

# MENÜ TASARIMI VE YAZDIRMA İŞLEMLERİ



## İÇİNDEKİLER

- Görsel Programlama Menü Tasarımı ve Yazdırma İşlemleri
  - Menü Tasarımı
  - Yazdırma İşlemleri



## HEDEFLER

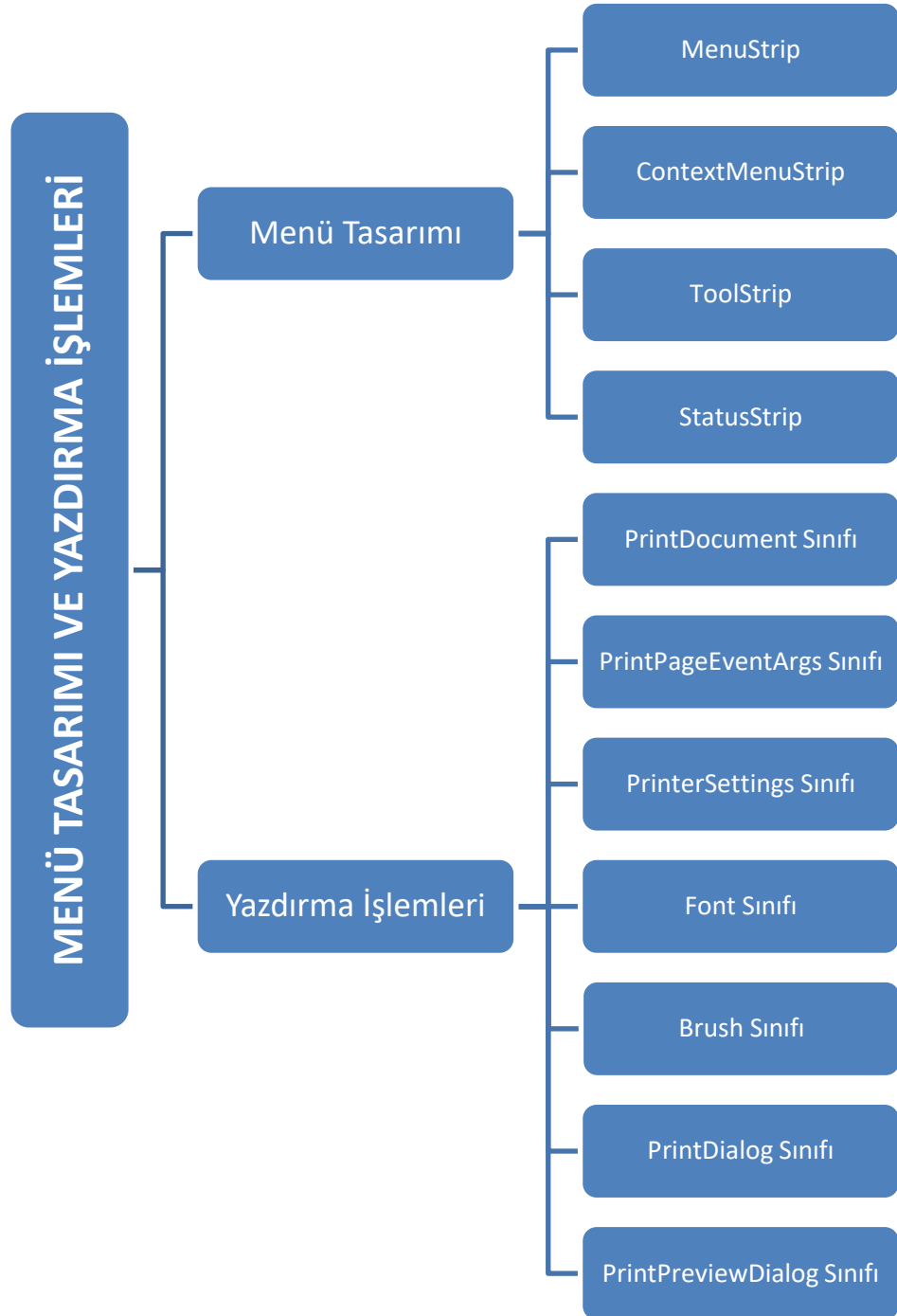
- Bu üniteyi çalıştıktan sonra;
- İstenilen özellikte menüler tasarlayabilecek,
- Hazır menüleri uygulamanızda kullanabilecek,
- Uygulamalarınız üzerinde baskı ön izleme, yazdırma işlemleri yapabileceksiniz.



**Atatürk Üniversitesi**  
Açıköğretim Fakültesi

## GÖRSEL PROGRAMLAMA I İshak Metehan SİS

## ÜNİTE 13



## GİRİŞ

Görsel Programlamada yer alan nesneler ile form üzerinde çeşitli özelliklere sahip menüler oluşturulabilir. Bunun için “MenuStrip”, “ContextMenuStrip”, “ToolStrip” ve “StatusStrip” olmak üzere 4 farklı nesne kullanılmaktadır. Farklı nesneler kullanılarak farklı özelliklere sahip menüler oluşturulabilir. Bu nesnelerin ortak noktası ise hepsi *Strip* yani *Şerit* nesneleridir ve menü üzerinde şerit görünümü verirler.

C# uygulamalarında sıkça kullanılan “MenuStrip” genelde üst kısımda yer alan ana menü ve onun alt menülerinden oluşan bir yapıdır. “StatusStrip” ise MenuStrip nesnesinin aksine formun alt kısmında yer alır ve genelde kullanıcıya bilgi vermek için kullanılır. “ToolStrip” nesnesi MenuStrip’e benzer bir yapıdadır. Ancak burada MenuStrip’ten farklı olarak label dışında *ToolStripButton*, *ToolStripComboBox* gibi farklı nesneler de menü üzerine eklenebilmektedir. “ContextMenuStrip”, kullanıcı sağa tıklandığında görünen menü olarak tanımlanabilir.

C# uygulamalarında form, menü tasarımı gibi işlemler üzerinde çalışılmış ise genelde bu uygulamalarda yazıcıdan çıktı alma bölümü de yer alır. Çünkü kullanıcı girdiği verilerin kayıtlarını, yaptığı işlemlerin sonuçlarını yazdırmak isteyebilir. C#’ta yazdırma işlemi için “PrintDocument”, “PrintDialog”, “PrintPreviewDialog” gibi farklı sınıflar kullanılmaktadır (Algan,2003).

Bu ünite de ilk olarak C# menü tasarımı için kullanılan 4 menü nesnesi hakkında bilgi verilip konu örnekler ile detaylandırılacaktır. Daha sonra ise C# uygulamalarında yazdırma işlemlerinde kullanılabilecek sınıflardan, bu sınıflara ait özelliklerden bahsedilip baskı ön izleme ve yazdırma işlemleri için örnek uygulamalar yapılacaktır.



MenuStrip hem form üzerine hem de Visual Studio IDE penceresine eklenir.

## MENÜ TASARIMI

C# dilinde menü tasarımı için için MenuStrip, ContextMenuStrip, ToolStrip ve StatusStrip olmak üzere 4 farklı kontrol kullanılmaktadır.

### MenuStrip

MenuStrip birçok uygulamada kullanılan genelde üst kısımda yer alan ana menü ve altında ki alt menülerden oluşan bir yapıdır. Çoğu programda görülen üst menüler MenuStrip nesnesi ile yapılmıştır. Bir Notepad sayfası açıldığında üst tarafta görülen Dosya, Düzen, Görünüm gibi seçenekler MenuStrip’e örnek olarak verilebilir.

MenuStrip windows formuna *toolboxtan* sürükleyip bırak yöntemi ya da kod ile menuStrip oluşturarak eklenebilir. Bu bileşen üzerinde istenilen özellikler eklenerek özelleştirilmiş menüler oluşturabilirken, erişim tuşları, kısa yol tuşları, onay işaretleri, resimler ve ayırıcı çubuklar ekleyerek menülerin kullanılabilirliği ve okunabilirliğini iyileştirebilir (Dhamodaran,2020).

## MenüStrip özellikleri

Bu bölümde MenuScript nesnesinin başlıca özelliklerine yer verilmiştir (Aktaş,2018).

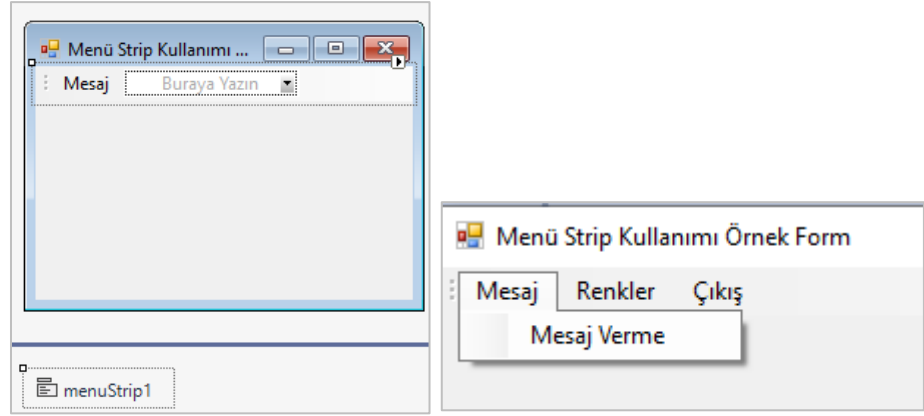


Horizontal ile yazı yatay, Vertical90 ile dikey 90 derece, Vertical270 ile yatay 270 derecelik açı ile görünür.

- **Checked:** Varsayılan değeri *False'dir*. Bu değer True olarak değiştirildiğinde menü öğesinin sol tarafına check yani tik, onay işareti eklenir.
- **CheckOnClick:** Varsayılan olarak *False* değeri seçilidir. True değeri seçilerek menü öğesine tıklandığında öğenin seçilmesi; seçili ise seçimin kaldırılması sağlanır.
- **Enabled:** Varsayılan değeri *True'dur*. Bu değer False olarak değiştirilirse ilgili menü öğesi pasif yapılarak kullanılmaz duruma getirilir. Bu özellik genelde menüde bazı alanların bazı kullanıcılara aktif olarak görünmesi gereken durumlarda kullanılır.
- **ShortcutKeys:** Seçili menü elemanına kısa yol tuşu atar. Kısa yol tuşları için harf, rakam, fonksiyon tuşları veya kontrol tuşları ile birlikte tuş kombinasyonları kullanılabilir.
- **ShowShortcutKeys:** Varsayılan değeri *True'dur*. Menüdeki öğeye atanan kısa yol tuşlarının kullanılabilmesine olanak sağlar. True değeri seçili iken menüdeki öğeye atanan kısa yol tuşları kullanılabilir. Bu değer False yapıldığında menü öğesi ile birlikte kısa yol tuşu kullanılmaz duruma gelir.
- **ShortCutKeyDisplayString:** Kısa yol tuşu atandıktan sonra bu tuş ile birlikte açıklayıcı bilgide gösterilmek isteniyorsa bu özellik kullanılır.
- **ToolTipText:** Menüdeki elemanlara ait açıklamalar bu özellik yardımı ile eklenir. Mouse ile ilgili öğenin üzerine gelindiğinde öğeye ait açıklama görünür.
- **DisplayStyle:** Menüde yer alan elemanların gösterim şeklini belirler. None, Text, Image, ImageAndText gibi değerler alır. None ile menü öğesi boş olarak görünmektedir. Text, menü elemanında sadece yazı görünmesini sağlarken Image, sadece resim görünmesini sağlamaktadır. Hem yazı hem de image özelliğinin birlikte görünmesi isteniyorsa ImageAndText değeri kullanılır.
- **Items:** Ana menünün altında yer alan alt menüye ait öğelere erişmeyi sağlar.
- **TextAlign:** Menü öğelerinde yer alan yazıların hizalanma biçimini belirtir.
- **TextDircion:** Menü elemanlarında yer alan yazının yönünü belirtir. Horizontal, Vertical90, Vertical270 gibi değerler alır.

MenuStrip kontrolü projede kullanılacak ise ilk olarak form üzerine araç kutusundan MenuStrip eklenir. Daha sonra menüde yer almasını istediğimiz seçenekler *Buraya Yazın* kısmına tıklanarak menü oluşturulur. Daha sonra MenuStrip özellikleri kullanılarak menü özelleştirilir.

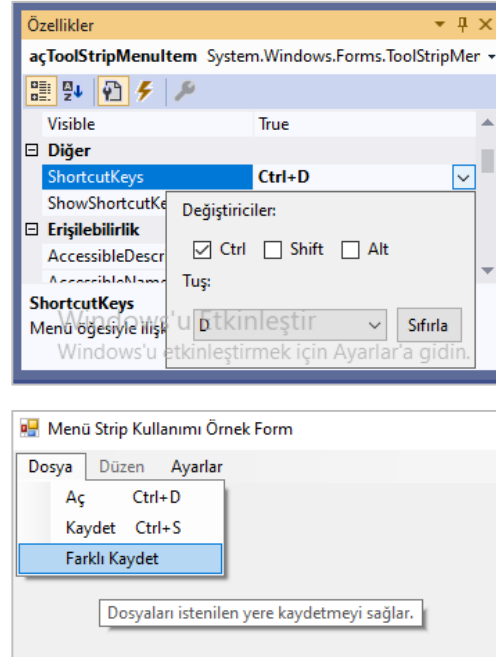
Şekil 13.1'de 3 tane seçeneği bulunan örnek bir menü oluşturulmuştur. Daha sonra örnek menü üzerinde MenüStrip kontrolüne ait birkaç özelliğin nasıl eklendiği gösterilmiştir.



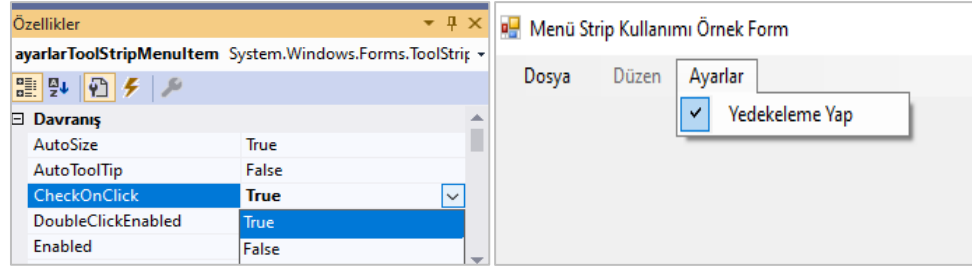
Şekil 13.1. MenuStrip Kontrolünün Form Üzerine Eklenmesi

Menü üzerinde seçeneklere özellik eklemek için seçenek tıklanır ve özellik penceresinden istenilen özellikler eklenir. Örneğin özellikler penceresinden ShowShortcutKeys değeri true yapıp ShortCutKeys kısmından değerler seçilerek kısayol eklentisi yapılabilir. Yine özellikler kısmından ToolTipText özelliği ile menüdeki seçenek üzerine gelerek o seçenek hakkında kullanıcıya bilgi verilmesi sağlanabilir.

Şekil 13.2’de ShortCutKeys, ToolTipText ve Enabled özelliklerinin kullanımı örneklendirilmiştir. “Düzen” seçeneğinin Enabled değeri False yapılarak pasifleştirilmiştir.



Şekil 13.2. Kısa Yol Tuşlarının Eklenmesi ve ToolTipText Özelliği

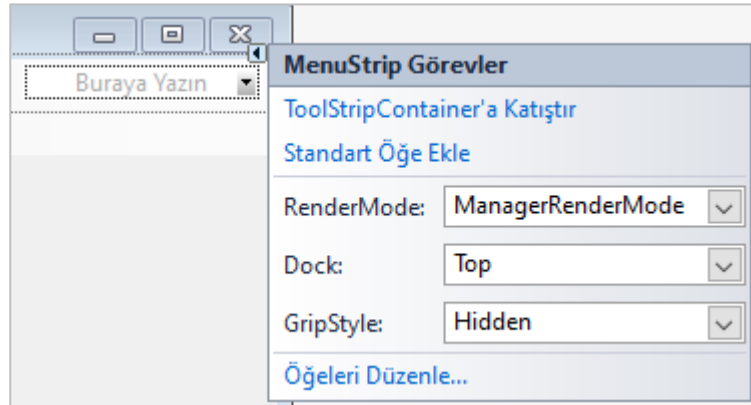


Şekil 13.3. CheckOnClick Özelliğinin True Durumu

Menü tasarımı tamamlandıktan sonra menü üzerinde yer alan seçeneklerin Kaydetme, Düzenleme gibi yapacağı işlemler ilgili seçeneğin, *Click* kısmına yazılan kodlar ile sağlanmaktadır. İlgili seçeneğe çift tıklanarak Click kısmına ulaşılabilir.

Çoğu uygulamada Dosya, Düzen, Araçlar gibi seçenekler menülerde ortak olarak kullanılmaktadır. Bu tip ortak menüler MenüStrip kontrolünün sağ üst köşesinde yer alan kulakçıya tıklanıp *Standart Öğe Ekle* seçeneği seçilerek kolayca oluşturulabilir. MenüStripTask menüsü ekleme penceresinde RenderMode, Dock ve GripStyle olmak üzere üç farklı seçenek bulunmaktadır.

- **RenderMode:** System, Professional, ManagerRenderMode değerlerinden birini alarak menü çubuğunun görünümünü ayarlar. System seçeneği ile menü çubuğu varsayılan Windows ayarları ile görüntülenirken Professional seçeneği ile Office uygulamaları menü görünümü şeklinde görünür. Özel menü tasarımları için ManagerRenderMode seçeneği kullanılır.
- **Dock:** Menünün form üzerinde ki konumunu ayarlar.
- **GripStyle:** Menü çubuğunu taşımak için kullanılan taşıyıcı işaretlerinin gösterilip gösterilmeyeceğini ayarlar.



Şekil 13.4. MenuStripTask



MenuStripTask kontrolü ile hazır menüler forma basitçe eklenebilir.



Şekil 13.5. MenuStripTask ile Oluşturulmuş Hazır Menü

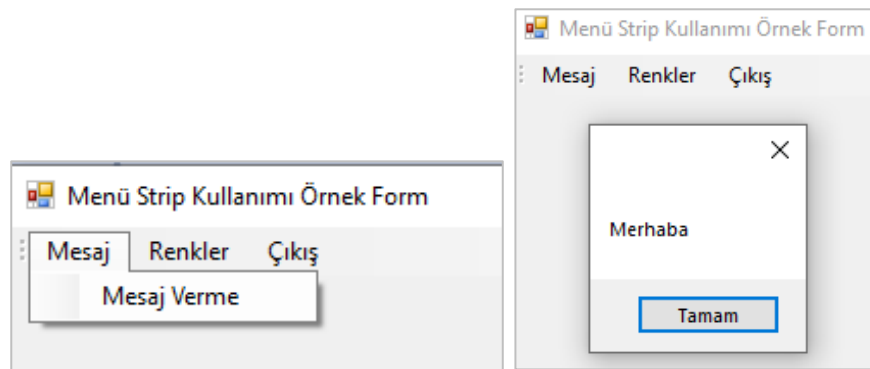
MenuStrip ile menü seçenekleri oluşturulduktan sonra seçeneğin “Click” kısmına yapacağı işlemin kodları yazılır. Aşağıda örnek bir uygulama gösterilmiştir.

```
private void mesajVermeToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Merhaba");
}

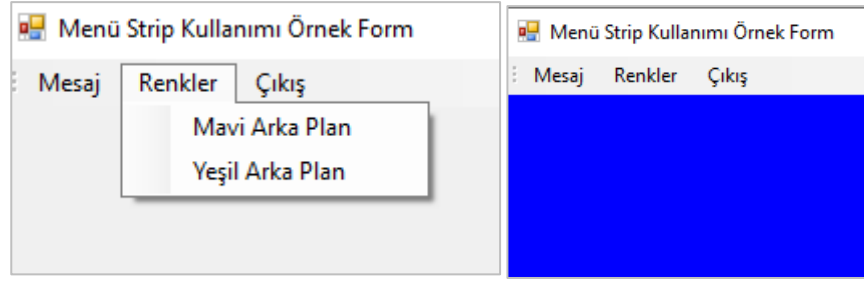
private void maviArkaPlanToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}

private void yeşilArkaPlanToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Green;
}

private void çıkışToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```



Şekil 13.6. Mesaj Verme Seçeneğine Tıklanma Durumu



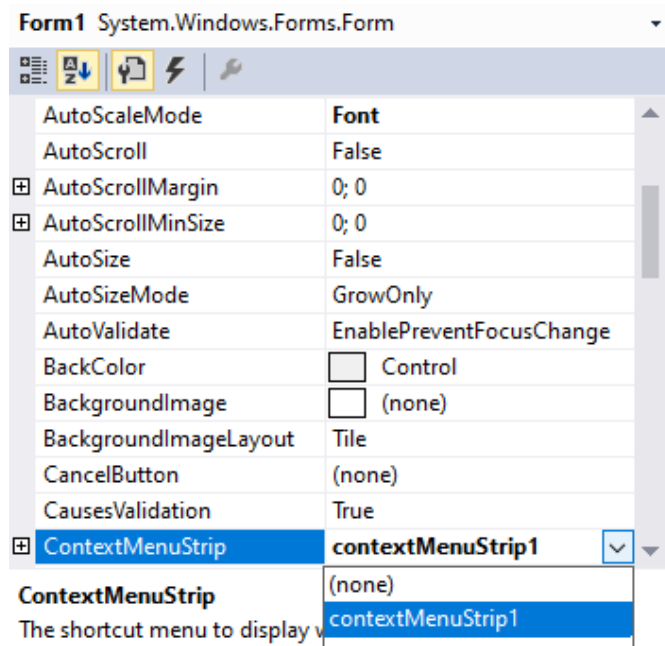
Şekil 13.7. Renkler Seçeneğinden Mavi Arka Plan Tıklanma Durumu

## ContextMenuStrip



ContextMenuStrip özellikler kısmında seçilip bağlantı yapılmazsa uygulama çalıştırıldığında menü görünmez.

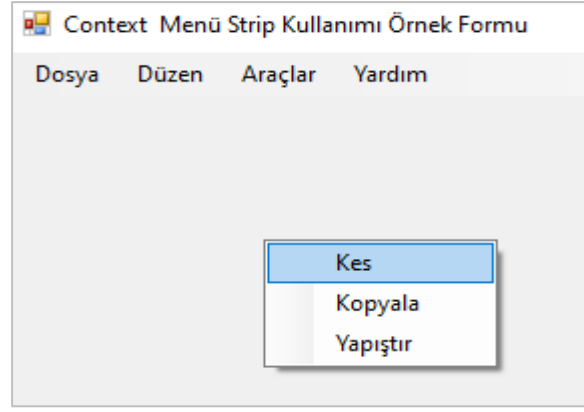
Herhangi bir kontrole farenin sağ tuşu ile tıklandığında açılan menüleri oluşturmak için ContextMenuStrip kontrolü kullanılır. Böyle bir menü oluşturmak için ilk olarak araç kutusundan ContextMenuStrip seçilerek form üzerine eklenir. Eklenen kontrol hem formun üstüne hem de Visual Studio IDE'nin alt tarafına konumlanır. Daha sonra MenuStrip'te olduğu gibi *Buraya Yazın* kısmına, kullanıcıya gösterilmek istenen seçenekler yazılır. Uygulama çalıştırılıp sağ tuşa tıklanınca menünün açılması için oluşturulan ContextMenuStrip kontrolünün bağlantısının yapılması gerekir. Örneğin form üzerinde herhangi bir yere sağ tıklayınca menünün açılması isteniyorsa bu formun özellikler kısmında yer alan ContextMenuStrip alanından oluşturulan ContextMenuStrip seçilerek bağlantı sağlanmalıdır. Şekil 13.8'de ContextMenuStrip kontrolünün forma nasıl bağlandığı gösterilmiştir.



Şekil 13.8. ContextMenuStrip Bağlantısı

Bağlantı işlemi yapıldıktan sonra program çalıştırılıp form üzerine sağ tıklandığında oluşturulan menü açılır.





Şekil 13.9. Sağ Tıklanınca Açılan Menü

ContextMenuStrip kontrolünün birçok özelliği MenüStrip nesnesi ile aynı olduğu için bu kısımda tekrar değinilmemiştir.



Bireysel Etkinlik

- ContextMenuStrip kontrolünün özelliklerini inceleyerek bu özelliklerin yer aldığı bir menü tasarlayınız..

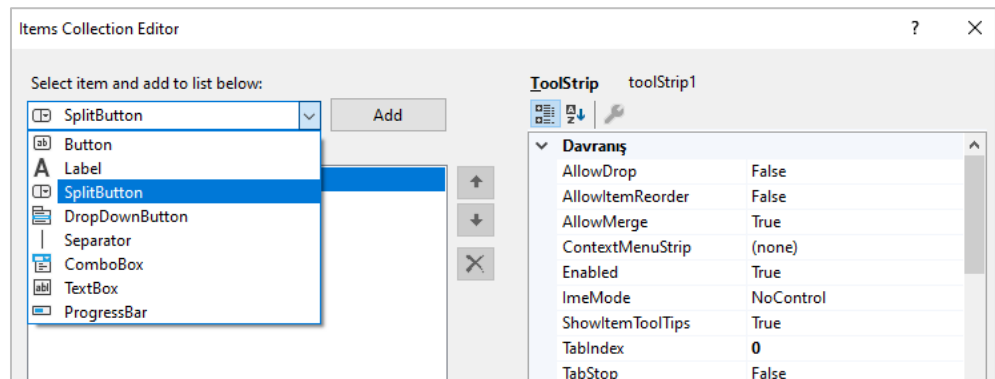
## ToolStrip



ToolStrip üzerine Button, Comobox, DropDownButton, Label, ProgressBar Separator, TextBox ve SplitButton eklenebilir.

ToolStrip, genelde uygulamalarda üst kısımda yer alan araç çubuğunu oluşturmak için kullanılan bir kontroldür. ToolStrip kontrolü uygulamaya eklendiğinde hem formun üstüne hem de Visual Studio IDE'nin alt tarafına konumlanır.

ToolStrip form üzerine eklendikten sonra **Özellikler** penceresinin Items kısmından araç çubuğunda olması istenilen öğeler seçilerek ekleme yapılır. MenüStrip kontrolünden farklı olarak burada ToolStripButton, ToolStripLabel, ToolStripSplitButton, ToolStripDropDownButton, ToolStripSeperator, ToolStripComboBox, ToolStripTextBox, ToolStripProgressBar gibi öğeler de eklenebilmektedir (Bükülmez, 2020).



Şekil 13.10. Items Menüsünden Öğe Eklenmesi



Standart Öğe Ekle seçeneği kullanılarak hazır araç çubukları uygulamalara eklenebilir.

- **ToolStripButton:** Üzerine tıklandığında bir işlem yapmayı sağlayan yapıdır.
- **ToolStripLabel:** Kullanıcıya bilgi vermek için kullanılır.
- **ToolStripSplitButton:** Normal button nesnesinden farklı olarak aynı zamanda açılır menü olarak da kullanılır. Nesnenin ayanındaki *aşağı ok* simgesine tıklandığında menü gibi davranır ve menü seçenekleri görünür.
- **ToolStripSeparator:** ToolStrip'e eklenen araçların arasına ayraç ekler.
- **ToolStripComboBox:** ComboBox, kullanıcının bilgi girişi yapabilmesini sağlamak ve verileri görüntülemek için kullanılan bir nesnedir. İki kısımdan oluşur. Üst tarafı bir textbox yapısındadır ve kullanıcının metin girmesi sağlanır. Alt tarafı ListBox gibidir. Kullanıcının seçebileceği öğeler burada listelenir.
- **ToolStripDropDownButton:** DropDown kullanıcıya açılır bir listeden eleman seçimi yaptırmak için kullanılır. ToolStripDropDown ile ise tıpkı MenüStrip'de olduğu gibi menüler eklenebilir.
- **ToolStripTextBox:** Kullanıcının form üzerinden veri girmesini sağlar.
- **ToolStripProgressBar:** C#'ta progressbar herhangi bir işlemin ilerleme durumunu göstermek için kullanılır.

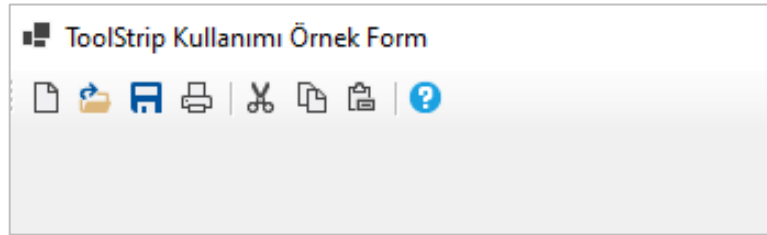
ToolStrip üzerine eklenen öğeler üzerinde değişiklikler yapılarak istenen yapıda araç çubuğu oluşturulabilir. Items kısmından değiştirilmek istenen öğe seçilir ve Özellikler penceresinden istenilen özellik değiştirilir. Tablo 13.1'de yaygın olarak kullanılan özellikler yer almaktadır.

**Tablo 13.1.** ToolStrip Menü için Kullanılan Özellikler

	Açıklama
<b>DisplayStyle</b>	Araç çubuğu üzerinde yer alan öğenin nasıl görüneceğini belirler. Öğe üzerinde resim, yazı ya da her ikisi birlikte gösterileceği gibi hiçbir şey gösterilmeyebilir.
<b>ImageScalling</b>	Öğe üzerine konulacak resmin boyutunu ayarlar.
<b>BackColor</b>	Öğenin arka plan rengini ayarlar.
<b>TextDirection</b>	Öğe üzerinde yer alan yazının yönü belirler.
<b>Visible</b>	Öğenin araç çubuğunda görünüp görünmemesini ayarlar.
<b>ToolTipText</b>	Araç çubuğundaki öğenin üzerine fare ile gelince bilgilendirme mesajının görünmesini sağlar.
<b>Font</b>	Araç çubuğu üzerinde yer alan öğenin metin boyutunu, stilini ayarlar.
<b>Image</b>	Araç çubuğunda bulunan öğelere resim eklemeyi sağlar.

ToolStrip kontrolünün sağ üst köşesinde yer alan kulakçığa tıklanıp *Standart Öğe Ekle* seçeneği tıklanarak hazır araç çubuğu eklenebilir. Bu araç çubuğunda Yeni, Aç, Kaydet, Yazdır, Kes, Kopyala, Yapıştır ve Yardım seçenekleri yer almaktadır. Bu seçeneklerin stilleri, yerleri değiştirilebilir ya da araç çubuğundan

silinebilir. Bu işlem için de yine Özellikler penceresinde yer alan *Items* özelliği kullanılır.



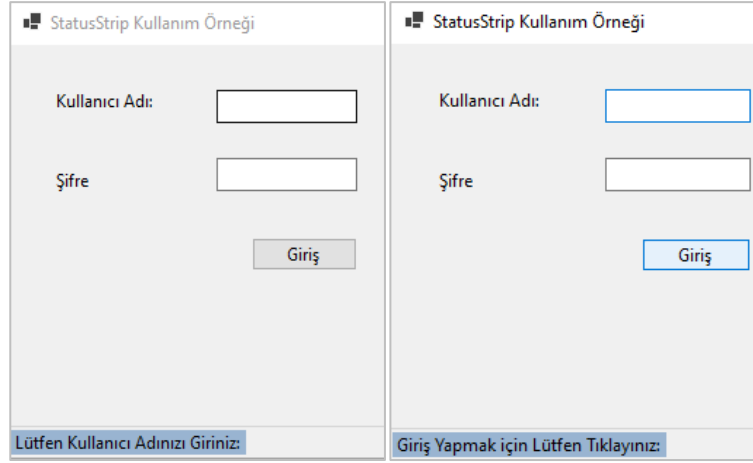
Şekil 13.11. Hazır Araç Çubuğu

## StatusStrip

Genelde kullanıcıya bilgi vermek için kullanıldığından *Bilgi Şeridi* olarak da adlandırılır. Formun en altına konumlanır. Kullanıcıya anlık tarih ve saat bilgisi, aktif kullanıcı bilgisi gibi bilgileri göstermek için kullanılır. StatusStripLabel, StatusStripProgressBar, StatusStripDropDownButton ve StatusSplitButton gibi 4 farklı ögesi bulunmaktadır. Bu nesnelerin detaylarına ToolStrip bölümünde değinilmiştir.

Aşağıdaki örnek uygulamada form üzerinde yer alan öğelerin üzerine fare ile gelindiğinde yapılması gereken işlemler StatusStripLabel kullanılarak kullanıcıya bilgilendirme mesajı olarak gösterilmiştir.

```
private void textBox1_MouseHover(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "Lütfen Kullanıcı Adınızı Giriniz:";
}
private void textBox1_MouseLeave(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "";
}
private void textBox2_MouseHover(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "Lütfen Şifrenizi Giriniz:";
}
private void textBox2_MouseLeave(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = ""
}
private void button1_MouseHover(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "Giriş Yapmak için Lütfen Tıklayınız:";
}
private void button1_MouseLeave(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "";
}
```



Şekil 13.12. Fare İlgili Alanlara Getirilince

## YAZDIRMA İŞLEMLERİ



PrintDocument sınıfı ile tanımlama yapıldığında yazdırma işlemi default yazıcı ile yapılmaktadır.

C# ile geliştirilen uygulamaların çoğunda yazıcıdan çıktı alma işlemi de yapılır. Böylece kullanıcı gerek duyduğunda uygulama içerisinde yazdırma işlemi yapabilir. Yazdırma işlemi sırasında yazı boyutunu, rengini, konumunu ayarlamak için “Font”, “SolidBrush”, “PrintDocument” gibi farklı sınıflar kullanılmaktadır. Bu bölümde yazdırma işlemi sırasında kullanılacak sınıflar, bu sınıflara ait metod ve özellikler hakkında bilgi verilecektir.

### PrintDocument Sınıfı

PrintDocument sınıfı yazdırma işlemleri için kullanılan bir sınıftır. PrintDocument kullanılabilmesi için programa *System.Drawing.Printing* namespace eklenmelidir.

- PrintDocument yazıcıCiktisi = new PrintDocument();  
şeklinde nesne tanımlanır. Yazdırma işlemi için “Print” metodu kullanılır.

### PrintPageEventArgs Sınıfı

Yazdırma işlemleri sırasında yazdırılacak dokümanın özelliklerinin belirlenmesi, yazdırmanın nasıl yapılacağı gibi işlemler için PrintPageEventArgs sınıfına ait özellikler kullanılmaktadır. Tablo 13.2’de bu sınıfa ait ve yaygın olarak kullanılan özellikler ve açıklamaları yer almaktadır.

Tablo 13.2. PrintPageEventArgs Sınıfı Özellikleri

	Açıklama
<b>Graphics</b>	Yazdırılacak dokümanın bilgilerini belirleyen grafik nesnesidir.
<b>HasMorePages</b>	Birden fazla sayfanın yazdırılacağı durumda bu özelliğin değeri <i>true</i> olarak değiştirilmelidir.
<b>PageBounds</b>	Yazdırılacak dokümanın en ve boy bilgisini belirtir.
<b>PageSettings</b>	Yazdırılacak dokümanın rengi, konumu, kağıt büyüklüğü gibi sayfa ayarlarını yapar.

<b>Cancel</b>	Çıktı alma işleminin iptali ile ilgili bir nesnedir. Bool değer döndürür. Eğer değer true ise işlem iptal edilir.
---------------	---

## PrinterSettings Sınıfı

Yazdırılacak dokümanın nasıl yazdırılacağı hakkında bilgi almak için PrinterSettings sınıfı kullanılmaktadır. Bu sınıfta yer alan ve yaygın kullanılan özellikler Tablo 13.3'te gösterilmiştir.

**Tablo 13.3.** PrinterSettings Sınıfı Özellikleri (Algan,2003)

	<b>Açıklama</b>
<b>Collate</b>	Yazdırılan dokümanın harmanlanmış olup olmadığını gösterir.
<b>Copy</b>	Yazdırılacak dokümanın kopya sayısını belirler.
<b>DefaultPageSettings</b>	Varsayılan sayfa ayarlarını gösterir.
<b>Duplex</b>	Arkalı önlü baskı özelliğini belirler.
<b>PrinterName</b>	Kullanılacak yazıcının adını belirtir.
<b>IsDefaultPrinter</b>	"PrinterName" ile belirtilen yazıcının, bilgisayardaki varsayılan yazıcı olup olmadığını gösterir.
<b>IsValid</b>	"PrinterName" ile belirtilenin yazıcının sistemde geçerli bir yazıcı olup olmadığını gösterir.
<b>PaperSizes</b>	Yazıcı tarafından desteklenen sayfa boyutlarını gösterir.
<b>FromPage</b>	Yazdırılacak ilk sayfanın sayfa numarasını belirtir.
<b>CanDuplex</b>	Bool tipinde değer döndürür. Yazıcının çift taraflı yazdırmayı destekleyip desteklemediğini belirtir.

## Font Sınıfı

Yazdırılacak metnin stilini, font tipini, boyutunu ayarlamak için kullanılır. Örneğin yazdırılacak metnin Arial tipinde 12 punto boyutunda olmasını istiyorsak aşağıdaki gibi tanımlama yapılması gerekir.

- Font yaziTipi= new Font ("Arial", 12);

**Tablo 13.4.** Font Sınıfı Özellikleri

	<b>Açıklama</b>
<b>Bold</b>	Font nesnesinin, koyu yazma özelliğine sahip olup olmadığını gösterir.
<b>Italic</b>	Font nesnesinin, italik özelliğine sahip olup olmadığını gösterir.
<b>Height</b>	Yazı tipinin satır aralığını belirtir.
<b>Style</b>	Font nesnesinin stil bilgisini gösterir.
<b>Underline</b>	Font nesnesinin alt çizili yazma özelliğinin olup olmadığını belirtir.
<b>Size</b>	Nesnenin font büyüklüğünü belirtir.



Brush sınıfı abstrac bir sınıf olduğu için SolidBrush, TextureBrush, LinearGradientBrush gibi Brush sınıfından türetilen sınıflar kullanılır.

## Brush Sınıfı

Fırça oluşturup şekillerin içini doldurmak için kullanılan abstract bir sınıftır. Örneğin siyah renk kullanılarak doküman yazdırılmak isteniyorsa aşağıdaki gibi bir nesne oluşturulur.

- SolidBurush firca= new SolidBrush(Clr.Black);

## PrintDialog Sınıfı

Kullanıcılar herhangi bir belgeyi yazdırırken yazdırma özelliklerinin yer aldığı bir pencere açılır. Bu pencerede yazıcı ismi, yazdırılacak sayfa numaraları, kopya sayısı gibi ayarlar yapılabilir. C#’ta da yazdırma işlemleri sırasında “PrintDialog” sınıfı ile bu pencereye benzer bir pencere oluşturulup gerekli ayarlamalar yapılabilir.

Uygulamada PrintDialog sınıfı kullanacak ise ilk önce ToolBox’tan forma PrintDialog kontrolü sürükleyip bırak yöntemi ile eklenmelidir. Daha sonra kod içerisinde nesnesi oluşturulur.

- PrintDialog pd= new PrintDialog();
- pd.ShowDialog();



PrintDialog kontrolü form üzerine bırakıldığında bu kontrole dair herhangi bir simge görünmez.

**Tablo 13.5.** PrintDialog Sınıfı Özellikleri (Gürsoy, 2012).

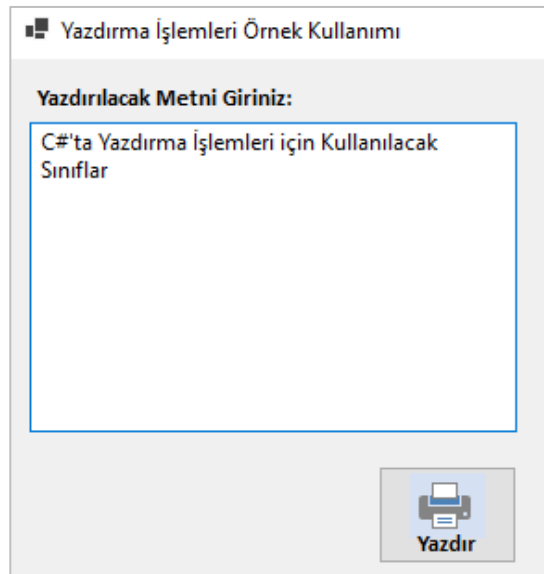
	Açıklama
<b>AllowSelection</b>	Belli bir seçimin yapılup yapılmayacağına izin verir.
<b>ShowHelp</b>	Yardım durumunu gösterir.
<b>ShowNetwork</b>	Network durumunu gösterir.
<b>PrintToFile</b>	Herhangi bir dosyanın yazdırılıp yazdırılmayacağını belirtir.
<b>AllowSomePages</b>	Belirli sayfaların yazdırılıp yazdırılmayacağına izin verir.

Yukarıda anlatılan sınıflar kullanılarak textbox içinde ki verileri yazdıran örnek;

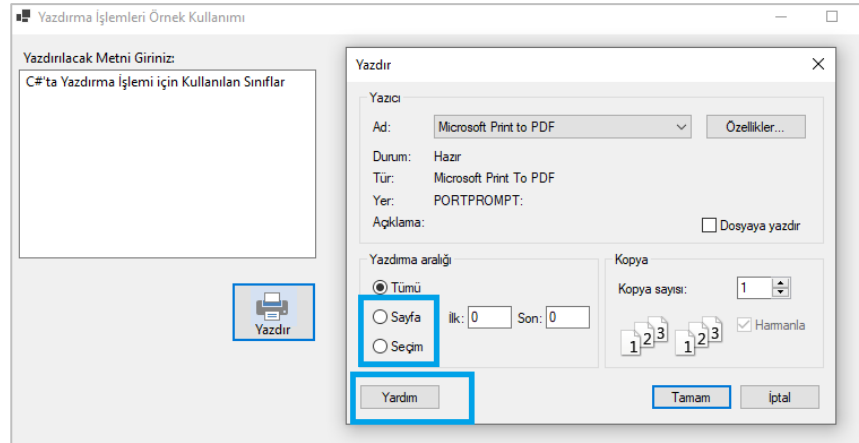
```

PrintDialog pd = new PrintDialog();
PrinterSettings ps = new PrinterSettings();
private void buttonYazdir_Click(object sender, EventArgs e)
{
    PrintDocument kagit = new PrintDocument();
    DialogResult yazmalslemi;
    //Seçim yapabilmek için
    pd.AllowSelection = true;
    //Sayfa seçimi için
    pd.AllowSomePages = true;
    //Yardım seçeneği için
    pd.ShowHelp = false;
    //Yazma penceresi için
    yazmalslemi = pd.ShowDialog();
    kagit.PrintPage += new PrintPageEventHandler(kagit_PrintPage);
    kagit.Print();
}
private void kagit_PrintPage(object sender, PrintPageEventArgs e)
{
    //Yazı rengi
    SolidBrush fırça = new SolidBrush(Color.Black);
    //Yazının stili
    Font print_font = new Font("Times New Roman", 12, FontStyle.Bold);
    //textBox'a girilen yazının belirlenen özellikler ile yazdırılması
    e.Graphics.DrawString(textBox1.Text, print_font, fırça, 200, 25);
}

```



Şekil 13.13. TextBox'a Yazdırılacak Veri Girişi



Şekil 13.14. Belirtilen Özelliklerdeki Yazdırma Penceresi

## PrintPreviewDialog Sınıfı

Yazdırılacak olan dokümanın ön izlemesi "PrintPreviewDialog" kontrolü ile elde edilebilir.

Baskı ön izleme işlemi için ilk olarak araç kutusundan PrintPreviewDialog ve PrintDocument seçilerek form üzerine eklenir ve birbirleriyle iletişim kurmaları sağlanır. Böylece PrintDocument'de yer alan veriler PrintPreviewDialog kısmına aktarılmış olur. Aşağıda örnek kullanım gösterilmiştir.



Örnek

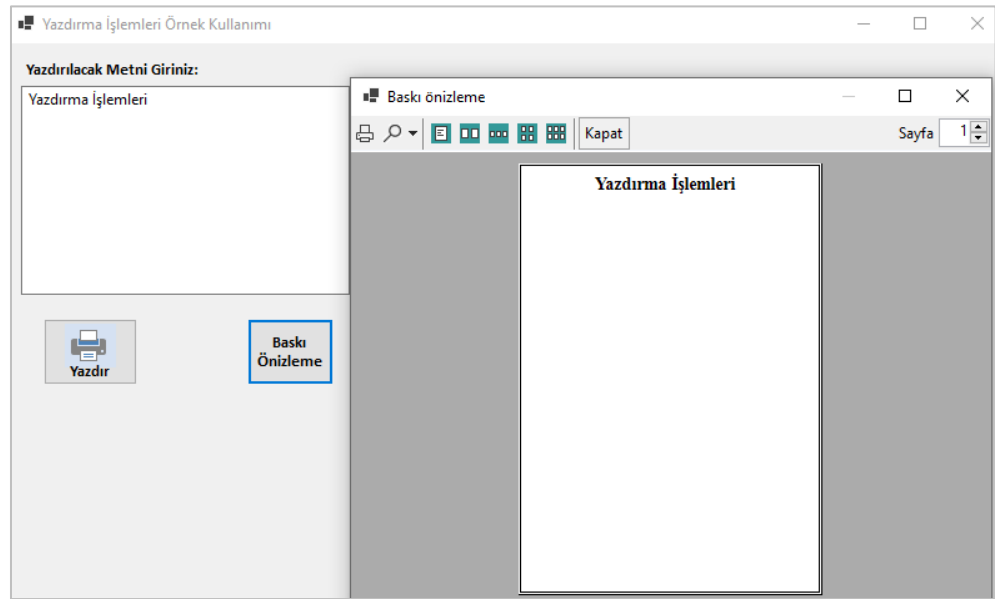
```
private void baskiOnizleme_Click(object sender, EventArgs e)
{
    printPreviewDialog1.Document = printDocument1;
    printPreviewDialog1.ShowDialog();
}
```

TextBox içine girilen veriyi yazdırmadan önce baskı ön izleme yaparak gösteren uygulama;

```
private void button1_Click(object sender, EventArgs e)
{
    printPreviewDialog1.Document = printDocument1;
    printPreviewDialog1.ShowDialog();
}
private void printDocument1_PrintPage(object sender, PrintPageEventArgs e)
{
    //yazı rengi
    SolidBrush fırça = new SolidBrush(Color.Black);
    //Yazının stili
    Font print_font = new Font("Times New Roman", 36,
    FontStyle.Bold);
    //textBox'a girilen yazının belirlenen özellikler ile yazdırılması
    e.Graphics.DrawString(textBox1.Text, print_font, fırça, 200, 25);
}
```



Şekil 13.15. TextBox'a Yazdırılacak Veri Girişi



Şekil 13.16. Baskı Ön İzleme



#### Bireysel Etkinlik

- Bir mağazada barkod numarasına göre aranan ürünün bilgilerini (Ürün Adı, Barkod No, Fiyatı, Miktarı gibi) yazdıran, yazdırma işleminden önce baskı ön izleme yaptıran bir uygulama yapınız. Ürün bilgilerini datagridview içinde listeleyebilirsiniz.



## Özet

### •MENÜ TASARIMI

•Görsel Programlamada yer alan nesneler ile form üzerinde çeşitli özelliklere sahip menüler oluşturulabilir. Bunun için "MenuStrip", "ContextMenuStrip", "ToolStrip" ve "StatusStrip" olmak üzere 4 farklı nesne kullanılmaktadır.

### •MenuStrip

•MenuStrip birçok uygulamada kullanılan genelde üst kısımda yer alan ana menü ve altında ki alt menülerden oluşan bir yapıdır. MenuStrip windows formuna *toolboxtan* sürükleyip bırak yöntemi ya da kod ile menuStrip oluşturularak eklenebilir. Bu bileşen üzerinde istenilen özellikler eklenerek özelleştirilmiş menüler oluşturabilirken, erişim tuşları, kısa yol tuşları, onay işaretleri, resimler ve ayırıcı çubuklar ekleyerek menülerin kullanılabilirliği ve okunabilirliğini iyileştirebilir.

### •ContextMenuStrip

•Herhangi bir kontrole farenin sağ tuşu ile tıklandığında açılan menüler oluşturmak için ContextMenuStrip kontrolü kullanılır. Böyle bir menü oluşturmak için ilk olarak araç kutusundan ContextMenuStrip seçilerek form üzerine eklenir. Daha sonra "*Buraya Yazın*" kısmına, kullanıcıya gösterilmek istenen seçenekler yazılır. Uygulama çalıştırılıp sağ tuşa tıklanınca menünün açılması için ContextMenuStrip kontrolünün bağlantısının yapılması gerekir.

### •ToolStrip

•ToolStrip, genelde uygulamalarda üst kısımda yer alan araç çubuğunu oluşturmak için kullanılan bir kontroldür. MenuStrip kontrolünden farklı olarak burada ToolStripButton, ToolStripLabel, ToolStripSplitButton, ToolStripDropDownButton, ToolStripSeperator, ToolStripComboBox, ToolStripTextBox, ToolStripProgressBar gibi öğeler de eklenebilmektedir.

### •StatusStrip

•Genelde kullanıcıya anlık tarih ve saat bilgisi, aktif kullanıcı bilgisi gibi bilgileri göstermek için kullanılır. StatusStripLabel, StatusStripProgressBar, StatusStripDropDownButton ve StatusStripSplitButton gibi 4 farklı öğesi bulunmaktadır.

### •YAZDIRMA İŞLEMLERİ

•C# ile geliştirilen uygulamaların çoğunda yazıcıdan çıktı alma işlemi de yapılır. Böylece kullanıcı gerek duyduğunda uygulama içerisinde yazdırma işlemi yapılabilir. Yazdırma işlemi sırasında yazı boyutunu, rengini, konumunu ayarlamak için "Font", "SolidBrush", "PrintDocument" gibi farklı sınıflar kullanılmaktadır.

### •PrintDocument Sınıfı

•PrintDocument sınıfı yazdırma işlemleri için kullanılan bir sınıftır. PrintDocument kullanılabilmesi için programa *System.Drawing.Printing* namespace eklenmelidir.

### •PrintPageEventArgs Sınıfı

•Yazdırma işlemleri sırasında yazdırılacak dokümanın özelliklerinin belirlenmesi, yazdırmanın nasıl yapılacağı gibi işlemler için PrintPageEventArgs sınıfına ait özellikler kullanılmaktadır.

### •PrinterSettings Sınıfı

•Yazdırılacak dokümanın nasıl yazdırılacağı hakkında bilgi almak için PrinterSettings sınıfı kullanılmaktadır.

### •Font Sınıfı

•Yazdırılacak metnin stilini, font tipini, boyutunu ayarlamak için kullanılır.

### •Brush Sınıfı

•Fırça oluşturup şekillerin içini doldurmak için kullanılan abstract bir sınıftır.

### •PrintDialog Sınıfı

•Kullanıcılar herhangi bir belgeyi yazdırırken yazdırma özelliklerinin yer aldığı bir pencere açılır. Bu pencerede yazıcı ismi, yazdırılacak sayfa numaraları, kopya sayısı gibi ayarlar yapılabilir.

### •PrintPreviewDialog Sınıfı

•Yazdırılacak olan dokümanın ön izlemesi "PrintPreviewDialog" kontrolü ile elde edilebilir.

## DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi MenuStrip nesnesinin özelliklerinden biri değildir?
  - a) CheckOnClick
  - b) Close
  - c) ShortcutKey
  - d) Checked
  - e) DisplayStyle
2. Aşağıdakilerden hangisi ToolStripMenu ile oluşturulan araç çubuğu üzerine eklenebilecek elemanlardan biri değildir?
  - a) Button
  - b) Label
  - c) DataGridView
  - d) Separator
  - e) ComboBox
3. Kullanıcıya bilgi vermek için aşağıdaki menü kontrollerinden hangisi kullanılır?
  - a) MenuStrip
  - b) ContextMenu
  - c) ToolStrip
  - d) StatusStrip
  - e) MenuStripTask
4. PrintDocument sınıfını kullanabilmek için aşağıdaki namespace'lerden hangisinin uygulamaya eklenmiş olması gerekir?
  - a) System.IO
  - b) System.Linq
  - c) System.Text
  - d) System.Threading.Task
  - e) System.Drawing.Printing
5. Yazdırma işlemi sırasında yazdırma penceresinin açılması için aşağıdaki sınıflardan hangisi kullanılır?
  - a) PrinterSettings
  - b) Font
  - c) Brush
  - d) PrintDocument
  - e) PrintDialog

6. Aşağıdakilerden hangisi Font sınıfı özelliklerinden biri değildir?
- a) Italic
  - b) Underline
  - c) Cancel
  - d) Bold
  - e) Style
7. Aşağıdaki özelliklerden hangisi yazdırma işlemi sırasında yazdırılacak sayfa aralığını seçmeyi sağlar?
- a) ShowHelp
  - b) PrintToFile
  - c) ShowNetwork
  - d) AllowSomePages
  - e) PaperSize
8. Aşağıda verilen MenuStrip özelliklerinden hangisi menü seçeneklerine kısa yol tuşu atanmasını sağlar?
- a) TextAlign
  - b) ShortcutKeys
  - c) TextAlign
  - d) Checked
  - e) DisplayStyle
9. Aşağıdaki özelliklerden hangisi PrintPageEventArgs sınıfı içerisinde yer almaz?
- a) Graphics
  - b) HasMorePages
  - c) PageBounds
  - d) Height
  - e) Cancel
10. Aşağıda verilen bilgilerden hangisi yanlıştır?
- a) PrinterSettings sınıfı altında yer alan Copy özelliği arkalı önlü baskı özelliğini belirler.
  - b) StatusStrip kullanıcıya bilgi vermek için kullanılan bir menü kontrolüdür.
  - c) C#'ta hazır araç çubuğu ToolStrip menü yardımı ile eklenebilir.
  - d) StatusStrip kullanıcıya bilgi vermek için kullanılan bir menü kontrolüdür.
  - e) Yazdırma işlemleri sırasında kullanılan sınıflar içerisinde Brush sınıfı da yer almaktadır.

**Cevap Anahtarı**

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

## YARARLANILAN KAYNAKLAR

- Aktaş, V. (2018). Her Yönüyle C# 7.0 Kodlab Yayıncılık İstanbul.
- Algan, S. (2003). Yazdırma işlemi için kullanılan Sınıflar. 27 Ağustos 2021 tarihinde <http://www.csharpnedir.com/articles/read/?id=128> adresinden erişildi.
- Bükülmez, M. (2021). C# MenuStrip, ToolStrip, StatusStrip, ContextMenuStrip 2 Eylül tarihinde [https://mustafabukulmez.com/2020/06/23/c-menustrip-ToolStrip-statusstrip-contextmenustrip/#C\\_ToolStrip\\_Kullanimi](https://mustafabukulmez.com/2020/06/23/c-menustrip-ToolStrip-statusstrip-contextmenustrip/#C_ToolStrip_Kullanimi) adresinden erişildi.
- Dhamodaran, S. (2020). MenuStrip Kontrolü. 1 Eylül tarihinde [https://coding-examples.com/csharp/c-menustrip-contextmenustrip-controls/#1-about-this-menustrip-control-example%20\[5\]%20http://www.csharpnedir.com/articles/read/?id=128%20\[6\]%20https://docs.microsoft.com/tr-dotnet/api/system.drawing.printing.printersettings?view=net-5.0](https://coding-examples.com/csharp/c-menustrip-contextmenustrip-controls/#1-about-this-menustrip-control-example%20[5]%20http://www.csharpnedir.com/articles/read/?id=128%20[6]%20https://docs.microsoft.com/tr-dotnet/api/system.drawing.printing.printersettings?view=net-5.0) adresinden erişildi.
- Gürsoy, İ. (2012). PrintDialog ve PrintDocument Kontrolü. 5 Eylül tarihinde <https://www.ismailgursoy.com.tr/printdialog-ve-printdocument-kontrolu/> adresinden erişildi.