanan formların çalışma esnasında tam ekranı kaplaması, simge durumunda küçülmesi veya görev çubuğunda görünmesiyle ilgili özellikler ayarlanabilmektedir. Bu durumda özellikler panelinde bulunan MaximizeBox, MinimizeBox ve WindowState özelliklerini kullanmamız gerekmektedir.



Özet (devamı)

- Opacity özelliği 0-100 arasında değer alabilen bir yapıdır. Aksi değerler girildiği takdirde hata mesajı vermektedir. İşlerliğinden bahsedecek olursak, formumuzun saydamlığını ayarlamamıza imkân sağlamaktadır. En küçük değer olan sıfır değerini yazdığımızda formun arkasında ne olduğunu tam olarak görebiliriz ancak forma ait işlemlerimizi gerçekleştirme noktasında güçlük çekebiliriz. Yüz değerini yazdığımızda ise formun arkası görülmemektedir.
- ShowInTaskbar özelliği projemizin simgesinin çalışma esnasında görev çubuğu üzerinde bulunup bulunmayacağını ayarlamamıza imkân sağlar. True veya False olmak üzere iki farklı değer alabilir. ShowInTaskbar özelliği True olarak ayarlandığında proje simgesi görev çubuğu üzerinde görüntülenmektedir. Aksi takdirde proje simgesi görev çubuğunda görünmeyecektir.
- AcceptButton ve CancelButton Özellikleri: Projelerimizde bazen ilgili kutucukları klavye ile doldurup onay işlemi vermemiz gerekebilir. Durum tersi şeklinde ilerleyerek iptal etmemiz de söz konusu olabilmektedir. Bu gereklilikler karşısında onay ve iptal işlemleri için formumuza butonlar yerleştiririz. Burada kullanacağımız özellik bu butonları fare yardımı kullanmadan işlevsel yapmamızı yani klavyeden ayrılmadan onaylamamızı veya iptal etmemizi sağlayacaktır.
- BackColor özelliği, formun arka plan rengini değiştirme gereği duyulduğunda kullanılan bir özelliktir. Bu özelliğin yanındaki oka tıkladığımızda Custom, Web ve System olmak üzere üç sekme altında bize renk alternatifi sunmaktadır.
- Cursor Özelliği: Fare işareti, yapacağı göreve veya bulunduğu duruma göre farklı şekillerde olabilmektedir. Projemizin çalışması esnasında fare ile form üzerinde gezinirken farenin görünmesini istediğimiz şekli belirlemeye yarayan özellik cursor'dür.

DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi kod editörüne geçiş yapabilmek için kullanabileceğimiz fonksiyon tuşudur?
 - a) F5
 - b) F6
 - c) F7
 - d) F8
 - e) F9
2. Aşağıdaki pencerelerden hangisi kullanılarak projeye yeni bir form eklenebilir?
 - a) Yazı tipi penceresi
 - b) Yeni öge ekle penceresi
 - c) Yeni proje oluştur penceresi
 - d) Çözüm gezgini
 - e) Özellikler paneli
3. Aşağıdakilerden hangisi formlar arası geçiş işlemi gerçekleştirilirken açılan pencere kapatılmadan diğerine geçiş izni vermeyen metottur?
 - a) Visible();
 - b) Show();
 - c) Hide();
 - d) Showdialog();
 - e) Close();
4. Aşağıdaki özelliklerden hangisi, uygulamada var olan SDI formun MDI form olarak değiştirilmesine imkan tanımaktadır?
 - a) Text
 - b) Name
 - c) StartPosition
 - d) Font
 - e) IsMdiContainer
5. Aşağıdakilerden hangisi özellikler panelinin gösterilmesini sağlayan fonksiyon tuşudur?
 - a) F1
 - b) F2
 - c) F3
 - d) F4
 - e) F5

6. Aşağıdaki özelliklerden hangisi formun kod editöründe kullanılan ismini ifade etmek için kullanılır?
 - a) Name
 - b) Text
 - c) Font
 - d) Opacity
 - e) Locked

7. Aşağıdaki seçeneklerden hangisi StartPosition özelliğinin ayarları arasında bulunmaz?
 - a) Manual
 - b) Centerscreen
 - c) Windowsdefaultlocation
 - d) Windowsdefaultbounds
 - e) Windowstate

8. Aşağıdakilerden hangisi opacity özelliğinin alabileceği maximum ve minimum değerleri göstermektedir?
 - a) 100 – 0
 - b) 95 – 40
 - c) 200 – 0
 - d) 100 – 10
 - e) 80 – 10

9. Projede var olan bir formun arka plan rengini değiştirmek için kullanacağımız backcolor özelliği hangi sekmelerden oluşmaktadır?
 - a) Custom – Net – System
 - b) Color – Web
 - c) Web – System
 - d) Custom – System
 - e) Custom – Web – System

10. Aşağıdakilerden hangisi formumuz üzerinde bulunan butonlardan herhangi birine enter tuşu ile işlem yapabilme özelliğini aktarmak için kullanılır?
 - a) Acceptbutton
 - b) Cancelbutton
 - c) Showintaskbar
 - d) Backcolor
 - e) Font

Cevap Anahtarı

1.c, 2.b, 3.d, 4.e, 5.d, 6.a, 7.e, 8.a, 9.e, 10.a

YARARLANILAN KAYNAKLAR

- Aktaş, V. (2010). *Visual studio 2010 ile her yönüyle C# 4.0* (2. Baskı). İstanbul: Kodlab Yayın.
- Aktaş, V. (2020). *Visual studio 2019* (1.Baskı). İstanbul: 01 Yayınları.
- Atasever, V. (2017). *C# 7 Uygulamalarla C# programlama dilini keşfedin* (2. Baskı). Kocaeli: Level Kitap.
- Demirli, N. & İnan, Y. (2005). *Visual C# .Net* (3. Baskı). Ankara: Palme Yayıncılık.
- Sharp, J. (2011). *Adım adım Microsoft visual C# 2010*. (Çev. T. Buldu). Ankara: Arkadaş Yayınevi.
- Uzunköprü, S. (2017). *Projeler İle C# 7.0 ve SQL Server 2016* (7. Baskı). İstanbul: Kodlab Yayın.
- Yanık, M. (2008). *Visual studio 2008 ile Microsoft visual C# 3.0 for .Net framework 3.5* (1. Baskı). Ankara: Seçkin Yayıncılık.
- Yücedağ, M. (2020). *C# Eğitim kitabı* (3. Baskı). İstanbul: Dikeyeksen.