

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

GELİŞMİŞ İNTERNET UYGULAMALARI İLE TARAYICI İŞLEMLERİ VE ARAYÜZ OLUŞTURMA

Ankara, 2013

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-2	3
1. TARAYICI İŞLEMLERİ	3
1.1. Tarayıcı Yazılımının Rolü	3
1.2. Sayfa Yükleme Olayları	4
1.2.1. Sayfa Yükleme	4
1.2.2. Yükleme Sayfasının Çalışma Prensibi	4
1.2.3. Yükleme Sayfası Hazırlama	6
1.3. Sayfa Hata Olayları ve Hata Ayıklama	11
1.3.1. Hata Yönetimi	11
1.3.2. Hata Türleri	11
1.3.3. Hata Ayıklama	13
UYGULAMA FAALİYETİ	15
ÖLÇME VE DEĞERLENDİRME	17
ÖĞRENME FAALİYETİ-2	18
2. KULLANICI ARAYÜZÜ OLUŞTURMA	18
2.1. Olaylar	18
2.2. Metodlar	20
2.3. Sürükle-Bırak İşlemi	20
UYGULAMA FAALİYETİ	22
ÖLÇME VE DEĞERLENDİRME	24
MODÜL DEĞERLENDİRME	25
CEVAP ANAHTARLARI	26
KAYNAKÇA	27

AÇIKLAMALAR

ALAN	Bilişim Teknolojileri
DAL / MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Gelişmiş İnternet Uygulamaları ile Tarayıcı İşlemleri ve Arayüz Oluşturma
MODÜLÜN TANIMI	Gelişmiş internet uygulamaları ile tarayıcı işlemlerini gerçekleştirmek ve kullanıcı arayüzü oluşturmak için gerekli bilgi, beceri ve tutumların kazandırıldığı bir öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	
YETERLİK	Tarayıcı işlemlerini yapabilmek ve kullanıcı arayüzü hazırlayabilmek.
MODÜLÜN AMACI	Genel Amaç Bu modül ile; gerekli ortam sağlandığında yazılım geliştirebileceksiniz. Amaçlar 1. Tarayıcı işlemlerini gerçekleştirerek sayfa olaylarını düzenleyebileceksiniz. 2. Kullanıcı arayüzünü programlayabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Bilgisayar laboratuvarı Donanım: Animasyon düzenleme yazılımını çalıştırabilecek yeterlikte bilgisayar
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen, modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Her geçen gün önemini artıran ve yaygınlaşan XAML tabanlı zengin internet uygulamalarıyla çalışırken en sık karşılaştığınız konulardan ikisi tarayıcı işlemlerini yönetebilmek ve kullanıcı arayüzleri hazırlamaktır.

Tarayıcı üzerinden çalışan tüm uygulamaların verimliliği ve işlevselliği tarayıcı kaynaklarının etkin kullanımına bağlıdır. Modülün içerisindeki konuları tamamladığınızda uygulamanızın sağlıklı ve sorunsuz biçimde çalışmasını sağlayacak adımları atabileceksiniz.

Kullanıcılar açısından bir uygulamanın en önemli kısmı şüphesiz arayüzüdür. Bu modülden faydalanarak arayüz elemanlarının rolleri, kullanım alanları ve birbirleriyle ilişkilendirilmeleri konularında bilgi ve beceri sahibi olabileceksiniz.

Gelişmiş internet uygulamaları oluşturabilmek için kullanılacak çok sayıda farklı ürün mevcuttur. Bu modülde süreç XAML teknolojisi üzerinden ele alınacaktır.

İnternet üzerinde performansı yüksek ve görselliği yüksek uygulamalar gerçekleştirmek için iyi bir kaynak olabilecek bu modülü başarıyla tamamlamanız dileğiyle.

ÖĞRENME FAALİYETİ-1

AMAÇ

Tarayıcı işlemlerini gerçekleştirerek sayfa olaylarını düzenleyebileceksiniz.

ARAŞTIRMA

- Yaygın internet tarayıcılarını inceleyerek benzerlik ve farklılıklarını tespit ediniz.
- Bir internet tarayıcısının yerine getirmesi gereken görevleri tartışınız.
- Arkadaşlarınızın internette gezinirken hata mesajlarıyla karşılaşp karşılaşmadıklarını, karşılaştımlarsa mesajların ne tür içeriği olduğunu araştırınız.
- Yazılımların neden hata mesajları hazırladığını tartışınız.

1. TARAYICI İŞLEMLERİ

1.1. Tarayıcı Yazılımının Rolü

İnternet sayfalarını görüntülemeyi sağlayan tarayıcı yazılımları, sade görünümlerinin arka planında oldukça karmaşık ve önemli işlemler yerine getirir. Gelişmiş *İnternet* uygulamaları tarayıcı üzerinde çalışan yazılımlardır. Bu nedenle tarayıcı yazılımıyla devamlı olarak etkileşim hâlinindedir. Ayrıca tarayıcı yazılımı, *İnternet* uygulamalarının gerçekleştiremeyeceği bazı işlemleri yerine getirebildiğinden, yapabileceklerimizin sınırlarını genişletmek konusunda da iş birliği yapacağımız bir sistemdir.



Resim 1.1.: XAML uygulamalarının tarayıcıda çalışma prensibi

XAML tabanlı gelişmiş internet uygulamalarında tarayıcı yazılımı kullanıcının doğrudan etkileşime geçtiği kısımdır. Uygulamanın tarayıcı tarafından kullanıcının bilgisayarına yüklenmesi ve hatasız bir şekilde çalıştırılabilmesi son derece önemlidir. Muhtemel hata durumlarında ise kullanıcının doğru ve yeterli bir şekilde bilgilendirilmesi ve yönlendirilmesi gereklidir.

1.2. Sayfa Yükleme Olayları

1.2.1. Sayfa Yükleme



Resim 1.2: Sayfa yükleme

Bir internet sitesine bağlandığımızda tarayıcı yazılımı, adres satırında belirlediğiniz web sunucusuna bağlanarak sayfanın görüntülenmesi için gerekli tüm içeriği sunucudan alır ve bu içeriği öngörülen formatta görüntüler. Bu süreci sayfanın yüklenmesi olarak ifade edebiliriz.

Resim 1.1.'de gösterildiği üzere XAML tabanlı internet uygulamanız “.XAP” uzantılı bir dosyadır. Dolayısıyla XAML uygulaması bulunan bir sayfanın yüklenmesi, ilgili XAP dosyasının indirilip görüntülenmesi demektir. Gelişmiş internet uygulamaları zengin içerik sağlamaktadır. Bu nedenle XAP dosyası büyük bir veri ihtiva ediyor olabilir. Özellikle video, resim, animasyon gibi çoklu ortam öğeleri içeren gelişmiş internet uygulamalarının yüklenmesi esnasında kullanıcılar uzun süre beklemek zorunda kalabilir.



Resim 1.3: Sayfa yükleme animasyonu

Özellikle *Internet* ortamında hızın son derece önemli olduğu günümüzde, kullanıcıların yükleme işlemi sırasında uygulamaya odaklanmalarını sağlamak için “Yükleme ekranları (SplashScreen)” hazırlamak iyi bir yöntemdir. Resim 1.3'te sayfa yükleme esnasında görüntülenen varsayılan yükleme animasyonu görülmektedir.

1.2.2. Yükleme Sayfasının Çalışma Prensibi

Yükleme ekranları çok çeşitli olabilir. Örnek olarak *Internet* uygulamalarında en yaygın olarak gördüğümüz, yükleme işlemindeki ilerlemeyi kullanıcıya yüzdelik değer

olarak bildiren basit geiř sayfaları tasarlanabilir. Ykleme ekranını uygulamanızla baėdařtırmayı ğrendikten sonra hazırlayabileceėiniz farklı ykleme ekranları yalnızca hayal gcnzle sınırlıdır.

Kendi ykleme ekranımızı hazırlarken  zelliėi kullanmak gerekecek:

splashscreen	Hazırladıėımız ykleme sayfasının yolunu, hmtl dosyasına bildirirken kullanacaėımız parametredir.
OnSourceDownloadProgressChanged	Ykleme iřleminde ilerleme gerekleřtiėinde tetiklenerek alıřan JavaScript olayıdır.
OnSourceDownloadComplete	Ykleme iřlemi tamamlandıėında tetiklenen JavaScript olayıdır.

Tablo 1.1: Ykleme parametre ve olayları

Hazırlanacak ykleme animasyonu sayfası projenin web tarafında yer almalıdır. Bylece kullanıcının eriřim isteėine ncelikle bu sayfa yklenerek cevap verilecektir. Hazırladıėımız sayfa ierisinde, Tablo 1.1’de gsterilen JavaScript olayları sayesinde asıl uygulamamızın tm verilerinin aktarılıp aktarılmadıėı kontrol edilecektir. İřlem tamamlandıėında, asıl uygulamamız tarayıcı yazılımı tarafından grntlenecektir.



Resim 1.4: Ykleme ekranı alıřma ilkesi

1.2.3. Yükleme Sayfası Hazırlama

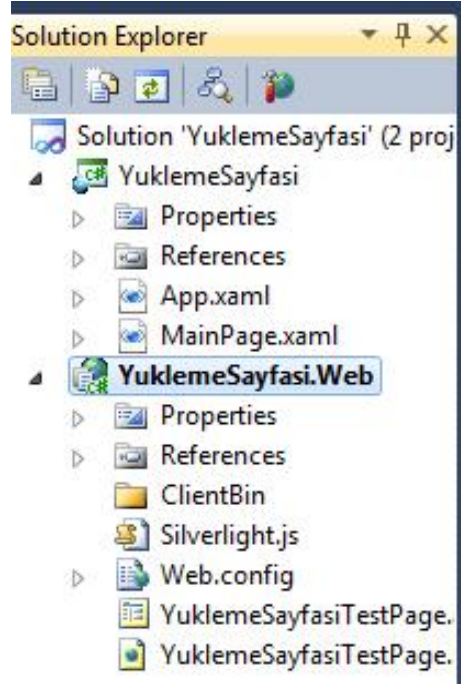


Resim: 1.5: Yükleme sayfası hazırlama

İlk olarak Visual Studio 2010 içerisinde yeni bir Silverlight uygulaması oluşturalım. Uygulamamızın ismini “**YuklemeSayfasi**” olarak belirleyelim.

Web projemiz için çeşitli ayarlamalar yapabileceğimiz bir iletişim kutusu görüntülenecektir.

Resim 1.5’te gösterildiği üzere varsayılan ayarlarda herhangi bir değişiklik yapmadan devam edelim.



Resim 1.6: Yükleme sayfası hazırlama

Ekranın sağ kısmında yer alan “Solution Explorer” bölümünde projemiz içerisinde yer alan tüm dosyalar görüntülenmektedir.

Bu kısımdan **YuklemeSayfasi.Web** üzerinde sağ tıklayalım. Açılan menüde “Properties” seçeneğine tıklayalım. Böylece web projemiz için gerekli ayarlamaları yapabileceğimiz arayüz görüntülenecektir.

Sol kısımdaki “Web” sekmesini seçtiğimizde, projemizin “SpecificPage” özelliğinin “YuklemeSayfasiTestPage.aspx” olarak belirlenmiş olduğunu görebiliriz.

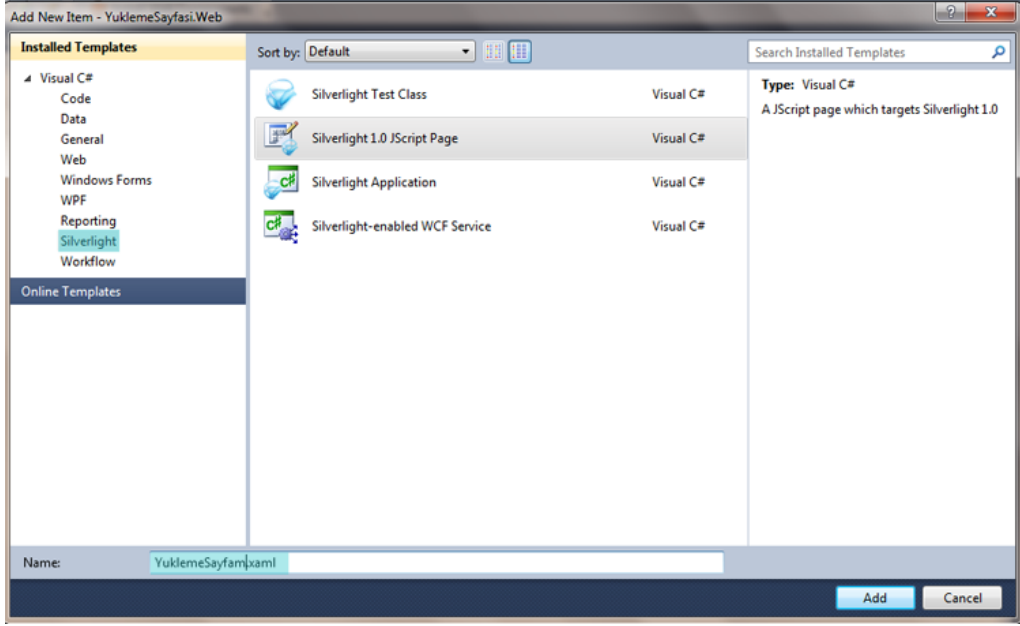
Bu kısımdan SpecificPage özelliğini “YuklemeSayfasiTestPage.html” olarak değiştirelim.



Resim 1.7: Yükleme sayfası hazırlama

Böylece, varsayılan olarak **aspx** türünde tanımlanan web sayfasını devre dışı bırakmış oluyoruz. Projemizi HTML üzerinden çalıştırmamız yeterli olacaktır. Web projemize yükleme ekranı olarak kullanacağımız sayfayı ekleme aşamasına gelmiş bulunuyoruz. Tekrar **YuklemeSayfasi.Web** üzerinde sağ tıklayarak sırasıyla “Add”, “New Item” seçeneklerini seçelim.

“Add New Item” penceresinden sol tarafta bulunan Silverlight sekmesini işaretledikten sonra “**Silverlight1.0 JScript Page**” dosyasını seçelim. Sayfamızı “**YuklemeSayfam.xaml**” olarak isimlendirelim ve “Add” butonuna tıklayalım.



Resim 1.8: Web projesine yükleme sayfasının eklenmesi

Sağ kısımdaki Solution Explorer penceresinde “**YuklemeSayfam.xaml**” ve “**YuklemeSayfam.js**” isimlerine sahip iki yeni dosyanın daha yer aldığını görebilirsiniz.

Eğer otomatik olarak düzenlemek üzere görüntülenmemiş ise Solution Explorer penceresinden seçerek **YuklemeSayfam.xaml** dosyamızı açalım. Hâlihazırda bir kod bloğunun sayfamızda var olduğunu görebiliriz. Sayfadaki tüm kodları silelim. Bu noktadan sonra tamamen istediğinize göre bir yükleme sayfası için gerekli XAML kodlarını sayfanıza eklemelisiniz.

Bizim örneğimizde, yükleme işlemindeki ilerlemeyi yüzdelerik değerler olarak göstermek üzere iki adet TextBlock, bir elips şekli ve yükleme ilerlemesini temsil edecek biçimde eni artacak bir dikdörtgen şeklinden oluşan bir sayfa tasarımı yapacağız.

YuklemeSayfam.xaml dosyası için aşağıdaki kodları kullanalım:

```
<Canvas
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="400">
<Ellipse Canvas.Left="132" Canvas.Top="100" Height="84"
Stroke="Black" Width="84" Fill="#FF4BCB12" StrokeThickness="5"/>
<TextBlock FontSize="20" Canvas.Left="174" Canvas.Top="126"
Height="23" Text="0" x:Name="textBlock1" FontWeight="Bold"
Foreground="#FFF1F8EE" />
<TextBlock Canvas.Left="145" Canvas.Top="126" FontSize="20"
FontWeight="Bold" Height="23" Text="%" Width="25"
Foreground="#FFF3F8F1" />
<Rectangle Canvas.Left="123" Canvas.Top="195" Height="11"
Stroke="Black" Width="100" Fill="Black" />
<Rectangle x:Name="cubuk" Canvas.Left="122" Canvas.Top="195"
Height="11" Stroke="Black" Width="7" Fill="Red" />
</Canvas>
```

Kodlarımızı sayfamıza ekledikten sonra artık uygulamamız yüklenene kadar kullanıcılara gösterilecek yükleme sayfamız hazır. Bir sonraki aşamada hazırladığımız sayfayı web projenizin html sayfası içerisinden çağırmanız gerekir.

Solution Explorer penceresinde düzenlemek üzere “YuklemeSayfasiTestPage.html” dosyasına çift tıklayalım. Kodları incelediğimizde uygulamamızın <object> etiketleri arasında tanımlandığını görebiliriz. Bu kısımda var olan “param” etiketlerinin altına aşağıdaki satırları ekleyelim:

```
<param name="splashscreensource" value="YuklemeSayfam.xaml"/>
<param name="onSourceDownloadProgressChanged"
value="onSourceDownloadProgressChanged"/>
```

İlk satırdaki kod sayesinde daha önce Tablo 1.1’de belirtmiş olduğumuz **splashscreensource** özelliğini belirlemiş oluyoruz. Varsayılan yükleme sayfası yerine bizim hazırladığımız **YuklemeSayfam.xaml** sayfasının çağrılmasını sağlıyoruz.

İkinci satırda ise yükleme işlemini denetleyecek bir JavaScript olayı çağırılmaktadır. Bu JavaScript olayını tanımlamak için Solution Explorer penceresinden YuklemeSayfasiTestPage.html isimli dosyayı düzenlemek üzere açalım. **Bu dosyadaki**

```
<script type="text/javascript">  
</script>
```

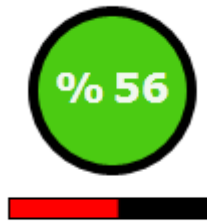
blokları arasında bir takım JavaScript kodları mevcuttur. Bu JavaScript kodlarının altına aşağıdaki metod kodlarını yazınız.

```
function onSourceDownloadProgressChanged(sender, eventArgs)  
{  
    sender.findName("textBlock1").Text =  
        Math.round((eventArgs.progress * 100), 0).toString();  
    sender.findName("cubuk").Width = eventArgs.progress*100;  
}
```

Tanımladığımız bu `onSourceDownloadProgressChanged` fonksiyonu ile YuklemeSayfam.xaml sayfamız içerisindeki TextBlock nesnemizde görüntülenecek değeri ve ilerlemeyi temsil edecek dikdörtgen nesnesinin genişliğini güncelleyeceğiz.

EventArgs.Progress özelliği içinde yükleme ilerlemesi değeri bulunur. Bu değer 0 ile 1 arasında bir değerdir. % cinsinden ifade etmek için Progress değeri 100 ile çarpılmıştır. Hem textBlock1 üzerinde gösterilmiş hem de alttaki yatay çubuk şeklindeki dikdörtgenin genişlik özelliği bu değere göre artmıştır.

Böylece kendi yükleme sayfamızı hazırlamak için gerekli adımları tamamlamış oluyoruz. **File** menüsünden **SaveAll** seçeneğine tıklayarak tüm dosyalarımızı kaydedelim. Projenizi derleyerek (F5 Kısayolunu kullanabilirsiniz) yükleme sayfanızın işleyişini test edebilirsiniz.



Resim 1.9: Yükleme animasyonunun görüntüsü

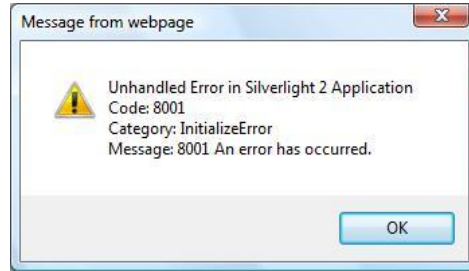
Yükleme sayfalarını oluşturma amacımızın büyük miktarda veri aktarılırken kullanıcıyı bilgilendirmek olduğunu hatırlayalım. Bu yüzden hazırladığımız yükleme sayfasını denerken resim, ses, video gibi büyük miktarda verilerden oluşan bileşenler içeren bir gelişmiş *Internet* uygulamasını kullanmalısınız.

1.3. Sayfa Hata Olayları ve Hata Ayıklama

1.3.1. Hata Yönetimi

Yazılım geliştirme süreci de çok sayıda değişkene bağlı ve oldukça kompleks işlemler gerektirir. Bu nedenle her yazılımın bir hata yönetimi boyutu vardır. Bu durum bizim konumuz olan gelişmiş *İnternet* uygulamaları için de geçerlidir.

Bir kullanıcı olarak kullandığımız uygulamanın bir hataya sebep olması durumunda sizi haberdar etmesini, sorunun nereden kaynaklandığını açıklamasını ve mümkünse bu sorunu nasıl çözebileceğinize dair sizi yönlendirmesini beklersiniz. Hata yönetimi, uygulamalarımız için bu işlevleri yerine getiren bir destek mekanizması kurmak anlamına gelir.



Resim 1.10: Örnek bir Silverlight hata mesajı

1.3.2. Hata Türleri

Hatalar farklı durumlardan kaynaklanabilir. Programcının tüm olası ihtimallere karşılık uygun bir tepki hazırlamamış olmasından, yani istisnalar bulunmasından kaynaklanan hatalara “İstisna (Exception)” adı verilir. “Exception” terimi artık her tür hata durumunu ifade eden genel bir ifade hâline gelmiştir.

Hataları gerçekleştikleri zaman göre şöyle sınıflayabiliriz:

Derleme Hataları (Compile Errors)	Kod yazımından kaynaklı, uygulamanın çalıştırılmasından önce meydana gelen hatalar
Çalışma Zamanı Hataları (Runtime Errors)	Uygulama çalıştırıldıktan sonra meydana gelen hatalar

Tablo 1.2 HatalarıN gerçekleştikleri zamana göre sınıflanması

Hatalar kaynaklandıkları nedenlere göre de çeşitlilik arz eder. Uygulamanızın niteliğine göre gerçekleşme olasılığı diğerlerinden daha yüksek olan durumlar için kendi hata yöneticinizi düzenlemek isteyebilirsiniz.

Web projemizin HTML sayfasında Silverlight uygulamamıza ait <object> etiketleri içerisinde varsayılan olarak bir hata işleyici fonksiyon tanımlanmıştır.

```
<param name="onError" value="onSilverlightError"/>
```

Aynı HTML sayfa içerisinde bu satırda işaret edilen “**onSilverlightError**” isimli JavaScript fonksiyonu ise şöyle tanımlıdır:

```
<script type="text/javascript">
    function onSilverlightError(sender, args) {
        var appSource = "";
        if (sender != null && sender != 0) {
            appSource = sender.getHost().Source;
        }

        var errorType = args.ErrorType;
        var iErrorCode = args.ErrorCode;

        if (errorType == "ImageError" || errorType == "MediaError")
        {
            return;
        }

        var errMsg = "Unhandled Error in Silverlight Application " +
appSource + "\n" ;

        errMsg += "Code: " + iErrorCode + "    \n";
        errMsg += "Category: " + errorType + "    \n";
        errMsg += "Message: " + args.ErrorMessage + "    \n";

        if (errorType == "ParserError") {
            errMsg += "File: " + args.xamlFile + "    \n";
            errMsg += "Line: " + args.lineNumber + "    \n";
            errMsg += "Position: " + args.charPosition + "    \n";
        }
        else if (errorType == "RuntimeError") {
            if (args.lineNumber != 0) {
                errMsg += "Line: " + args.lineNumber + "    \n";
                errMsg += "Position: " + args.charPosition + "
\n";
            }
            errMsg += "MethodName: " + args.methodName + "    \n";
        }

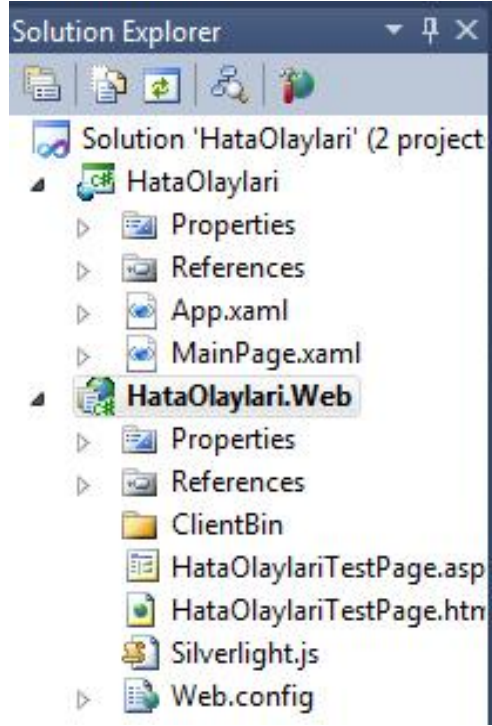
        throw new Error(errMsg);
    }
</script>
```


Çeşitli hata türleri için hatanın nitelik ve kaynağına dair bildirimler üreten bu JavaScript fonksiyonunu kendi ihtiyaçlarınıza göre yeniden düzenleyebilir ya da tamamen yeni bir fonksiyonu baştan yazarak projenize dâhil edebilirsiniz.

1.3.3. Hata Ayıklama

Doğası gereği hatalara oldukça açık olan uygulama geliştirme sürecinde problemleri tespit etme ve düzeltme işlemine “hata ayıklama”(debugging) adı verilir. Bu modülde konu edindiğimiz Silverlight teknolojisi ile uygulama geliştirirken Visual Studio 2010 içerisinde bulunan Hata Ayıklayıcı (Debugger) özelliği size çok yardımcı olacaktır.

Hata ayıklayıcı hem uygulamanızı kodlarken hem de çalışma zamanında denerken problemleri denetleyecek ve size ayrıntılı raporlar sunacaktır. Bu özellik varsayılan olarak aktif değilse aktifleştirmeniz gerekir.

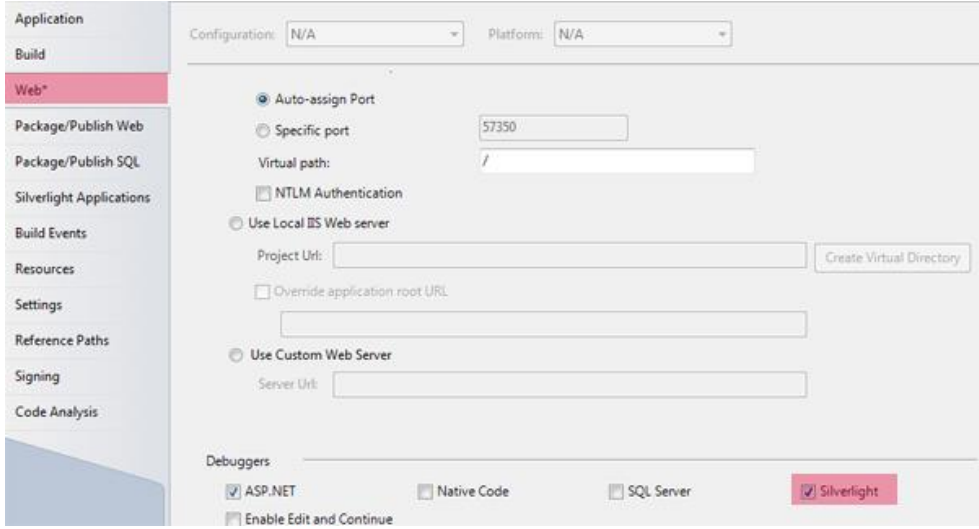


Resim 1.11: Web projesi seçme

İlk olarak Visual Studio 2010 içerisinde, sağ kısımda bulunan Solution Explorer penceresinden Web projemizi seçelim ve sağ tıklayalım.

Açılan menüde en alt kısımda bulunan “Properties” seçeneğine tıklayarak devam edelim.

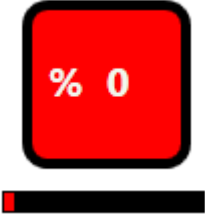
Ekranda görüntülenen “Properties” penceresinde sol kısımdan “**Web**” sekmesini seçelim. Bu sekmede sayfanın en altında yer alan “Debuggers” alanında “Silverlight” seçeneğini işaretleyelim ve projemizi kaydedelim.



Resim 1.12: Kaydetme

Hata ayıklama işlemlerine ilişkin detaylı bilgiyi Gelişmiş İnternet Uygulamalarında Dinamik Programlama modülünde bulabilirsiniz.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Proje editör programını kullanarak yeni bir Silverlight projesi oluşturunuz. 	<ul style="list-style-type: none"> ➤ File→New Project→Silverlight→Silverlight Application
<ul style="list-style-type: none"> ➤ Projenin web tarafına bir xaml dosyası ekleyiniz. 	<ul style="list-style-type: none"> ➤ Projenin web klasörünün üzerinde sağ tuş menüsünü görüntüleyiniz. ➤ Add→New Item→Silverlight→Silverlight 1.0 Javascript Page
<ul style="list-style-type: none"> ➤ Xaml dosyası içerisine yükleme animasyonunu için gerekli xaml kodlarını yazınız. 	<ul style="list-style-type: none"> ➤ Üç adet Rectangle ve iki adet TextBlock kullanarak aşağıdaki gibi bir tasarım yapabilirsiniz (Üzerinde 0 yazan TextBlock ismi "textBlock1" ve alttaki küçük kırmızı dikdörtgenin adı "kutu" olmalıdır.). 
<ul style="list-style-type: none"> ➤ Projenin html uzantılı olan dosyasını başlangıç sayfası yapınız. 	<ul style="list-style-type: none"> ➤ Sayfa üzerinde sağ tuşa tıklayınız. ➤ Set As Start Page komutunu tıklayınız.
<ul style="list-style-type: none"> ➤ Projenin html kodlarını görüntüleyiniz. 	
<ul style="list-style-type: none"> ➤ <object></object> bloğunun içine yan taraftakine benzer parametre tanımlamalarını yapınız (Kendi oluşturduğunuz sayfa adına göre). 	<pre><param name="splashscreen" value="yukleme.xaml"/> <param name="onSourceDownloadProgressChanged" value="onSourceDownloadProgressChanged"/></pre>
<ul style="list-style-type: none"> ➤ JavaScript kod bloğu içindeki kodların altına yan taraftaki fonksiyon tanımlamasını yapınız. 	<pre>function onSourceDownloadProgressChanged(sender, eventArgs) { sender.findName("textBlock1").Text = Math.round((eventArgs.progress * 100), 0).toString(); sender.findName("cubuk").Width = eventArgs.progress*100; }</pre>
<ul style="list-style-type: none"> ➤ Projeyi çalıştırınız. 	<ul style="list-style-type: none"> ➤ F5 tuşunu kullanınız.

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Yükleme sayfalarının işlev ve gerekliliğini kavradınız mı?		
2. HTML-JavaScript-Silverlight ortamları arasındaki ilişkiyi kavradınız mı?		
3. Yükleme sayfasından asıl Silverlight uygulamasına geçişi sağlayabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı yeniden gözden geçiriniz. Kendinizi yeterli buluyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme” bölümüne geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Yükleme sayfası tasarımında kullanılan teknolojilerin işleyiş sırası hangisinde doğru verilmiştir?
A) Silverlight-HTML-JavaScript
B) JavaScript-Silverlight-HTML
C) HTML-Silverlight-JavaScript
D) HTML-JavaScript-Silverlight
2. Bir yükleme sayfasının bulunması gereken yer aşağıdakilerden hangisidir?
A) Silverlight projesi
B) Silverlight Web projesi
C) Debug klasörü
D) Herhangi Bir XAML dosyası
3. Aşağıdakilerden hangisi yükleme işleminin tamamlanıp tamamlanmadığını denetleyen JavaScript olayıdır?
A) onError
B) splashscreenSource
C) onSourceDownloadComplete
D) onLoad
4. Aşağıdakilerden hangisi HTML dosyası içerisinde yükleme sayfası kaynağını belirleyen parametredir?
A) object
B) src
C) splashscreenSource
D) id
5. Silverlight uygulamasının veri aktarımının denetlenmesi gibi sistem kaynaklarıyla ilgili bir görevi yerine getirmek için aracı olarak kullandığı ortam hangisidir?
A) JavaScript
B) HTML
C) XAML
D) XML
6. Kullanıcının uygulamayla ilgili sorunlardan haberdar olması ve çözüme yönlendirilmesi işine ne ad verilir?
A) Hata Düzeltici
B) Hata Yönetimi
C) Hata Ayıklama
D) Hata Yakalama

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Gelişmiş *İnternet* uygulamalarında arayüz hazırlamayı öğreneceksiniz.

ARAŞTIRMA

- Masaüstü uygulamalarında ya da internette gezinirken bilgisayara bir komut vermek ya da bir bilgi girmek için kullandığınız elemanların bir listesini yapınız.
- Bir uygulamanın kullanımında arayüzün etkisini arkadaşlarınıza ve size ait tecrübeleri derleyerek değerlendiriniz.

2. KULLANICI ARAYÜZÜ OLUŞTURMA

XAML tabanlı gelişmiş internet uygulamalarında butonlar, metin kutuları, liste kutuları gibi kontrollerden faydalanarak çeşitli arayüz tasarımları yapabilmekteyiz.

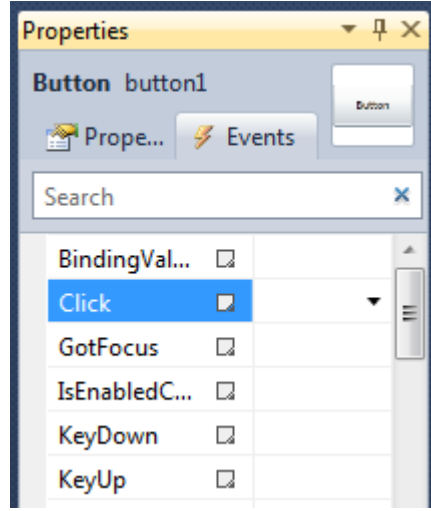


Resim 2.1: Çeşitli arayüz elemanları

2.1. Olaylar

Gelişmiş *İnternet* uygulamaları hazırlarken en çok kullanacağımız programlama kavramlarından birisi “**olay**”dır. Olaylar (Event), belirli durum veya eylemlerin gerçekleşmesi hâlinde yürütülecek işlemleri belirleyen yapılardır. Örneğin bir kontrole tıklanması, bir metin girilmesi, bir nesnenin taşınması

Tasarım editöründe bir kontrole olay eklemek için öncelikle nesne seçilir. Ekranın sağ kısmındaki Properties penceresinde yer alan Events düğmesine tıklayarak o nesneye ilişkin kullanılabilir olaylar listelenir.



Resim 2.2: Buton olayları

Kontrollere ait olayları programlayabilmek için kontrollere isim verilmesi gerekmektedir.

XAML tabanlı uygulamalarda kontrollerin tümü için kullanılan ortak olaylar olduğu gibi yalnızca bazı kontrollere özgü olaylar da mevcuttur.

Events panelinde kullanılmak istenen olayın sağ yanında bulunan alana çift tıklanarak olay oluşturulur. Böylece olayla ilgili kodlamanın yapılacağı C# kod sayfası görüntülenir.

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    |
}
```

Resim 2.3. Click olayına ait C# kod bloğu

Olay için hazırlanan metod içerisine ilgili olay gerçekleştiğinde yürütülmesi istenen kodlar yazılır.

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    button1.Width = button1.Width + 50;
}
```

Resim 2.4: Click olayına ait örnek C# kodu

XAML tabanlı gelişmiş internet uygulamaları içerisinde farklı ihtiyaçlar için çok sayıda olay tanımlanmıştır. Arayüz programlarken en çok kullanılan bazı olaylar ve kısa açıklamaları Tablo 2.1’de verilmiştir.

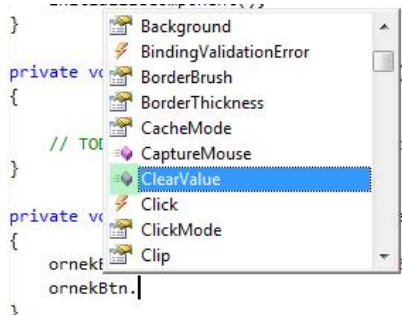
En çok kullanılan olaylar	
Click	Kontrol üzerine tıklanmasıyla tetiklenir.
MouseDown	Farenin sol tuşuna her basıldığında tetiklenir.
MouseUp	Basılı durumdaki farenin sol tuşu her bırakıldığında tetiklenir.
MouseMove	Fare hareket ettikçe tetiklenir.
KeyDown	Klavyeden herhangi bir tuşa basıldığında tetiklenir.
GotFocus	Kontrol seçili hâle getirildiğinde tetiklenir.
MouseEnter	Fare kontrol üzerine geldiğinde tetiklenir.
MouseLeave	Fare kontrol üzerinden ayrılınca tetiklenir.

Tablo 2.1: Silverlight içerisinde en çok kullanılan olaylar ve tanımları

2.2. Metodlar

Gelişmiş *İnternet* uygulamaları tasarlarken kullandığımız her kontrolün yerine getirebildiği kendisine özgü işlemler bulunur. Bu işlemlere “metod” adı verilir.

Tasarım editöründe C# kod sayfasında ilgili kontrole verilen isim yazıldıktan sonra “.”(Nokta) ayracı konulduğunda IntelliSense özelliği o kontrol için mevcut bulunan özellik, olay ve metodları listeler. Bu liste içerisinde Şekil 7.1’de gösterilen simge metodları ifade etmektedir.



Resim 2.5: Metod ekleme

2.3. Sürükle-Bırak İşlemi

Sayfa tasarımında kullandığımız kontrolleri sayfa üzerinde istenilen bölgeye taşıyabilmek için ilgili nesnenin `MouseDown`, `MouseUp` ve `MouseMove` olaylarını kullanmalıyız.

Sürüklenecek nesnenin bir canvas içerisinde bulunması gereklidir. Program koduyla canvas içerisindeki nesnelerin üst konum özelliğini değiştirmek için `Canvas.SetTop()` metodunu kullanmalıyız. Benzer şekilde bir kontrolün üst konum değerini elde etmek için `Canvas.GetTop()` metodu kullanılır.

Kontrollerin sol konum değerlerini almak için `Canvas.GetLeft()` metodu kullanılır. Ayrıca sol konum değerlerini ayarlamak için `Canvas.SetLeft()` metodu kullanılmalıdır.

Örnek: Sayfa üzerinde bulunan bir elips nesnesinin fareyle istenen bölgeye taşınabilmesi için gerekli olay ve metodları kullanalım.

```
bool surukleniyor = false;
Point ilkkonum;

private void ellipse1_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
{
    surukleniyor = true;
    ilkkonum = e.GetPosition(ellipse1);
}
private void ellipse1_MouseMove(object sender, MouseEventArgs e)
{
    Point yenikonum=e.GetPosition(ellipse1);
    if (surukleniyor)
    {
        Canvas.SetTop(ellipse1,Canvas.GetTop(ellipse1)+(yenikonum.Y-
ilkkonum.Y));
        Canvas.SetLeft(ellipse1, Canvas.GetLeft(ellipse1) + (yenikonum.X -
ilkkonum.X));
    }
}
private void ellipse1_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    surukleniyor = false;
}
```

Bu program kodlarının çalışma yöntemi ise şöyledir:

Kullanıcı elipsin üzerinde farenin sol tuşuna bastığı anda **MouseLeftButtonDown** olayı tetiklenir. Bu olay içerisinde surukleniyor değişkeni true yapılır ve farenin elips üzerindeki koordinatları ilkkonum isimli değişkene alınır.

Kullanıcı fareyi her hareket ettirdiğinde farenin elips üzerindeki koordinatları yenikonum isimli değişkene alınır. Ve ayrıca surukleniyor değişkeni kontrol edilir. Eğer bu değişken true ise farenin sol tuşu basılı haldeyken hareket ettiriliyor demektir. Bu şarta bağlı olarak yenikonum ile ilk konum arasındaki fark kadar elipsin konumu kaydırılır.

Kullanıcı fareyi bıraktığında surukleniyor değişkeni false yapılır.

Ekran görüntüsü: Nesnenin fareyle sürüklenbildiği görülecektir.



Resim 2.6: Ekran görüntüsü

UYGULAMA FAALİYETİ

Uygulama: Yönergeleri takip ederek Silverlight ile temel sürükle-bırak işlemlerini içeren bir gelişmiş internet uygulaması hazırlayınız.

İşlem Basamakları	Öneriler
➤ Proje editör programını kullanarak yeni bir Silverlight projesi oluşturunuz.	➤ File→New Project→Silverlight→Silverlight Application
➤ MainPage.xaml sayfası üzerine bir elips ekleyiniz.	➤ İç dolgu rengi ve boyutlarını gerekli şekilde ayarlayınız.
➤ Elipsin MouseLeftButtonDown metodunu oluşturunuz.	➤ Elips seçildikten sonra Properties panelindeki Events bölümünü kullanınız.
➤ Metod içine gerekli kodları yazınız.	<pre>private void ellipse1_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) { surukleniyor = true; konum = e.GetPosition(ellipse1); }</pre>
➤ Elipsin MouseMove metodunu oluşturunuz.	➤ Elips seçildikten sonra Properties panelindeki Events bölümünü kullanınız.
➤ Metod içine gerekli kodları yazınız.	<pre>private void ellipse1_MouseMove(object sender, MouseEventArgs e) { Point mesafe=e.GetPosition(ellipse1); if (surukleniyor) { Canvas.SetTop(ellipse1,Canvas.GetTop(ellipse1)+(mesafe.Y-konum.Y)); Canvas.SetLeft(ellipse1, Canvas.GetLeft(ellipse1) + (mesafe.X - konum.X)); } }</pre>
➤ Elipsin MouseLeftButtonUp metodunu oluşturunuz.	➤ Elips seçildikten sonra Properties panelindeki Events bölümünü kullanınız.
➤ Metod içine gerekli kodları yazınız.	<pre>private void ellipse1_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) { surukleniyor = false; }</pre>
➤ Projeyi çalıştırınız.	➤ F5 tuşunu kullanabilirsiniz.

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Olay ve metod kavramlarını kavrayabildiniz mi?		
2. Kontrollere olay ekleme işlemini gerçekleştirebildiniz mi?		
3. Sürükle-Bırak işlemini sorunsuz bir şekilde gerçekleştirebildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme” bölümüne geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi XAML tabanlı zengin internet uygulamalarında kullanılan bir olaydır?
A) MouseMove
B) Row
C) Column
D) Width
2. Bir nesneyle ilgili belirli durumlarda gerçekleştirilecek işlemlerin tanımlanacağı kod sayfasına erişmek için editör içinde hangi kısım kullanılır?
A) Objects AndTimeline paneli
B) Events paneli
C) Properties penceresi
D) Solution Explorer sekmesi
3. Bir nesnenin sol konum değerini ayarlamak için aşağıdaki metodlardan hangisi kullanılır?
A) Canvas.SetLeft()
B) Top
C) Left
D) Canvas.SetTo()
4. Bir nesnenin üst konum değerini ayarlamak için aşağıdaki metodlardan hangisi kullanılır?
A) Top
B) Left
C) Canvas.SetTop()
D) Canvas.SetLeft()
5. Bir nesnenin üzerinde fare ile hareket ettirilmesi durumunda aşağıdaki olaylardan hangisi çalışır?
A) Click
B) MouseLeave
C) MouseMove
D) MouseOut

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Yükleme sayfasının XAML projesine dahil edilmesinde hangisi kullanılır?
A) C# C) HTML
B) XAML D) XML
2. Aşağıdakilerden hangisi yükleme sayfası olarak kullanılmaya uygun **değildir**?
A) JOption sayfası
B) XAML sayfası
C) HTML sayfası
D) Grid kontrolü
3. Aşağıdakilerden hangisi bir sürükle-bırak işleminde kullanılan temel olaylardan birisi **değildir**?
A) MouseLeftButtonDown C) MouseLeftButtonUp
B) MouseEnter D) MouseMove
4. Aşağıdakilerden hangisi XAML tabanlı gelişmiş *Internet* uygulamalarında hata yönetimi için **kullanılmaz**?
A) Bir XAP dosyası C) Visual Studio 2010 Debugger
B) C# fonksiyonları D) JavaScript fonksiyonları

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

5. () Söz diziminden kaynaklı sorunlar derleme hatası olarak isimlendirilir.
6. () Visual Studio 2010 içerisinde entegre bir Silverlight hata ayıklayıcısı mevcuttur.
7. () Uygulamanın kullanıcıyla etkileşim kuran kısmına arayüz adı verilir.
8. () Kontroller içerisinde buldukları Layout kontrolünden bağımsız olarak daima aynı Özellikler ile tanımlanır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

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

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	A
2	B
3	A
4	C
5	C

MODÜL DEĞERLENDİRME'NİN CEVAP ANAHTARI

1	B
2	A
3	B
4	A
5	Doğru
6	Doğru
7	Doğru
8	Yanlış

KAYNAKÇA

- KARAKAŞ, Yusuf, **Silverlight'a Giriş**, İstanbul, 2010.
- GHODA, Ashish, **IntroducingSilverlight 4**, <http://it-ebooks.info/book/461/>, (10.04.2013 / 14:00)