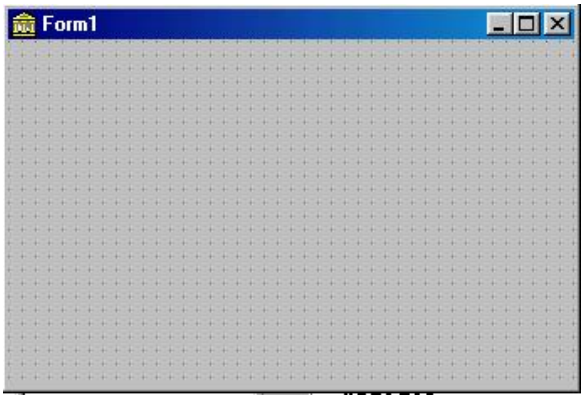
DELPHI DERS NOTLARI

Delphi'yi açtığınızda karşınıza ana form, component palet ve object inspector (özellikler ve olaylar penceresi) gelir.

FORM

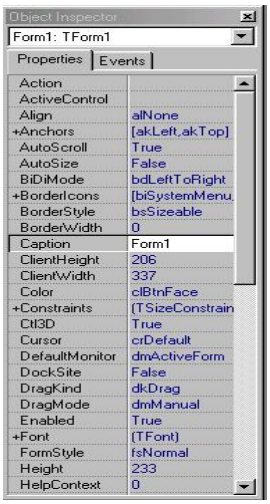


Form yazdığımız programlar için kullanıcı arabiriminin oluşturulduğu bir nesnedir diyebiliriz.

COMPONENT PALET

Bu palet üzerinde delphi'de kullanacağımız kontroller (nesneler) bulunur. Bunlar programlarımız için arabirim oluşturmakta kullandığımız form üzerine yerleştirilir.

OBJECT INSPECTOR



Visual dillerde ve dolayısıyla delphi'de her kontrolün özeliikleri vardır. Bu kontrollerin özelliklerine program dizayn edilirken object inspector denilen pencereden ulaşılır. Bu pencerede görüldüğü gibi iki sekme bulunmaktadır. Bunlardan birisi properties yani özellikler, diğeri ise events yani olaylardır. Olayların ne demek olduğundan ileride bahsedilecektir. Form üzerinde hangi nesne seçiliyse object inspector o nesnenin özelliklerini gösterir. Mesela yandaki şekilde object inspector form1 nesnesinin özelliklerini göstermektedir.

Şimdi sırayla delphi'de kullanacağımız componentleri (nesneleri) ve bunların özelliklerini anlatmaya çalışacağım.

 İlk olarak edit bileşeni ve bunun özellikleriden (properties) bahsedeceğiz.

Kullanıcıların bilgi girmesi için kullanılan nesnedir.

Autoselect: Şayet bu özellik true ise, form üzerindeki nesneler arasında tab tuşu ile dolaşırken edit objesi üzerine gelindiğinde içerisindeki ifadenin seçili olmasını sağlar.

Autosize: Şayet bu özellik true ise, edit objesinin yazı fontu büyütüldüğünde veya küçültüldüğünde edit nesnesi de aynı oranda büyür veya küçülür. Yani yazının tamamı görülebilir.

Borderstyle: Şayet bu özellik bssingle ise edit nesnesi etrafındaki çerçeve görünür, bsnone ise edit nesnesi etrafındaki çerçeve görünmez.

Charcase: Şayet bu özellik ecnormal ise edit nesnesi içerisine normal ifade yazılabilir. eclowercase olması durumunda ifadenin hepsi küçük harfe çevrilerek yazılır. ecuppercase olması durumunda hepsi büyük harfe çevrilerek yazılır.

Color: Edit objesinin arka plan rengini değiştirmeyi sağlar.

Ctl3d: Şayet bu özellik true ise nesne 3 boyutlu olarak görünür.

Cursor: Nesne üzerine gelindiğinde cursor'un alacağı simgeyi seçmek için kullanılır.

Enabled: Şayet bu özellik true ise obje aktiftir, yani çalışma sırasında obje üzerine tab tuşuyla ulaşabilir ve üzerinde değişiklik yapılabilir. Şayet bu özellik false ise obje pasiftir, yani obje üzerine tab tuşuyla ulaşılamaz ve üzerinde değişiklik yapılamaz.

Font: Nesnenin font özelliklerini değiştirmek için kullanılır.

Height: Editbox'un yüksekliğini değiştirmek için kullanılır.

Hint, ShowHint: Kullandığınız programlarda obje üzerine mouse gidildiğinde o objenin hemen altında işleviyle ilgili bir açıklma görünür. İşte bu açıklamayı sizde bu özellikler vasıtasıyla kendi programlarınızda yapabilirsiniz. Objenin hint özelliğine altta görmek istediğiniz açıklamayı yazdıktan sonra, showhint özelliğini true yaparsanız programı çalıştırdığınızda ve özelliklerini değiştirdiğiniz obje üzerine gittiğinizde yazdığınız açıklamanın o objenin hemen altında belirdiğini görürsünüz.

Left: Objenin formun üzerinde soldan ne kadar uzakta bulunacağını belirlemenizi sağlar.

Maxlength: Nesnenin içerisine girilecek ifadenin max uzunluğunu belirlemede kullanılır.

Name: Her nesnenin bir ismi vardır. Bu isimler nesneleri birbirinden ayırır.

Passwordchar: Şayet edit nesnesi bir şifre bilgisi alacaksa, bu nesne içerisine yazılan harflerin hangi karakterle maskeleneceğini belirlemede kullanılır. Örneğin, buraya * işareti konulursa, edit nesnesi içerisne yazılan her harf * olarak belirir.

Read Only: Eğer bu özellik true ise kullanıcı edit nesnesi içerisindeki bilgiye müdahelede bulunamaz.

TabOrder: Form üzerindeki nesneler üzerinde tab tuşuna basıldığında bir sonraki nesneye geçilir. İşte tab tuşuyla nesneler arasında hangi sırada dolaşılacağını tab order özelliği belirler. Buradaki sayı nesneye kaçıncı sırada gelineceğini belirlemede kullanılır.

Tabstop: Şayet bu özellik false ise objeye tab tuşu ile ulaşılamaz.

Text: Edit nesnesi içerisine daha önceden yazılan veya kullanıcı tarafından runtime anında girilen ifade bu özellik vasıtası ile öğrenilir.

Top: Nesnenin formun üztünden ne kadar uzaklıkta olacağını belirlemede kullanılır.

Visible: Şayet bu özellik false ise program çalıştığı zaman obje form üzerinde görünmez.

Width: Objenin genişliğini belirlemede kullanılır.

Etiket belirlemek için kullanılan nesnedir.

 **Caption:** Label nesnesi içerisindeki yazıyı değiştirmek için kullanılan özelliktir.

Bir işlevin yerine getirilmesi için kullanılan nesnedir.

Caption: Bu obje üzerindeki yazıyı değiştirmek için kullanılır. Bazı programlarda görüldüğü gibi tuşalr üzerindeki bir harfin altı çizilidir ve bu harf alt tuşu ile kullanıldığında bu tuşun işlevi yerine getirilir. Şayet böyle bir şey yapılma isteniyorsa bir harfin altı çizilmelidir. Bunun için de caption özelliği değiştirilirken altı çizilmek istenen harfin önüne **&** işareti konulamalıdır. Örneğin caption özelliği k&Apat şeklinde değiştirilirse A harfinin altı çizilir.

Cancel: Bu özellik true ise form çalışırken esc tuşuna basılırsa bu bubutonun işlevi yerine getirilir.

 Çeşitli seçenekler sunulup bu seçeneklerden birinin veya birden fazlasının seçilmesi gerektiğinde kullanılır.

Checked: Eğer bu özellik true ise checkbox seçili, false ise seçili değildir.

 Çeşitli seçenekler sunulup bu seçeneklerden birinin seçilmesi gerektiğinde kullanılır.

Checked özelliği bir önceki seçenekte anlatılan checkbox ile aynıdır.

Listbox nesnesi adında anlaşıldığı üzere belirleyeceğiniz elemanları listelemek için kullanılır.

Items: Listbox içerisine istediğimiz elemanları yazarak dizayn anında eklememizi sağlayan özelliktir.

 Combobox nesnesi açılan liste olarak da tanımlanabilir. Daha önceden belirlenerek listeye eklenmiş seçeneklerden birini seçmek için kullanılır.

Items özelliği listbox'da anlatıldığı gibi kullanılır.

Style:

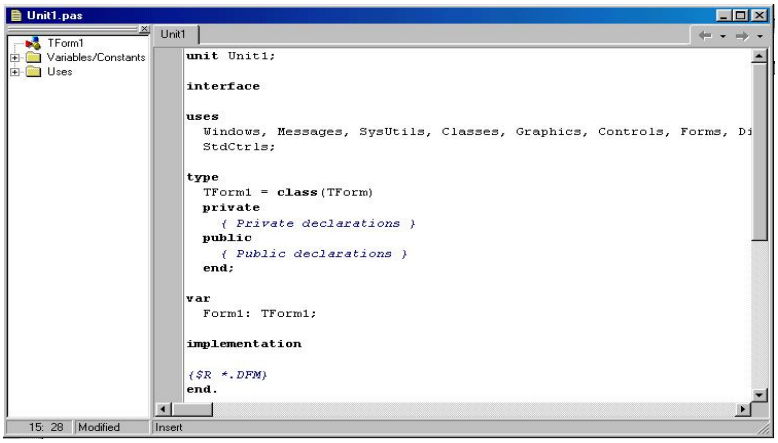
Csdropdown: Şayet bu seçenek seçili ise, kullanıcı bilgi girişi yapabilir. Ayrıca aşağı doğru açıldığı için seçeneklerden birini seçebilir.

Csdropdownlist: Şayet bu seçenek seçili ise, kullanıcı bilgi girişi yapamaz. Aşağı yukarı ok tuşlarını kullanabilir. Ayrıca combobox üzerindeyken vir harfe bastığı zaman o harf ile başlayan seçenek combobox'da görüntülenir. Aşağı doğru açıldığı için seçeneklerden birini seçebilir.

Cssimple: Şayet bu seçenek seçili ise, kullanıcı bilgi girişi yapabilir. Bu seçenekte aşağı doğru açılma yoktur. Seçenekleri aşağı yukarı yön tuşlarını kullanarak bulabilirsiniz.

Buraya kadar özelliklerden bahsettik. Event'lere geçmeden önce delphi'nin kod yazma penceresi hakkında biraz bilgi sahibi olmak gerekir.

Kod penceresine formun arkasındaki pencereye tıklanarak ulaşılır.



Burada değişken, procedure ve fonksiyon tanımlamak için Private ve Public olarak ayrılmış iki kısım görülmektedir. Şayet bir procedure private kısmında tanımlanmışsa bu procedure'e sadece o unit içerisinden ulaşılabilir. (Bu arada delphi'de her form bir unit'dir.). Public'de

tanımlanan bir procedure'e ise diğer unitlerden de ulaşılabilir. Değişken, procedure ve function tanımla şekli pascal ile aynı olduğu için burada tekrar anlatılmayacaktır.

Visual dillerin hepsinde siz bir objeye tıkladığınızda, o obje üzerine mouse ile gittiğinizde, o obje üzerinde bir değişiklik yaptığınızda çeşitli olaylar meydana gelir. Siz bu olaylar meydana geldiğinde hangi kodun işlemesini istiyorsanız ilgili olay procedure'une giderek istediğiniz kodu yazarsınız. Bu procedurler sizler için otomatik olarak oluşturulur. Sizin yapmanız gereken tek şey bu procedure'e giderek istediğiniz kodları yazmaktır. Delphi'de bir visual dil olduğuna göre yukarıda anlatılanlar delphi için de geçerlidir. Buna göre sizin tek bilmeniz gereken hangi olayın ne zaman gerçekleştiğidir.

Şimdi edit nesnesi baz alınarak sizlere hangi olayların ne zaman gerçekleşeceği anlatılacaktır.

OnChange: Nesne üzerinde bir değişiklik olduğu zaman bu olay meydana gelir. Mesela siz edit nesnesi içerisine bir harf eklediğinizde veya sildiğinizde bu olay meydana gelir.

OnClick: Nesne üzerine mouse'un sol tuşu ile tıkladığınızda bu olay meydana gelir.

OnDblClick: Nesne üzerine mouse'un sol tuşu ile çift tıkladığınızda bu olay meydana gelir.

OnEnter: Kontrol o obje üzerine geldiğinde çalışır. Kontrolden kasıt cursordür. Yani siz bir obje üzerine ilk geldiğinizde çalışan olaydır.

OnExit: O obje üzerinde ayrılırken oluşur.

OnKeydown: Obje üzerinde klavyden bir tuşa basıldığında meydana gelir. Basıldığı sürece meydana gelmeye devam eder. OnKeypress'in tanımadığı del, end, pagedown,pageup,insert, ctrl, alt gibi tuşarı da tanır. OnKeypress olayından sonra meydana gelir.

OnKeypress: Obje üzerinde klavyden bir tuşa basıldığında meydana gelir.

OnKeyup: Obje üzerinde basılan tuş bırakıldığında meydana gelir.

OnMouseDown: Mouse'un tuşlarından biri ile bir obje üzerine basıldığında meydana gelir.

OnMouseMove: Mouse ile bir obje üzerine gelindiğinde bu olay meydana gelir.

OnMouseup: Mouse'un tuşlarından biri ile obje üzerine basıldıktan sonra bu tuş bırakıldığında bu olay meydana gelir.

Form'un **oncreate** özelliği ise form ilk çalışırken meydana gelir.

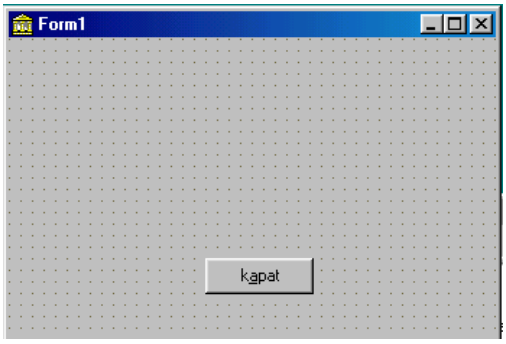
Bir olaya ulaşmak için objenin istenen olayı object inspector'ın events sayfasında bulunduktan sonra yanındaki beyaz boşlukta mouse ile çift tıklamak yeterlidir.

*****Bir objenin bir özelliğine kod yazarak ulaşmak için şu yazılım kuralı uygulanır:**

Objenin name'i .ulaşılmak istenen özellik

Şimdi artık ilk örnek programımızı yapabiliriz.

Örnek1: Bir formumuz ve bu form üzerinde bir butonumuz olsun. Bu butonumuzun caption'ı k&apat olsun. Bu butona tıklandığı zaman formun kapanması istensin. Bunun çin önce aşağıdaki gibi bir form oluşturulur.



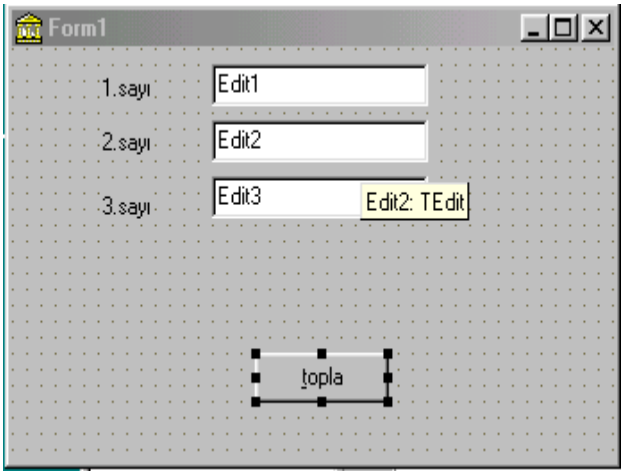
Daha sonra bu butonun onclick olayına gidilip şu kod yazılır.

```
procedure TForm1.Button1Click(Sender: TObject);

begin

        close;

end;
```

Close: bu komut formları kapatmak için kullanılır.

Daha sonra programı çalıştırmak için F9 tuşuna basmak yeterlidir. Ya da formun sol üst köşesindeki yeşil run tuşuna basılabilir.

Örnek2: Form üzerine üç editbox bir de buton ekleyelim. İki editbox'a sayıları girip butona tıkladığımız zaman üçüncü editbox'da iki sayının toplamını görelim.



Burada bilinmesi gereken önemli husus şudur. Editbox'ların içerisine girilen ifadelerin hepsi string ifade olarak kabul edilir. Bu yüzden şayet girilen değerler üzerinde matematiksel işlem yapılacaksa önce bu değerlerin nümeriğe çevrilemsi gerekir.

Ayrıca sonuç yine bir edit objesi içerisine yazdırılacaksa bu obje içerisine sadece string ifadeler yazılabildiğine göre sonucun tekrar stringe çevrilmesi gerekir.

Bu işlemlerin yerine getirilebilmesi için şu iki komutun iyi bilinmesi gerekir.

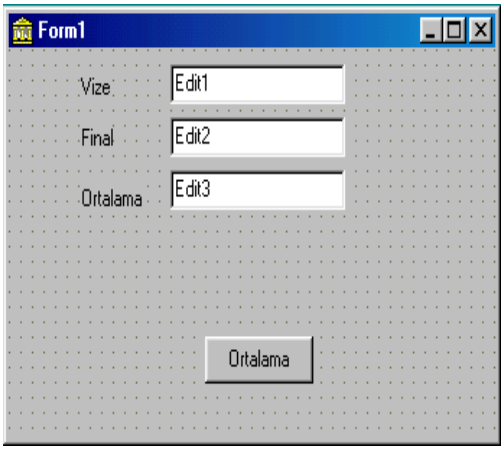
StrtoInt(string değer): String bir değeri nümeriğe çevirir.

InttoStr(numeric değer): Numeric değeri stringe çevirir.

Bunları öğrendiğimize göre topla butonunun onclick event'ine şu kod yazılmalıdır.

```
procedure TForm1.Button1Click(Sender: TObject);

begin

edit3.text:=inttostr(strtoint(edit1.text)+strtoint(edit2.text));

end;
```

Örnek3: Vize ve final notları girildiğinde ortalamayı yuvarlayarak veren bir program yazalım.




```
procedure TForm1.Button1Click(Sender: TObject);

var

ort:real;

begin

ort:=strtoint(edit1.text)*0.4+strtoint(edit2.text)*0.6;
```

```
if (ort-trunc(ort))>0.5 then

edit3.text:=inttostr(trunc(ort)+1)

else

edit3.text:=inttostr(trunc(ort));

end;
```

Buraya kadar hep procedure'ler bizim için oluşturuldu. Peki biz delphi'ye bir procedure veya function'ı nasıl ekleyeceğiz?

Önce procedure veya function'ı önce kullanım amacımıza göre private veya public bölümünde tanımlamalıyız. Daha sonra procedure'lerin yazıldığı bölüme eklemeliyiz.

Mesela yukarıdaki örneği bir procedure yazarak yapmaya çalışalım. Önce private bölümünde function'ımızı tanımlamalıyız.

```
private

function ortala(vize,final:byte):byte;
```

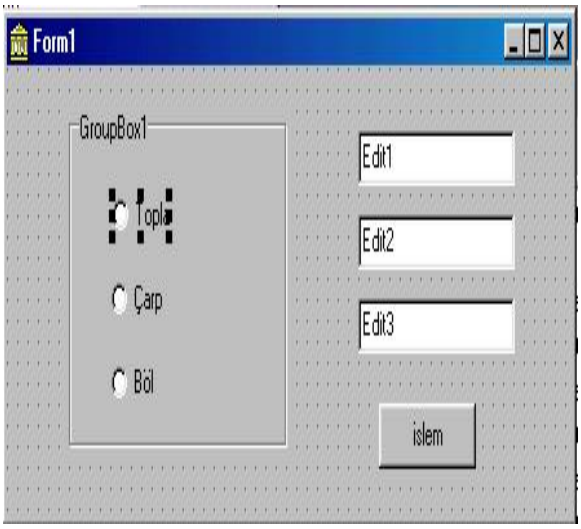
bu bölümü bu şekilde tanımladıktan sonra tanımlamanın hemen yanındayken **CTRL+SHIFT+C** tuşlarına beraber basarsanız aşağıda sizin için function'un gövdesinin sizin için aşağıda oluşturulduğunu görürsünüz.

```
procedure TForm1.Button1Click(Sender: TObject);

begin

edit3.Text:=inttostr(ortala(strtoint(edit1.text),strtoint(edit2.text)));

end;

function TForm1.ortala(vize, final: byte): byte;

var

ort:real;

begin

ort:=vize*0.4+final*0.6;

if (ort-trunc(ort))>0.5 then

ortala:=trunc(ort)+1
```

```
else

ortala:=trunc(ort);

end;
```

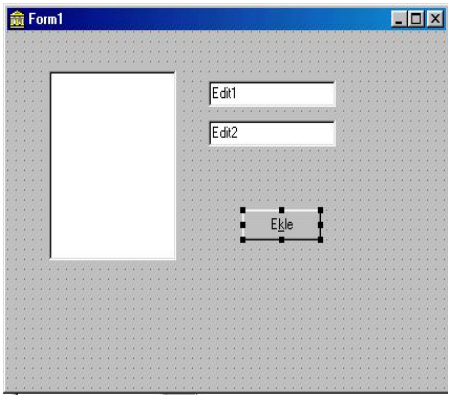
Buradaki tform1 bu procedure veya fonksiyonun form1 isimli forma ait olduğunu gösterir. Aynı function'u form2'de yazmış olsaydık, bu kısım TForm2 olacaktı.

Örnek4: Şimdi aşağıdaki formu oluşturalım ve seçilen seçeneğe göre butona tıklandığında istenilen işlemi yaptıralım.




```
procedure TForm1.Button1Click(Sender: TObject);

begin

if radiobutton1.checked=true then

edit3.Text:=inttostr(strtoint(edit1.text)+strtoint(edit2.text));

if radiobutton2.checked=true then

edit3.Text:=inttostr(strtoint(edit1.text)* strtoint(edit2.text));

if radiobutton3.checked=true then

edit3.Text:=inttostr(strtoint(edit1.text) div strtoint(edit2.text));

end;
```

Örnek5: Aşağıdaki formu oluşturlaım ve edit1'den edit2'ye kadar olan sayıları butona tıkladığımızda listbox!a ekleyelim.



Burada bilinmesi gereken şudur. Listbox'ın içerisine sadece string ifadeler eklenebilir.

Showmessage(string ifade): Bu komut parantez içerisinde verilen ifadeyi bir mesaj kutusunda görüntüler.

```
procedure TForm1.Button1Click(Sender: TObject);

var

i:integer;

begin

listbox1.Clear;//listbox'ın içini temizler

if strtoint(edit1.text)> strtoint(edit2.text) then

for i:=strtoint(edit1.text) to strtoint(edit2.text) do

listbox1.items.add(inttostr(i))

else

showmessage('ilk değer büyük olmalı');
```

```
end;
```

Örnek 6: Yukarıdaki örnekte edit1'den edit2'ye kadar olan sayıları listbox içerisine ekliyorduk. Bu örnekte aynı sayıların faktoriyellerini listbox içerisine ekliyeceğiz.

Önce private bölümünde fonksiyon tanımlandı.

```
private

function faktoriyel(gelen:integer):integer;


function TForm1.faktoriyel(gelen: integer): integer;

var

a,i:integer;

begin

a:=1;

for i:=1 to gelen do

a:=a*i;

faktoriyel:=a;

end;



procedure TForm1.Button1Click(Sender: TObject);

var

i:integer;

begin

listbox1.Clear;

for i:=strtoint(edit1.text) to strtoint(edit2.text) do

listbox1.items.add(inttostr(faktoriyel(i)));
```

```
end;
```