

# WINDOWS FORM KONTROLLERİ III



## İÇİNDEKİLER

- İleri Düzey Kontroller
  - Listview Kontrolü
  - Treeview Kontrolü
  - Tabcontrol Kontrolü
  - Notifyicon Kontrolü
  - Menustrip Kontrolü
  - Contextmenustrip Kontrolü
  - Toolstrip Kontrolü
  - Statusstrip Kontrolü



## HEDEFLER

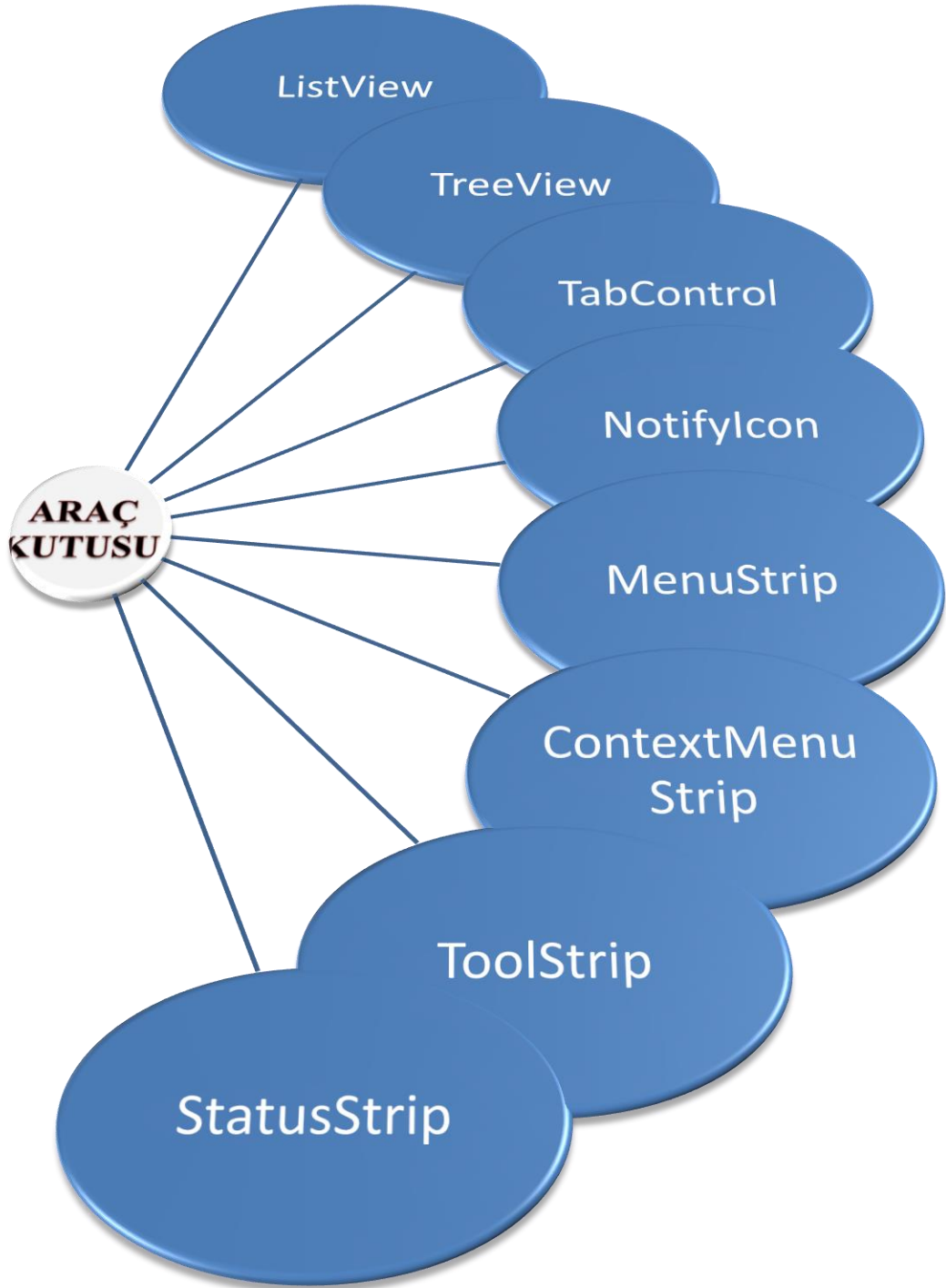
- Bu üniteyi çalıştıktan sonra;
  - Araç kutusundaki kontrolleri tanıyabilecek,
  - Windows form kontrolleri ile ileri düzey işlemler yapabilecek,
  - Projelerde kontrolleri etkili bir şekilde kullanabileceksiniz.



**Atatürk Üniversitesi**  
Açıköğretim Fakültesi

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

**ÜNİTE  
10**



## GİRİŞ

Microsoft tarafından geliştirilen .Net platformunda masaüstü, web, bulut, mobil gibi birçok türden proje geliştirebiliriz. Geliştirdiğimiz projeler hangi türde olursa olsun projede kullanılan her türlü seçenek özenle hazırlanmalıdır. Aksi takdirde proje amacına hizmet etmiş olsa dahi kullanımının etkililiğinde soru işaretleri oluşabilir.

Kaliteli bir proje hazırlamak için programlama bilgisine sahip olmakla beraber kullanılan uygulama geliştirme ortamının da iyi bilinmesi gerekmektedir. Bu bilgi birikimi, hazırladığımız projelerde en doğru seçeneği en doğru yerde kullanmamızı sağlayacaktır. Bu durum programlama mantığının temelinde olan en doğru çözüme en hızlı yoldan ulaşma anlayışından kaynaklanmaktadır.

Windows form uygulamalarımızı hazırlarken de kullanabileceğimiz kontrolleri ne kadar iyi tanırsak ve o kontrollerin kullanım amaçlarını ne kadar iyi bilirsek projelerimizi de o kadar iyi hazırlayabiliriz. Aynı zamanda kontrollerin özellikleri, metotları ve olayları hakkında bilgi sahibi olmamız bize kolaylık sağlayacaktır. Bazı işlemler gerçekleştirilirken birden fazla seçenek mevcut olabilir. Bunlar içinde en doğru ve sonuca en hızlı olanı tercih etmemiz gerekir. Ancak kontrollerin ne işe yaradığını veya özelliklerini bilmek her zaman yeterli olmayabilir. Programcının proje için özgün ve yaratıcı fikirler de üretmesi gerekmektedir.

Önceki ünitelerde Windows form uygulamaları hazırlanırken kullanılan standart kontroller ve gelişmiş kontrollerden bahsedilmiştir. Bu ünite de projelerimizin arayüzlerini daha zengin, etkili ve göze hitap eden bir şekilde tasarlayabilmek için ileri düzey kontrollerden bahsedilecektir. Bu kontroller;

- ListView
- TreeView
- TabControl
- NotifyIcon
- MenuStrip
- ContextMenuStrip
- ToolStrip
- StatusStrip



Visual Studio 2019 programında araç kutusunda benzer özellikleri taşıyan kontroller sekmeler halinde sunulmuştur.

## İLERİ DÜZEY KONTROLLER

Visual Studio 2019 programında araç kutusunda benzer özellikleri taşıyan kontroller sekmeler halinde sunulmuştur. Bununla beraber ilk sekme olan **Tüm Windows Formları** seçeneğinde bütün kontroller listelenmiştir. Ünitelerde kontroller anlatılırken basitten karmaşığa doğru ilerleyen bir sıra takip edildiği için sekmelere göre değil kontrollerin özelliklerine ve kullanım sıklıklarına göre anlatım sırası belirlenmiştir.

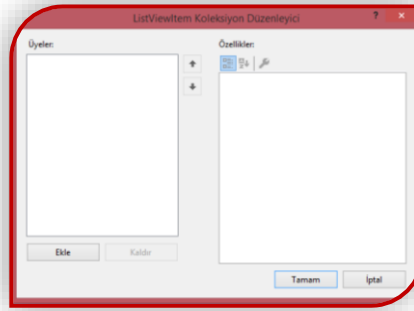
## ListView Kontrolü

Listesini oluşturmak istediğimiz elemanların ListBox kontrolünde görünmesini sağlayabiliriz. Ancak bu kontrolün yetersiz olduğu durumlar olabilir. Örneğin elemanları sadece satırlar halinde değil sütunlar halinde de listelemek isteyebiliriz. Bu durumda ListView kontrolünü kullanmak gerekecektir. Aynı zamanda ListView kontrolü elemanların simge olarak listelenmesini sağlamaktadır.



ListView kontrolüne eleman eklemek için özellikler panelinde bulunan Items seçeneğini kullanabiliriz.

ListView kontrolüne elemanları yerleştirmek için özellikler panelinde bulunan Items özelliğini kullanabiliriz. Aynı şekilde form üzerine eklediğimiz ListView kontrolünün sağ üst köşesinde bulunan ok'a tıklayıp açılan pencereden **Öğeleri Düzenle...** seçeneğini işaretleyerek de ListView kontrolüne eleman ekleyebiliriz. Bu işlemlerden herhangi biri tercih edildikten sonra karşımıza Şekil 10.1.'de gösterilen **ListViewItem Koleksiyon Düzenleyici** penceresi gelecektir. Bu pencerede bulunan **Ekle** ve **Kaldır** butonları sayesinde kontrolümüze yeni elemanlar ekleyip, seçili olan elemanları kaldırabiliriz. Eklediğimiz her bir elemanın değerini, ListViewItem Koleksiyon Düzenleyici penceresinin sağ tarafında bulunan özellikler panelindeki **Text** özelliğinden girebiliriz.

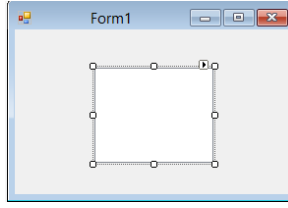


**Şekil 10.1.** ListViewItem Koleksiyon Düzenleyici Penceresi

Bu kontrole çalışma anında gerçekleştirilen bir etkileşim ile eleman ekleme imkanına da sahibiz. Bu işlemi gerçekleştirmek için **Items.Add()** metodunu kullanmamız gerekmektedir. Ayrıca çalışma anında kontrol üzerinde seçilen bir elemanın **Text** özelliğini kullanmak isteyebiliriz. String veri tipinde görüntülenen bu değere kontrolün **SelectedItem** özelliği ile ulaşılabilir.

## TreeView Kontrolü

Form üzerindeki görüntüsü ListBox kontrolüne benzer şekilde olan TreeView kontrolünü, içindeki bilgilerimizi hiyerarşik bir yapıda görüntüleyerek kullanıcının seçimine sunmak için kullanabiliriz. İçerisine herhangi bir bilgi girilmemiş TreeView kontrolünün ekran görüntüsü Şekil 10.2.'de gösterilmektedir.

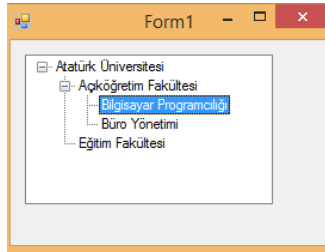


Şekil 10.2. Treeview Kontrolü



TreeView kontrolüne eklediğimiz kök ve alt öğelerin isimlerini, pencerenin sağ tarafında bulunan özellikler alanındaki Text seçeneğinden düzenleyebiliriz.

TreeView kontrolünün içine değer girebilmek için özellikler panelinde bulunan **Nodes** özelliğine tıklamamız gerekmektedir. Aynı pencereye tasarım ekranındayken kontrolün sağ üst köşesinde bulunan ok'a tıkladıktan sonra açılan alan üzerinde **Düğmeleri Düzenle...** seçeneğiyle de ulaşmamış mümkündür. Bu işlemin ardından **TreeNode Düzenleyicisi** penceresi açılır. Bu pencerede bulunan **Kök Ekle** butonu ile kontrolümüzde görüntülemek istediğimiz ana seçenekleri oluşturabiliriz. Girilen her ana kök için seçim işlemi yapıldıktan sonra **Alt Öğe Ekle** butonuna tıklanarak alt seçenekleri belirlenebilir. Eklediğimiz kök ve alt öğelerin isimlerini, pencerenin sağ tarafında bulunan özellikler alanındaki **Text** seçeneğinden düzenleyebiliriz. Ayrıca kökler eklenirken sıralamanın veya hiyerarşinin yanlış yapıldığı durumlarda ilgili kök veya alt öğe seçildikten sonra **TreeNode Düzenleyicisi** penceresinde bulunan yukarı veya aşağı yönlü ok işaretlerinin bulunduğu butonlar yardımıyla düzenleme yapabilirsiniz. Üzerinde çarpı işareti bulunan butonu kullanarak da seçtiğiniz bir kök veya alt öğeyi listeden kaldırabilirsiniz. Yaptığınız değişikliklerin ardından **tamam** butonuna tıklayarak tasarım ekranına dönebilirsiniz. Şekil 10.3.'de verilen kök ve alt öğeler eklenmiş bir TreeView kontrolünün ekran görüntüsünü inceleyebilirsiniz.

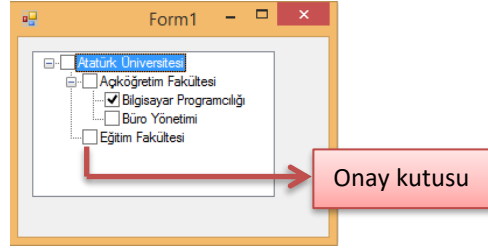


Şekil 10.3. Treeview Kontrolü

TreeView kontrolü hazırlandıktan sonra köklerin yanında artı (+) ve eksi (-) işaretlerini görebilmekteyiz. Bu işaretlere tıklanarak köklerin alt öğeleri gösterilmekte veya gizlenmektedir.

Kontrol form üzerine eklendikten sonra sağ üst köşede bulunan buton yardımıyla veya özellikler paneli ile daha önceden hazırlanmış bir **ImageList** kontrolü bağlanabilir. Bu kontrolün özellikleri bir önceki ünite anlatılmıştır. **ImageList** içerisindeki resimler seçilen kök veya alt öğe için kullanılabilir. Bu işlem özellikler panelinde bulunan **ImageIndex** veya **ImageKey** seçenekleri ile gerçekleştirilebilir.

TreeView kontrolü içinde bulunan kök ve alt öğelerin seçilebilir özellikler halinde görüntülenmesi sağlanabilir. İlgili kontrolün özellikler panelinde bulunan **CheckBoxes** seçeneğini **True** değerini alacak şekilde düzenlememiz gerekir. Böyle bir TreeView kontrolüne ait ekran görüntüsü Şekil 10.4.'de gösterilmektedir.



Şekil 10.4. Checkboxes = True

## TabControl Kontrolü

Windows form uygulamalarında çoklu formlarla çalışılabilmektedir. Farklı formlar açarak yapacağımız işlemleri tek bir form üzerinden de gerçekleştirebiliriz. Yani form alanımız içerisinde sekmelerin olması ve bu sekmeler yardımıyla farklı veya aynı kontrolleri formumuza yerleştirebiliriz. Böyle bir durumu gerçekleştirebilmek için TabControl kontrolünü kullanmamız gerekmektedir.

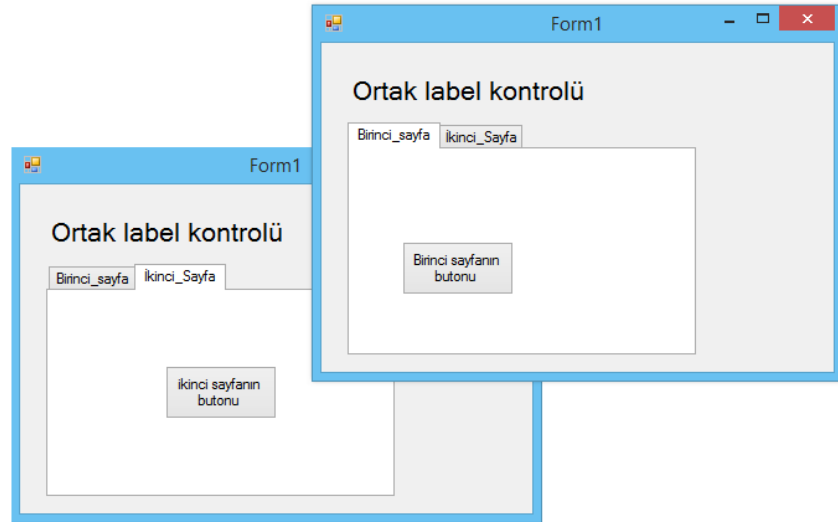


TabControl kontrolüne yeni sekmeler eklemek için özellikler panelinde bulunan TabPages seçeneğine tıklamamız gerekmektedir.

TabControl kontrolü formumuza eklendikten sonra boyutları formun tamamına sığacak şekilde ayarlanabilir. Bununla beraber formun belirli kısımlarını sekmeler arasında geçiş yaparken sabit kontroller ile tasarlayarak uygulamamızın içeriğine göre çeşitli amaçlarla kullanabiliriz. Örneğin bir Excel dosyası için her sayfada bulunan satır numaraları ve sütun harfleri, form üzerinde sabit bir şekilde tasarlanarak aynı görüntü elde edilmeye çalışılabilir. Sekmeler de bir Excel dosyasında olduğu gibi alt tarafta listelenebilir. Sekmelerin yerlerini değiştirebilmek için özellikler panelinde bulunan Alignment seçeneği kullanılmalıdır. Burada *Top*, *Bottom*, *Right* ve *Left* olmak üzere dört farklı seçenek mevcuttur. Tasarımımıza bağlı olarak bunlardan herhangi biri tercih edilebilir.

Kontrolle yeni sekmeler eklemek için özellikler panelinde bulunan *TabPage* seçeneğine tıklamamız gerekmektedir. Bu işlemin ardından *TabPage Koleksiyon Düzenleyici* penceresi açılacaktır. Kontrolde var olan sekmeler pencerenin sol tarafındaki *Üyeler* alanında görüntülenir. Pencerede bulunan *Ekle* butonuna tıklayarak yeni sekmeler ekleyebilir, *Kaldır* butonuna tıklayarak seçili olan sekme kaldırılabilir. Sekmelerin isimlerini düzenleyebilmek için TabPage Koleksiyon Düzenleyici penceresinde sağ tarafta bulunan özellikler alanındaki *Text* seçeneği kullanılmaktadır.

GroupBox kontrolünde olduğu gibi sekmeler üzerinde görüntülenmek istenen kontroller sürükleyip bırak yöntemi ile kontrolün üzerine eklenebilmektedir. Şekil 10.5.'de farklı sekmeler üzerine eklenmiş kontrollerin bulunduğu TabControl kontrolüne ait ekran görüntüsü bulunmaktadır.



Şekil 10. 5. Tabcontrol Üzerinde Bulunan Farklı Kontroller ve Form Üzerinde Bulunan Ortak Kontrol

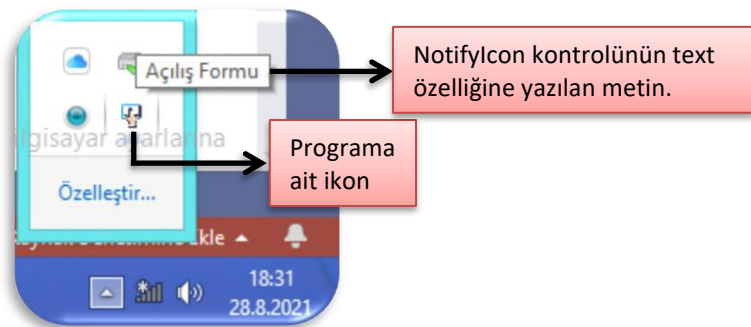
## NotifyIcon Kontrolü

Hazırladığımız uygulamaya ait belirlediğimiz bir simgenin görev çubuğunda sistem saatin olduğu bölgede görüntülenmesini sağlayabiliriz. Forma eklemiş olduğumuz NotifyIcon kontrolü ile bu işlem gerçekleştirilmektedir. Eklediğimiz kontrol form üzerinde görüntülenmez. Formun alt tarafında açılan panel üzerinden kontrole erişebiliriz.



NotifyIcon kontrolünün Icon özelliği ile .ico uzantılı bir resim yükleyerek proje simgesini kendimiz belirleyebiliriz.

NotifyIcon kontrolü eklendikten sonra özellikler panelinde bulunan *Icon* seçeneği ile .ico uzantılı bir resim yükleyerek istediğimiz simgenin görüntülenmesini sağlayabiliriz. Bu simge uygulamanın çalışma anında görüntülenmektedir. Ayrıca uygulama çalışırken simge üzerine fare ile geldiğimizde çıkacak olan metni belirleyebiliriz. Bu metin kontrolün özellikler panelinde bulunan *Text* seçeneğinden ayarlanmaktadır. Text özelliği “*Açılış Formu*” olarak ayarlanmış bir NotifyIcon kontrolünün ekran görüntüsü Şekil 10.6.’da verilmiştir.



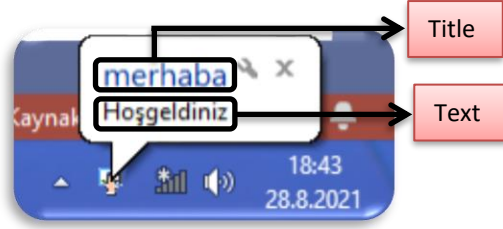
Şekil 10. 6. Icon ve Fare ile Görüntülenen Metin

Bazı durumlarda bu simgenin belirli şartlara bağlı olarak görüntülenmesini veya gizlenmesini isteyebiliriz. Bu işlemi özellikler panelinde bulunan *Visible* seçeneği ile gerçekleştirebileceğimiz gibi kod editöründe de yapabiliriz. Simgenin gösterilmesini istediğimiz etkileşim alanına *notifyIconAdi.visible = True;* kodunu

yazmamız gerekmektedir. Gizlenmesi için de kod kısmında bulunan True değeri yerine **False** değerini aktarmalıyız.




Uygulama çalışırken NotifyIcon simgesinin üzerine sağ tıklayarak belirli komutların gerçekleştirilmesi sağlanabilir. Böyle bir etkileşim sağlayabilmek için uygulamamıza ContextMenuStrip kontrolünü eklememiz gerekmektedir. Bu kontrolün özelliklerine ünitenin ilerleyen sayfalarında yer verilecektir. Burada yapmamız gereken işlem ContextMenuStrip kontrolü eklendikten sonra NotifyIcon kontrolünün özellikler panelinden bağlantısının gerçekleştirilmesidir.

NotifyIcon kontrolünün text özelliğine yazdığımız metin ile kullanıcıya bilgilendirme işlemi gerçekleştirebileceğimizden bahsetmiştik. Buna benzer bir işlem, Şekil 10.7.'de gösterildiği gibi balon (bildirim) yardımıyla da gerçekleştirilebilir. Özellikler panelinde üç farklı seçenek ile bildirim ayarları yapılabilir. Bunlar BalloonTipIcon, BalloonTipText ve BalloonTipTitle seçenekleridir.



Şekil 10.7. BalloonTipText = "Hoşgeldiniz"; BalloonTipTitle = "merhaba";

**BalloonTipIcon**: Açılacak olan bildirimin türünü ayarlayacağımız özelliktir. Bu özelliğe ait 4 farklı alternatif bulunmaktadır. Bunlar;

- **None**: Bildirim başlığında herhangi bir simgenin bulunmadığı ve bildirim türünün belirtilmediği seçenektir.
- **Info**: Bildirim başlığında  simgesinin bulunduğu ve **bilgilendirme** türünde bildirim yapıldığı zaman kullanılan seçenektir.
- **Warning**: Bildirim başlığında  simgesinin bulunduğu ve **uyarı** türünde bildirim yapıldığı zaman kullanılan seçenektir.
- **Error**: Bildirim başlığında  simgesinin bulunduğu ve **hata** türünde bildirim yapıldığı zaman kullanılan seçenektir.

**BalloonTipText**: Bildirim alanında açıklama yapacağımız metinleri gireceğimiz özelliktir. Buraya yazacağımız metinler başlığın alt kısmında görüntülenmektedir.

**BalloonTipTitle**: Bildirim başlığını yazacağımız özelliktir. Yazdığımız başlık, bildirim alanının üst satırında bildirim simgesinin yanında görüntülenmektedir.

NotifyIcon kontrolüne ait bildirim ayarları kontrolün gizli olmadığı durumlarda yani visible özelliğinin True olduğu durumlarda ortaya çıkmaktadır. Bununla birlikte bildirim alanının ekranda ne kadar kalacağını da belirtmemiz gerekmektedir. Bu işlem ilk parametresi zorunlu olan ve dört parametre alabilen **ShowBalloonTip()**; özelliği ile gerçekleştirilmektedir. Özellikler panelinde



NotifyIcon kontrolüne ait bildirim ayarları kontrolün visible özelliğinin True olduğu durumlarda ortaya çıkmaktadır.



bulunmayan bu seçeneğin parametreleri kod editöründen gerçekleştirilebilir. Zorunlu olan ilk parametresine, bildirimin kaç saniye ekranda görüneceğini *milisaniye* cinsinden yazmamız gerekmektedir. İsteğe bağlı olarak ShowBallonTip özelliğinin;

- İkinci parametresine bildirim başlığını,
- Üçüncü parametresine bildirim metnini,
- Dördüncü parametresine de bildirim türünü yazabilirsiniz.

ShowBalloonTip() özelliğinin kullanım şekli:

`NotifyIconAdı.ShowBalloonTip(intMs,stringBaşlık,stringMetin,ToolTipIcon.Simge);`

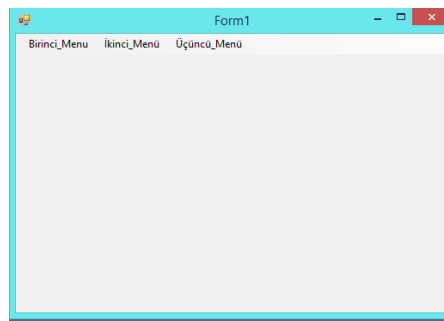
## MenuStrip Kontrolü

Windows formların bulunduğu hemen hemen uygulamada menülerin bulunduğunu gözlemleyebiliriz. Geniş bir kullanım alanına sahip olan menüler, .Net Framework 1.1'de MainMenu kontrolü kullanılarak hazırlanmaktaydı. Bu başlık altında anlatacağımız MenüStrip kontrolü ise ilk olarak .Net Framework 2.0'da kullanılmaya başlandı. Ancak .Net Framework 2.0'da başlangıçta araç kutusunda yer almamaktaydı. Uygulama geliştirme ortamlarının geliştirilmesiyle beraber MenuStrip kontrolü de rahat bir şekilde araç kutusundan form üzerine eklenebilmektedir.

MenuStrip kontrolünü uygulamamıza eklediğimiz zaman hem formun üst kısmında hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır. Formun içine eklenen MenuStrip kontrolündeki ilgili alanları doldurarak tasarım özellikleri hızlı bir şekilde ayarlayabiliriz. Menüde bulunmasını istediğimiz ilk seçeneği *Buraya Yazın* kısmına tıklayarak doldurabiliriz. Bu işlemin ardından yeni menü seçeneklerini girmek için sağ tarafta bulunan kutulara menü isimlerini yazarak menülerinizi oluşturabilirsiniz. Şekil 10.8.'de üç adet menüden oluşan bir formun ekran görüntüsü bulunmaktadır.



Menü komutlarını girmek için seçilen menünün altında bulunan Buraya Yazın alanını doldurmamız gerekmektedir.

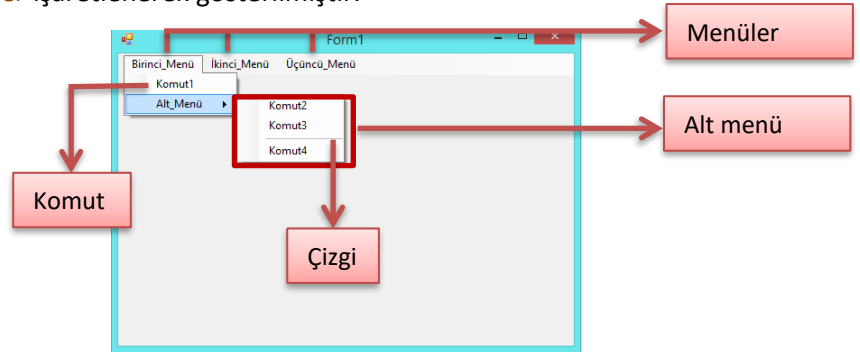


Şekil 10.8. Menustrip Kontrolü

Menü komutlarını girmek için seçilen menünün altında bulunan Buraya Yazın seçeneği ile bulunmasını istediğiniz komutları yazabilirsiniz. Benzer özellikler taşıyan komutların gruplandığını göstermek amacıyla komutlar arasında çizgiler konulabilir. Bu işlem, komutlar yazıldıktan sonra önceki komutlardan ayrı özellikler taşıdığı düşünülen komuta fare ile sağ tıklama işlemi yapıp açılan menüden *Ekle>Separator* adımları takip edilerek gerçekleştirilmektedir.

Menüleri ve komutları oluşturma işlemi özellikler panelinden de ayarlanabilmektedir. Menustrip kontrolü eklendikten sonra *Items* özelliği tıklanarak *Öğeler Koleksiyonu Düzenleyicisi* penceresi açılmalıdır. Bu pencerede bulunan *Ekle* butonu ile menüler eklenip *Text* özellikleri ile isimleri düzenlenebilmektedir. Eklenen menülere komutlar eklemek için de, ilgili menü seçildikten sonra özellikler panelinde bulunan *DropDownItems* özelliği tıklanmalıdır. Açılacak olan ikinci *Öğeler Koleksiyonu Düzenleyicisi* penceresi ilgili menünün komutlarını girmeyi sağlamaktadır. Bu işlemlerin tasarım ekranında planlanması biraz daha pratik bir şekilde gerçekleşir. Bu yüzden alt menü oluşturma işlemini tasarım ekranını kullanacağımız şekilde anlatmanın daha sağlıklı olduğu düşünülmektedir.

Oluşturduğumuz menülerin içinde bulunan seçenekleri komut olarak kullanabileceğimiz gibi alt menü olarak da ayarlayabiliriz. İlgili menünün alt menü olarak düzenlenmek istenen seçeneği işaretledikten sonra sağında bulunan *Buraya Yazın* kısmına tıklayarak komutlar girilebilir ve böylelikle o seçenek artık bir alt menü olarak görev yapmış olur. Aşağıda bulunan Şekil 10.9.'da *menü*, *komut*, *çizgi* ve *alt menüler* işaretlenerek gösterilmiştir.



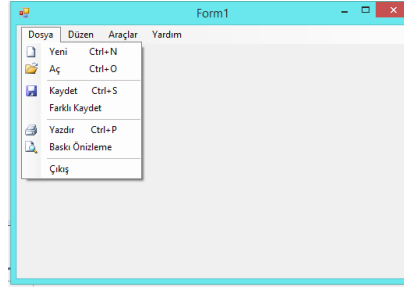
Şekil 10.9. Menü, Alt Menü, Komut ve Çizgi Örneği



MenuStrip kontrolünün sağ üst köşesinde bulunan ok'a tıklayıp Standart Öğe Ekle seçeneği işaretlenerek kullanıma hazır menüleri formumuza ekleyebiliriz.

Menülerimizi hazırlarken alt tarafında bulunan *Buraya Yazın* seçeneğine tıklayıp metin girdiğimiz zaman girilen metin komut olarak kabul edilmektedir. Ancak menülere komutlar dışında *TextBox*, *ComboBox* ve *Separator* ekleme imkanı da bulunmaktadır. Komut ekleme ve Separator ekleme işlemlerini gerçekleştirdik. Bir menü içine *TextBox* ve *ComboBox* eklemek için, ilgili menünün altında bulunan *Buraya Yazın* alanının sağ tarafındaki aşağı yönlü ok'a tıklayıp istediğimiz seçeneği işaretleyerek işlemimizi tamamlayabiliriz. Form üzerine eklediğimiz *TextBox* ve *ComboBox* kontrolleri ile yapılan işlemlerin aynısı menülerde de yapılabilir.

Birçok uygulamada ortak bir şekilde kullanılan Dosya, Düzen gibi menüler bulunmaktadır. Bu menüleri bahsedilen yöntemlerle tek tek hazırlamak yerine Form üzerine eklenen *MenuStrip* kontrolünün sağ üst köşesinde bulunan ok'a tıklayıp Standart Öğe Ekle seçeneği işaretlenerek bunları formumuza ekleyebiliriz. Kullanıma hazır bir şekilde eklenen menülerin ekran görüntüsü aşağıda bulunan Şekil 10.10.'da gösterilmektedir.



Şekil 10.10. Hazır Menüler

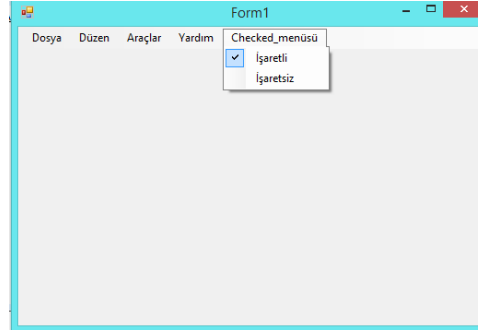
Hazır menüleri eklediğimiz alan içerisinde *RenderMode*, *Dock* ve *GripStyle* olmak üzere üç farklı seçenek bulunmaktadır. Bu seçeneklere özellikler panelinden de ulaşabiliriz. *RenderMode* seçeneği form üzerine eklenen menülerin çalışma anındaki görünümüyle alakalı seçenekleri ayarlamamızı sağlamaktadır. Bu seçeneğe ait *System*, *Professional* ve *ManagerRenderMode* olmak üzere üç farklı alternatif mevcuttur. Hazırlanan uygulamanın yapısı ve tasarımına göre istenilen bir seçenek tercih edilebilir. *Dock* seçeneği ise menülerin forma yerleşme durumunu ayarlamamızı sağlamaktadır. *GripStyle* ise çalışma anında menü çubuğunun konumunu değiştirmenizi sağlayacak olan tutamaçların gösterilmesini veya gizlenmesini sağlayan seçenektir.

Menülerinizin altında bulunan komutların işlevsel olabilmesi için gerekli kodlar yazılmalıdır. Tasarım ekranında ilgili komut üzerinde çift tıklayarak kod editörüne geçebilirsiniz. Açılan metot kalıbı içerisinde yaptırmak istediğiniz işlemlere yönelik kodları yazmanız gerekmektedir.



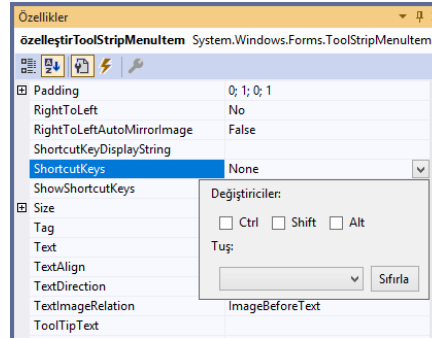
Menülerde kullanılan komutlar için de kısayol tuşları tanımlamak istiyorsak, ilgili komuta ait *ShortcutKeys* özelliğinden yararlanmamız gerekmektedir.

Bazı durumlarda menü içinde bulunan komutların işaretli olması istenebilir. Genelde o komutun devrede olup olmadığını anlayabilmek için kullanılmaktadır. Önünde onay işaretinin olmasını istediğimiz komuta tıklayıp özellikler panelinde bulunan *Checked* özelliğine *True* değerini atayarak tasarım ekranında u işlemi gerçekleştirmiş oluruz. Ancak sadece tasarım ekranında yapılan bu işlem çalışma anında o komutu sürekli olarak işaretli göstereceği için anlamsız bir sonuç ortaya çıkacaktır. Bu yüzden ilgili komut üzerinde çift tıklandıktan sonra kod editöründe tıklanma olayına yönelik açılacak olan metot kodlanmalıdır. Burada komutun işaretli olup olmadığına yönelik kontrol sağlandıktan sonra tıklamaya bağlı olarak durum değişikliği gerçekleştirilmelidir. Bu işlemin yapıldığı basit bir uygulamaya ait ekran görüntüsü Şekil 10.11.'de gösterilmiştir. Bu uygulama hazırlanırken tasarım alanında *Checked\_menüsü* isimli bir menü ve bu menüye ait *İşaretli* isimli bir komut hazırlanmıştır. İşaretli isimli komutun kod editöründe açılan *Click* metodunda; komut işaretliyse işaretsiz hale gelmesini, komut işaretli değilse işaretli hale gelmesini sağlayan kodlar komutun *Checked* özelliği kullanılarak yazılmıştır.



Şekil 10.11. Menülerde Checked Özelliği

Windows form uygulamalarında sıklıkla kullanılan işlemlerin daha hızlı bir şekilde yapılabilmesi ve klavyeyle çalışırken fareye dönmemesi gibi sebeplerden ötürü kısayol tuşları hazırlanabilmektedir. Menülerde kullanılan komutlar için de kısayol tuşları tanımlamak istiyorsak, ilgili komuta ait **ShortcutKeys** özelliğinden yararlanmamız gerekmektedir. Bununla beraber komuta ait **ShowShortcutKeys** özelliği de **True** değerini taşımalıdır. Bu özellikleri tasarım ekranında ayarlamak biraz daha kolay bir şekilde gerçekleşmektedir. Menülerde bulunan herhangi bir komut seçildikten sonra ShortcutKeys özelliği tıklanarak Şekil 10.12.'deki görüntü elde edilmektedir. Bu alan üzerinde kısayol tuşunu hangi yardımcı tuş ile eşleştirmek istiyorsak işaretlememiz gerekir. **Değiştiriciler** kısmında bulunan **Ctrl**, **Shift** ve **Alt** tuşlarından herhangi birini veya birkaçını seçebiliriz. Değiştiriciler kısmında belirttiğimiz tuşlarla beraber kullanılmak istenen tuş da, Tuş kısmında bulunan açılır listeden belirtilmelidir. Bu işlem tamamlandıktan sonra komutun hemen yanında ayarlamış olduğumuz kısayol tuşu görünecektir. Böylelikle çalışma anında, ayarlanan kısayol tuşuna bastığımız zaman komutun içinde tanımlamış olduğumuz işlem kendini gerçekleştirecektir.



Şekil 10.12. Shortcutkeys Özelliği



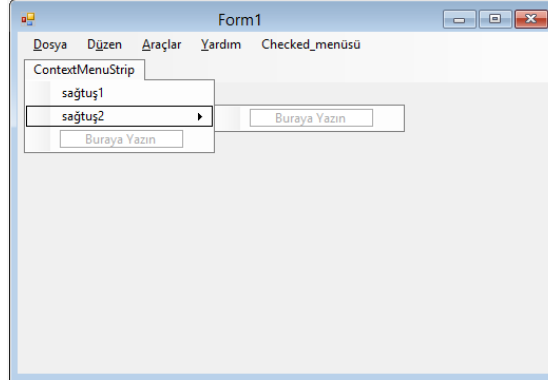
ContextMenuStrip kontrolü ile hazırlanan açılan menü, ilgili kontrolün özellikler panelinde bulunan ContextMenuStrip seçeneği ile bağlanabilmektedir.

## ContextMenuStrip Kontrolü

ContextMenuStrip kontrolü, hazırladığımız uygulamanın çalışma anında form üzerinde bulunan herhangi bir kontrol üzerinde farenin sağ tuşu ile açılan bir menü hazırlamamızı sağlamaktadır. Böylelikle kontrol üzerinde sağ tuş ile açılan menüde hazırlanan komutlardan herhangi birini kullanarak kullanıcıya yaptırmak istediğimiz seçenekleri belirleyebiliriz.

Kontroller üzerinde açılan bir menü oluşturmak için, tasarım ekranında çalışırken öncelikle araç kutusundan ContextMenuStrip eklememiz gerekmektedir. Eklenen ContextMenuStrip hem form üzerinde en üst kısımda hem Visual Studio

IDE penceresinin alt tarafında açılan panel üzerinde görüntülenecektir. Şekil 10.13.'de gösterildiği gibi kontrol yardımıyla oluşturacağımız oluşturmak istediğimiz komutları form üzerinden gerçekleştirebiliriz. Tasarım işlemi tamamlandıktan sonra komutların gerçekleştirmesini istediğimiz etkileşimi kod editöründen belirleyip ilgili kontrole bağlamamız gerekmektedir.

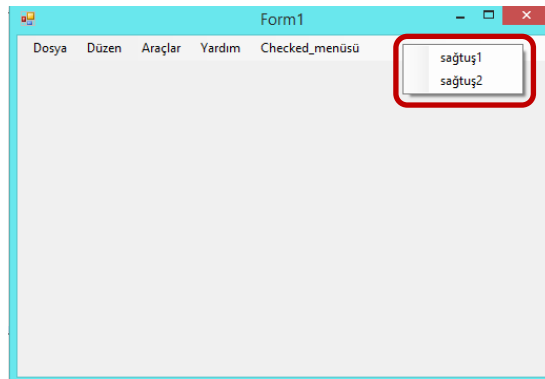


Şekil 10.13. Contextmenustrip Kontrolü

ContextMenuStrip kontrolü ile hazırlanan açılan menü, *ilgili kontrolün özellikler panelinde* bulunan *ContextMenuStrip* seçeneği ile bağlanabilmektedir. Bu bağlantı işlemi gerçekleştirildikten sonra uygulamayı çalıştırdığımız zaman kontrol üzerinde fare ile sağ tuşa tıklayarak hazırladığımız menüyü görüntüleyebiliriz. Şekil 10.14.'de menü çubuğuna bağlantısı yapılmış ContextMenuStrip kontrolü ile hazırlanan açılan menünün ekran görüntüsü bulunmaktadır. Bu görüntü uygulamanın çalışma anında menü çubuğu üzerinde farenin sağ tuşuna tıklandıktan sonra alınmıştır.



ToolStrip kontrolünün form üzerindeki konumunu değiştirmek için, özellikler panelinde bulunan Dock seçeneğini kullanabiliriz.



Şekil 10.14. Menüstrip Kontrolüne Bağlantısı Yapılmış Olan Contextmenüstrip Kontrolü

Uygulamayı çalıştırdığımız zaman ContextMenuStrip kontrolü ile verilen komutların C# uygulamasında bir tepki gösterebilmesi için kodlanması gerekmektedir. Bu işlemi tasarım ekranında çalışırken komut üzerinde çift tıklayarak geçiş yapacağımız kod editöründeki *Click* metoduna yazabiliriz. Örneğin açılan menüde “kes” ismi ile oluşturulan bir komutun çift tıklama ile kod editöründeki click metoduna *MessageBox.Show(“kes komutunu tıkladınız”);* yazarsak, uygulamayı çalıştırdığımızda hazırlanan mesajın ekrana geldiğini görebiliriz.

## ToolStrip Kontrolü

ToolStrip kontrolü, uygulamalarda genellikle konum olarak menü çubuğunun hemen altında bulunan araç çubuğunu oluşturmak için kullanacağımız kontroldür. Bu kontrolü kullanarak araç çubuğunun üzerine düğmeler (Button) ekleyebiliriz. Bazı durumlarda düğmeler yetersiz kalabilir. Bunun için ToolStrip kontrolü bize farklı seçenekler de sunmaktadır. Bunlar;

- Button
- Label
- SplitButton
- DropDownButton
- Separator
- ComboBox
- TextBox
- ProgressBar

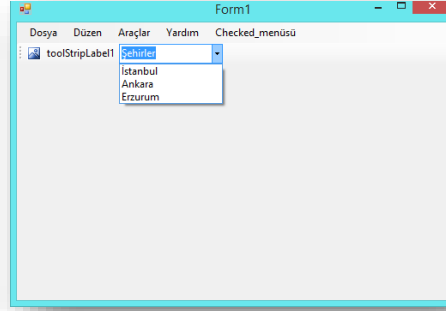
ToolStrip kontrolünü uygulamamıza eklediğimiz zaman hem formun üst kısmına hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır. Form üzerindeki konumunu değiştirmek için, kontrolü seçtikten sonra özellikler panelinde bulunan *Dock* seçeneğini kullanabiliriz.

Araç çubuğunun üzerine birçok uygulamada sıklıkla kullanılan seçenekleri hazır olarak ekleyebilirsiniz. ToolStrip kontrolünün sağ üst köşesindeki ok'a tıkladığınız zaman açılan menüden *Standart Öğeler Ekle* seçeneğini kullanmanız gerekmektedir. Bu işlemin ardından Yeni, Aç, Kaydet, Yazdır, Kes, Kopyala, Yapıştır ve Yardım olmak üzere 8 adet kullanıma hazır buton otomatik olarak eklenecektir. Bu seçeneklerden istediğinizi *Delete* tuşunu kullanarak silebilir, sürükleyip bırak yöntemi ile yerini değiştirebilir veya ToolStrip kontrolünde bulunan düğme ile yeni öğeler ekleyerek düzenleme işleminizi gerçekleştirebilirsiniz.

ToolStrip kontrolüne eklediğiniz kontrolleri özellikler panelinde bulunan *Items* seçeneği ile düzenleyebilirsiniz. Bu seçeneğe tıkladığınız zaman karşınıza *Öğeler Koleksiyonu Düzenleyicisi* penceresi açılacaktır. Tasarım ekranında gerçekleştirebildiğimiz silme, ekleme ve yerini değiştirme işlemlerini bu pencere üzerinden de gerçekleştirebiliriz. Ayrıca pencerenin sağ tarafında bulunan özellikler alanından ise her bir öğeye ait özellikleri düzenleyebilme imkanına da sahipsiniz. Örneğin araç kutusunda şehirlerin listeleneceği bir alan oluşturalım. Bu işlem için öncelikle araç kutusuna ComboBox kontrolünü eklememiz gerekmektedir. Ardından araç kutusuna ait Öğeler Koleksiyonu Düzenleyicisi penceresini açıp *Üyeler* kısmında bulunan ComboBox kontrolünü seçmeliyiz. Bu kontörle ait *Items* özelliği ile listelenmesini istediğimiz şehirleri, *Text* özelliğiyle de liste başlığını yazıp uygulamamızı çalıştırdığımızda Şekil 10.15.'de gösterildiği gibi bir ekran görüntüsü elde edebiliriz.



DisplayStyle özelliğiyle ToolStrip kontrolündeki öğelerin resimlerle gösterimine ek olarak isimlerinin de görünmesi sağlayabiliriz.



Şekil 10.15. ToolStrip Combobox Kontrolü

Özellikle butonlardan oluşan araç kutularının görselliğini tamamlayabilmek için kontroller üzerine resimler eklenebilmektedir. Resim eklenilmek istenen kontrol seçildikten sonra özellikler panelinde bulunan *Image* seçeneği kullanılmalıdır. Açılan *Kaynak Seç* penceresinde *Yerel kaynak* veya *Proje kaynak* dosyası seçeneklerinden her hangi biri tercih edilerek resim yükleme işlemi gerçekleştirilir ve *Tamam* butonuna tıklanır.

ToolStrip kontrolündeki öğelerin resimlerle gösterimine ek olarak isimlerinin de görünmesi istenebilir. Bunun için ilgili öğeye ait özellikler panelinde bulunan *DisplayStyle* özelliğinin kullanılması gerekmektedir. Burada *None*, *Text*, *Image* ve *ImageAndText* olmak üzere dört farklı alternatif sunulmaktadır.

- *None*: Hiçbir belirtinin olmadığı boş bir kutu gösterimidir.
- *Text*: Sadece metin bilgisinin yer aldığı gösterimdir.
- *Image*: Sadece yüklenen resmin yer aldığı gösterimdir.
- *ImageAndText*: Hem yüklenen resmin hem de metnin yer aldığı gösterimdir.

Ayrıca kontrol üzerindeki resmin boyutunu, eklenen yazının açısını ve eklenen yazının resim üzerindeki yerleşimini de ayarlayabiliriz. Bu işlemler için kontrolün *ImageScaling*, *TextDirection* ve *TextImageRelation* özelliklerini kullanmamız gerekmektedir.

ImageScaling özelliği *None* ve *SizeToFit* olmak üzere iki farklı alternatif sunmaktadır. None seçeneği resmin orijinal boyutu ile gösterilmesini sağlamaktadır. SizeToFit seçeneği ise resmin araç çubuğunun boyutuna göre otomatik olarak boyutlanmasını sağlamaktadır.

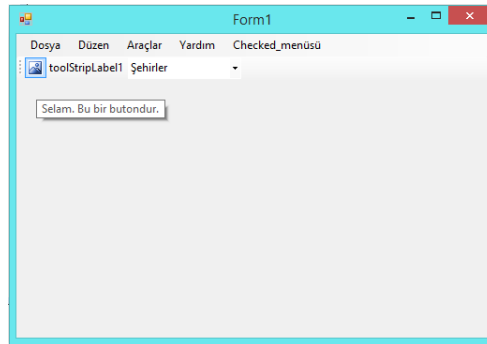
TextDirection özelliği *Inherit*, *Horizontal*, *Vertical90* ve *Vertical270* olmak üzere dört farklı alternatif sunmaktadır. Inherit seçeneği, hiyerarşik yapıların olduğu durumlarda özelliğin alt seçeneklere de uygulanması için kullanılan kalımla alakalı bir özelliktir. Horizontal seçeneği, resme ait metin bilgisinin yatay olarak gösterilmesini sağlamaktadır. Vertical90 seçeneği, yazının dikey olarak yukarıdan aşağıya doğru gösterilmesini sağlamaktadır. Vertical270 seçeneği ise yazının dikey olarak aşağıdan yukarıya doğru görüntülenmesini sağlamaktadır.



ToolStrip üzerinde bulunan kontrollerin metin-resim ilişkisi **TextDirection**, **TextImageRelation** gibi özellikler kullanılarak ayarlanabilmektedir.

**TextImageRelation** özelliği *Overlay*, *ImageAboveText*, *TextAboveImage*, *ImageBeforeText* ve *TextBeforeImage* olmak üzere beş farklı alternatif sunmaktadır. *Overlay* seçeneği işaretlendiğinde yazı resmin ortasına konumlanacaktır. *ImageAboveText* seçeneği işaretlendiğinde resim yazının üstünde konumlanır. *TextAboveImage* seçeneği işaretlendiğinde yazı resmin üstünde konumlanır. *ImageBeforeText* seçeneği işaretlendiğinde resim yazının solunda konumlanır. *TextBeforeImage* seçeneği işaretlendiğinde de yazı resmin solunda konumlanır.

Çalışma anında **ToolStrip** kontrolü üzerinde bulunan öğelerin üzerine fare ile gelip beklenildiği zaman açıklama metinleri eklenebilmektedir. Bu işlem için ilgili öğeye ait özellikler panelinde bulunan *ToolTipText* seçeneği kullanılmaktadır. Aşağıdaki Şekil 10.16.'da açıklama metnine "*Selam*" yazılan bir kontrolün ekran görüntüsü bulunmaktadır.



Şekil 10.16. Tooltiptext Özelliği



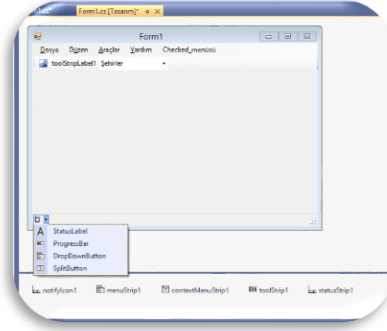
**StatusStrip** kontrolü üzerinde anlık olarak ilerleyen durumun bilgisi, **StatusLabel** kontrolü eklenerek kullanıcıya gösterilebilmektedir.

## StatusStrip Kontrolü

Uygulamanın çalışma anında kullanıcıya bilgi vermek amacıyla **StatusStrip** kontrolü kullanılabilir. Bu kontrol form üzerine eklendiği zaman hem formun alt tarafında hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır.

Şekil 10.17.'de gösterildiği gibi form üzerine eklenen kısımdaki buton yardımıyla durum çubuğuna *StatusLabel*, *ProgressBar*, *DropDownButton* ve *SplitButton* kontrolleri eklenebilir. *ProgressBar* kontrolünü anlatılırken etkili bir şekilde kullanabilmek için **Timer** kontrolü eklenip süre ilerlemesi için **Interval** özelliği kullanılarak milisaniye cinsinden belirtilmiştir. Buradaki kullanımı da aynı şekilde olduğu için tekrar değinilmeyecektir.





**Şekil 10.17.** ToolStrip Üzerine Eklenebilecek Kontroller

StatusStrip kontrolü üzerinde anlık olarak ilerleyen durumun bilgisi, *StatusLabel* kontrolü eklenerek kullanıcıya gösterilebilmektedir. Ayrıca çalışma daha önceden durum çubuğunda gösterimi sağlanan seçeneklerin bilgisi de *DropDownButton* kontrolü kullanılarak listelenebilmektedir.



**Bireysel Etkinlik**

- Form alanınızı bir tarayıcı yapısına benzer şekilde düzenleyerek bu alan üzerinde aratmış olduğunuz web sayfalarının isimlerini DropDownButton ekleyerek görüntüleyebileceğiniz bir uygulama geliştirin.



## Özet

- ListView Kontrolünün çalışma anında gerçekleştirilen bir etkileşim ile eleman ekleme imkanına da sahibiz. Bu işlemi gerçekleştirmek için Items.Add() metodunu kullanmamız gerekmektedir. Ayrıca çalışma anında kontrol üzerinde seçilen bir elemanın Text özelliğini kullanmak isteyebiliriz. String veri tipinde görüntülenen bu değere kontrolün SelectedItems özelliği ile ulaşılabilir.
- Form üzerindeki görüntüsü ListBox kontrolüne benzer şekilde olan TreeView kontrolünü, içindeki bilgilerimizi hiyerarşik bir yapıda görüntüleyerek kullanıcının seçimine sunmak için kullanabiliriz. Kontrolün içine değer girebilmek için özellikler panelinde bulunan Nodes seçeneğine tıklamamız gerekmektedir. Aynı pencereye tasarım ekranındayken kontrolün sağ üst köşesinde bulunan ok'a tıkladıktan sonra açılan alan üzerinde Düğmeleri Düzenle... seçeneğiyle de ulaşmamış mümkündür. Bu işlemin ardından TreeNode Düzenleyicisi penceresi açılır. Bu pencerede bulunan Kök Ekle butonu ile kontrolümüzde görüntülemek istediğimiz ana seçenekleri oluşturabiliriz. Girilen her ana kök için seçim işlemi yapıldıktan sonra Alt Öğe Ekle butonuna tıklanarak alt seçenekleri belirlenebilir. Eklediğimiz kök ve alt öğelerin isimlerini, pencerenin sağ tarafında bulunan özellikler alanındaki Text seçeneğinden düzenleyebiliriz. Ayrıca kökler eklenirken sıralamanın veya hiyerarşinin yanlış yapıldığı durumlarda ilgili kök veya alt öğe seçildikten sonra TreeNode Düzenleyicisi penceresinde bulunan yukarı veya aşağı yönlü ok işaretlerinin bulunduğu butonlar yardımıyla düzenleme yapabilirsiniz.
- TabControl Kontrolüne yeni sekmeler eklemek için özellikler panelinde bulunan TabPages seçeneğine tıklamamız gerekmektedir. Bu işlemin ardından tabPage Koleksiyon Düzenleyici penceresi açılacaktır. Kontrolde var olan sekmeler pencerenin sol tarafındaki Üyeler alanında görüntülenir. Pencerede bulunan Ekle butonuna tıklayarak yeni sekmeler ekleyebilir, Kaldır butonuna tıklayarak seçili olan sekme kaldırılabilir. Sekmelerin isimlerini düzenleyebilmek için tabPage Koleksiyon Düzenleyici penceresinde sağ tarafta bulunan özellikler alanındaki Text seçeneği kullanılmaktadır.
- NotifyIcon kontrolüne ait bildirim ayarları kontrolün gizli olmadığı durumlarda yani visible özelliğinin True olduğu durumlarda ortaya çıkmaktadır. Bununla birlikte bildirim alanının ekranda ne kadar kalacağını da belirtmemiz gerekmektedir. Bu işlem ilk parametresi zorunlu olan ve dört parametre alabilen ShowBalloonTip(); özelliği ile gerçekleştirilmektedir. Özellikler panelinde bulunmayan bu seçeneğin parametreleri kod editöründen gerçekleştirilebilir. Zorunlu olan ilk parametresine, bildirimin kaç saniye ekranda görüneceğini milisaniye cinsinden yazmamız gerekmektedir.
- MenuStrip kontrolünü uygulamamıza eklediğimiz zaman hem formun üst kısmında hem de Visual Studio IDE penceresinin alt tarafında açılan panel üzerinde konumlanmaktadır. Formun içine eklenen MenuStrip kontrolündeki ilgili alanları doldurarak tasarım özellikleri hızlı bir şekilde ayarlayabiliriz. Menüde bulunmasını istediğimiz ilk seçeneği Buraya Yazın kısmına tıklayarak doldurabiliriz. Bu işlemin ardından yeni menü seçeneklerini girmek için sağ tarafta bulunan kutulara menü isimlerini yazarak menülerinizi oluşturabilirsiniz.



## Özet (devamı)

- ContextMenuStrip kontrolü, hazırladığımız uygulamanın çalışma anında form üzerinde bulunan herhangi bir kontrol üzerinde farenin sağ tuşu ile açılan bir menü hazırlamamızı sağlamaktadır. Böylelikle kontrol üzerinde sağ tuş ile açılan menüde hazırlanan komutlardan herhangi birini kullanarak kullanıcıya yaptırmak istediğimiz seçenekleri belirleyebiliriz.
- ToolStrip kontrolünün sağ üst köşesindeki ok'a tıkladığınız zaman açılan menüden Standart Öğe Ekle seçeneğini kullanmanız gerekmektedir. Bu işlemin ardından Yeni, Aç, Kaydet, Yazdır, Kes, Kopyala, Yapıştır ve Yardım olmak üzere 8 adet kullanıma hazır buton otomatik olarak eklenecektir. Bu seçeneklerden istediğinizi Delete tuşunu kullanarak silebilir, sürükleyip bırak yöntemi ile yerini değiştirebilir veya ToolStrip kontrolünde bulunan düğme ile yeni öğeler ekleyerek düzenleme işleminizi gerçekleştirebilirsiniz.
- Uygulamanın çalışma anında kullanıcıya bilgi vermek amacıyla StatusStrip kontrolü kullanılabilir. StatusStrip kontrolü üzerinde anlık olarak ilerleyen durumun bilgisi, StatusLabel kontrolü eklenerek kullanıcıya gösterilebilmektedir. Ayrıca çalışma daha önceden durum çubuğunda gösterimi sağlanan seçeneklerin bilgisi de DropDownButton kontrolü kullanılarak listelenebilmektedir.

## DEĞERLENDİRME SORULARI

1. ListView kontrolüne elemanları yerleştirmek için özellikler panelinde bulunan hangi özellik kullanılmalıdır?
  - a) Name
  - b) Text
  - c) Items
  - d) Location
  - e) Size
  
2. TreeView kontrolünün içine değer girebilmek için özellikler panelinde bulunan hangi özellik kullanılmalıdır?
  - a) Text
  - b) Image
  - c) Size
  - d) CheckBoxes
  - e) Nodes
  
3. Aşağıdakilerden hangi form üzerine sekmeler eklemeye yarayan kontroldür?
  - a) TabControl
  - b) NotifyIcon
  - c) ListView
  - d) TreeView
  - e) MenuStrip
  
4. Uygulamanın çalışma anında görev çubuğuna yerleşen simgesini değiştirmek için hangi kontrol kullanılmalıdır?
  - a) TabControl
  - b) NotifyIcon
  - c) ListView
  - d) TreeView
  - e) ToolStrip
  
5. Aşağıdakilerden hangisi BalloonTipIcon özelliğinin seçenekleri arasında yer almaz?
  - a) None
  - b) Info
  - c) Warning
  - d) Abort
  - e) Error

6. NotifyIconAdı.ShowBalloonTip(2000);  
Yukarda verilen kod satırında bulunan parametre hangi işe yaramaktadır?
- a) Bildirimin 2000 piksel boyutunda olmasını sağlar.
  - b) Bildirim alanında 2000 yazısının görünmesini sağlar.
  - c) Bildirim alanının 2000 saniye ekranda kalmasını sağlar.
  - d) Bildirim alanının 2000 milisaniye ekranda kalmasını sağlar.
  - e) Bildirim alanına 2000 karakter girilmesine izin verir.
7. Öğeler koleksiyonu düzenleyicisi penceresiyle menülerin içine tanımlayacağımız komutları özellikler panelinde bulunan hangi özellik ile düzenleyebiliriz?
- a) Text
  - b) Items
  - c) DropDownItems
  - d) Seperator
  - e) Show
8. Aşağıdakilerden hangisi menülerde bulunan komutlara kısayol tuşu eklememizi sağlayan özelliktir?
- a) ShowShortcutKeys
  - b) Items
  - c) Nodes
  - d) Name
  - e) DropDownItems
9. Farenin sağ tuşu kullanılarak açılan bir menü oluşturmak için hangi kontrolü kullanmamız gerekir?
- a) MenuStrip
  - b) ContextMenuStrip
  - c) ToolStrip
  - d) StatusStrip
  - e) ListView
10. Aşağıdakilerden hangisi form üzerinde araç çubuğu oluşturmak için kullanılan kontroldür?
- a) MenuStrip
  - b) ContextMenuStrip
  - c) ToolStrip
  - d) StatusStrip
  - e) ListView

**Cevap Anahtarı**

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

## 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.