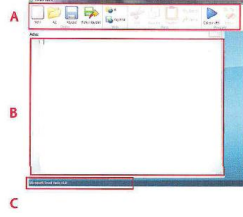


SMALL BASİC ÇALIŞMA NOTLARI

Yazılım, kullanıcıdan gelen istekleri nasıl yapacağını bilgisayara anlatan ve ona yol gösteren komutlar bütünüdür.

Programlama dilleri örnek : C# , C++, Visual Basic, J#, Java, Pascal, Java Script, Phyton, Delphi

Small Basic de yazdığın kodları çalıştırmak için klavyeden **F5** tuşuna basabilir veya araç çubuğunda bulunan **Çalıştır** düğmesini tıklayabilirsin.



Small Basic Bölümleri:

A-Araç Çubuğu

B-Kod Düzenleyici

C-Small Basic Güncelleme



Harf Duyarlı Kodlama: Bu özellik ile bir komutun ilk harfini yazdığında o harfle ilgili başlayan tüm komutlar listelenir.

Yazdığımız kodları **CTRL + S** tuşlarına beraber basarak veya araç çubuğunda bulunan **Kaydet** düğmesini tıklayarak kaydedebiliriz.

Komutlar Listesi :

TextWindow : Small Basic programında ekrana yazı yazmak, mesaj vermek, yazının rengini, arka planını değiştirmek gibi işlemleri yapmak için kullanılan komuttur.

WriteLine : TextWindow komutunun bir özelliğidir. Yazı ekranına bir satır metin yazmak için kullanılır. Metin ekrana yazıldıktan sonra imleç **bir satır alta geçer**. `TextWindow.WriteLine ("Merhaba Dünya")`

Write : TextWindow komutunun bir özelliğidir. Yazı ekranına bir satır metin yazmak için kullanılır. Metin ekrana yazıldıktan sonra imleç **bir satır alta geçmeden yazının yanında bekler**. `TextWindow.write ("Merhaba Dünya")`

BackgroundColor:Textwindow ekranına yazılacak yazının **arkaplan rengini** değiştirir. `textwindow.backgroundColor="red"`

ForegroundColor: Textwindow ekranına yazılacak yazının **rengini** değiştirir. `textwindow.foregroundColor="yellow"`

Read: Textwindow ekranında kullanıcıya bir metin-yazı girmesini sağlar. Girilen yazı-metin ekrandan okunup değişkene aktarılabilir.

Örnek : `Textwindow.Write("Adınızı giriniz: ")`
`isim= Textwindow.Read()`

Readnumber: Textwindow ekranında kullanıcıya bir sayı girmesini sağlar. Girilen sayı ekrandan okunup değişkene aktarılabilir.

Örnek : `Textwindow.Write("Lütfen yaşınızı giriniz = ")`
`yas= TextWindow.ReadNumber()`

İşletim Sistemi : Bir bilgisayardaki en temel ve önemli yazılım işletim sistemidir. Bir işletim sistemi olmayan bilgisayara başka program yükleyemezsin.

İşletim Sistemine Örnek : Microsoft Windows, Linux, Unix, Android, IOS, Mac OS, Symbian.

Yazılım Hataları : Yazılımlarda 2 tür hata ile karşılaşabiliriz; **kodlama hataları** ve **mantık hataları**. Mantık hatalarında program çalışır ancak kodlama hatalarında program hata vererek çalışmayı durdurur.

Hatalı Kod örneği : `Textwindo.write("selam")` → Yazım hatası yapılmıştır. Doğru yazım : `Textwindow.write("selam")`

GİRDİ - İŞLEM - ÇIKTI - DEPOLAMA

Girdi: Bilgisayarı kontrol etmek için kullanılan verilere denir. Bilgisayarlar ancak kendilerine söylenen şeyleri yapan araçlardır.

Girdi-Giriş Araçları Örnekleri : Fare , klavye, mikrofön, joystick, dokunmatik ekran, webcam, tarayıcı

İşlem : Bilgisayarlar en temelde girdi olarak gelen veriyi işleyerek çalışıyorlar. Gelen veriler üzerinde çeşitli hesaplamalar gerçekleştirirler. Veri İşlem Araçları Örnekleri : İşlemci (CPU), Ses kartı , Ekran Kartı, Geçici Bellek (RAM)

Çıktı : Girdi araçları tarafından gönderilen ve işlenen veriler çıktı olarak kullanıcıya sunulur. İşlenen verilerin sonuçlarının kullanıcıya gösterilmesi için cihazlar kullanılır. Veri Çıktı Araçları Örnekleri : Ekran(Monitör), Yazıcı, hoparlör, kulaklık

Depolama: Girdilerin işlenmesinin ardından oluşan çıktıların daha sonra kullanım için saklanmasıdır. Veri depolamak için çeşitli cihazlar kullanılır. Veri Depolama Araçları Örnekleri : Sabit Disk (Harddisk), CD-DVD-Blueray, USB bellek, harici disk

VERİ TÜRLERİ: **Tam sayı** (1-5-2569) , **Ondalıklı sayı** (1,5 - 4,67 - 567,841) , **Metin** ("okul" - "kırmızı araba" - "benim yaşım 15")

Değişken : Bir bilgisayar programının çalışma süresince gereksinim duyduğu **verileri depolamak** için kullandığı bir tür nesnedir.

Değişken Kullanımına Örnek : `isim="Onur"`

`Textwindow.Writeline(isim)` → Ekrana **Onur** yazar.

Değişken Kullanımı Örnekleri

Doğru örnek	Yanlış örnek	Açıklama
isim	ısım	Türkçe karakter kullanılmaz
Sayı1	1.sayı	Sayı ile başlamaz.
Kullanici_adi	Kullanıcı adı	Boşluk kullanılmaz.

Değişkenlerde Sayısal Değerleri Kullanmak: Tanımladığımız değişkenlere sayısal değerler atayabiliriz. **Örnek:** Sayı1=5

KOŞUL VE DALLANMA

Koşul ve dallanma ile programın akışını istediğimiz gibi değiştirebilir, duruma göre bazı kodların çalışmasını sağlar veya engelleyebiliriz.

Koşul Komutu: Değişkende saklanan bir verinin sahip olduğu değere göre karar vermek için kullanılır.

Örnek: if (Clock.Hour > 12) Then // Eğer saat 12 den büyükse
 TextWindow.WriteLine("günaydın") // Ekranda günaydın Yazar
 Else //Saat 12 den büyük değilse
 TextWindow.WriteLine("iyi akşamlar") // Ekranda iyi akşamlar Yazar
 Endif //Koşulu sonlandırır.

Dallanma: Programın akışını yukarıdan aşağı veya aşağıdan yukarı değiştirmemizi sağlar.

Örnek: degisken = 1 //Bir değişken tanımladık ve 1 değerini atadık.
 Basla: //etiket tanımladık. **Basla.**
 TextWindow.WriteLine(degisken) //Ekranaya değişkenin değerini yazdırdık
 Degisken = degisken + 1 //değişkenin değerini her defasında 1 arttırdık.
 If (degisken < 5) then // degisken 5 ten küçük olduğunda çalıştırdık.
 GoTo basla // yukarıdaki koşul doğru olduğunda **basla** etiketine yönlendirdik.
 Endif // if koşul cümlesini sonlandırdık.

Döngü: Bir veya daha fazla satırı **belli bir sayıda** veya **belli bir koşul yerine gelinceye kadar** çalıştırma imkanı verir.

Döngü Çeşitleri: * **For** Döngüsü
 * **While** Döngüsü

For Döngüsü: Bir veya daha fazla satırı **belli bir sayıda** çalıştırmak için kullanılır.

Step: Artış (adım sayısı).

Kullanımı: **For** değişken adı=ilk değer **To** son değer **Step** Artış miktarı
 Tekrar edilecek kod satırı
EndFor

Örnek:

```

1
4
7
10

```

For degisken = 1 **To** 10 **Step** 3 //degisken adında bir değişken tanımladık ve 1 başlangıç değerini verdik
 TextWindow.Writeline(degisken) //Ekranaya her seferinde değişkenin değerini yazdırdık.
EndFor //Döngüyü sonlandırdık.

While Döngüsü: Bir veya daha fazla satırı **belli bir koşul yerine gelinceye kadar** çalıştırmak için kullanılır.

Kullanımı: değişken adı= değer
While (koşul)
 Tekrar edilecek kod satırı
EndWhile

Örnek:

```

1
4
7
10

```

degisken=1 // degisken adlı değişkene 1 değerini verdik.
While (degisken<=10) //değişkenin değeri 10'a eşit veya küçük olduğu sürece çalışır.
 TextWindow.WriteLine(degisken) // Ekranaya her seferinde değişkenin değerini yazdırdık.
 Degisken = degisken + 3 //değişkenin değerini her seferinde 3 arttırdık.
EndWhile // Döngüyü sonlandırdık.

Dikkat!

- For döngüsünde artış miktarını Step komutu ile yapabilirken While döngüsünde böyle bir komut mevcut olmadığı için kendimiz değerini arttırmalıyız.
- While döngüsü dönüş sayısının ilk baştan belli olmadığı durumlarda kullanışlıdır.

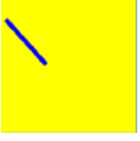
GRAFİKLERLE ÇALIŞMAK:

GraphicsWindow nesnesi ile ekrana çeşitli şekiller çizebilir ve bunları renklendirebiliriz.

GraphicsWindow'un dört önemli özelliği:

- ✓ BackgroundColor //Arka zemin rengi
- ✓ Title //Başlık
- ✓ Width //Genişlik
- ✓ Height //Yükseklik

Örnek:

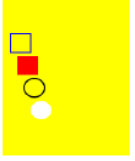


```
GraphicsWindow.BackgroundColor = "Yellow" //Zemin rengini sarı yaptık.
GraphicsWindow.Width = 300 //Pencere genişliğini 300 piksel olarak ayarladık.
GraphicsWindow.Height = 300 //Pencere yüksekliğini 300 piksel olarak ayarladık.
GraphicsWindow.Title = "İlk Çizim" //Programımıza bir başlık ekledik.
GraphicsWindow.PenWidth = 10 //Çizginin kalınlığını 10 piksel yaptık.
GraphicsWindow.PenColor = "Blue" //Çizgi rengini mavi yaptık.
GraphicsWindow.DrawLine(10,50,100,150) //Oluşturacağımız çizginin başlangıç ve bitiş koordinatlarını belirledik.
```

Farklı Geometrik Şekiller

- ✓ DrawRectangle //İçi boş dikdörtgen çizmek için kullanılan komut.
- ✓ FillRectangle //İçi dolu dikdörtgen çizmek için kullanılan komut.
- ✓ DrawEllipse //İçi boş elips çizmek için kullanılan komut.
- ✓ FillEllipse //İçi dolu elips çizmek için kullanılan komut.
- ✓ BrushColor //Sadece çizgi rengini değil çizginin içinde kalan alanında rengini belirler.

Örnek:



```
GraphicsWindow.BackgroundColor = "Yellow" //Zemin rengini sarı yaptık.
GraphicsWindow.Width = 300 //Pencere genişliğini 300 piksel olarak ayarladık.
GraphicsWindow.Height = 300 //Pencere yüksekliğini 300 piksel olarak ayarladık.
GraphicsWindow.PenColor = "Blue" //Çizgi rengini mavi yaptık.
GraphicsWindow.DrawRectangle(10,50,30,25) //İçi boş dikdörtgenin koordinatlarını uzunluk ve yükseklik ölçülerini belirttik.
GraphicsWindow.BrushColor = "Red" //Fırça rengini kırmızı yaptık.
GraphicsWindow.FillRectangle(20,80,30,25) //İçi dolu dikdörtgenin koordinatlarını uzunluk ve yükseklik ölçülerini belirttik.
GraphicsWindow.PenColor = "Black" //Çizgi rengini siyah yaptık.
GraphicsWindow.DrawEllipse(30,110,30,25) //İçi boş elipsin koordinatlarını uzunluk ve yükseklik ölçülerini belirttik.
GraphicsWindow.BrushColor = "White" //Fırça rengini beyaz yaptık.
GraphicsWindow.FillEllipse(40,140,30,25) //İçi dolu elipsin koordinatlarını uzunluk ve yükseklik ölçülerini belirttik.
```

GraphicsWindow.Show() : Small basic programlama dilinde grafikleri çizebileceğiniz pencereyi görüntüleyebilirsiniz.

Piksel Nedir? Dijital görüntüler, yatay ve dikey biçimde yan yana sıralanmış noktalardan oluşur. Bu noktalara **piksel** denir.

Örn : 1920 x 1080 piksellik ekran çözünürlüğü

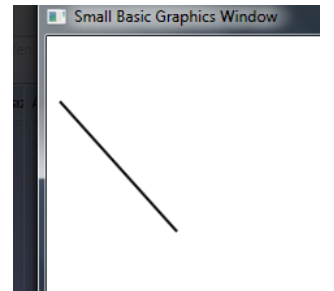
Her piksel ekranda **(X,Y)** gibi bir adreslemeye sahiptir.

Drawline komutu, oluşturacağı çizginin **başlangıç x,y** ve **bitiş x,y** değerlerini ister.

GraphicsWindow.DrawLine (10 , 50 , 100 , 150)

Başlangıç
X,Y

Bitiş
X,Y



Kaplumbağa ile Programlama

Turtle. Show() : Grafik penceresinde bir kaplumbağa gösterilir.

Turtle.Move(150) : Kaplumbağa 150 piksel hareket eder (Move→ Hareket ekmek). Parantez içindeki sayı kadar pixel ilerler.

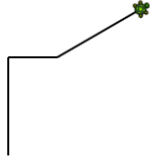
Turtle. TurnRight() : Kaplumbağayı sağa 90 derece döndürür.

Turtle. TurnLeft() : Kaplumbağayı sola 90 derece döndürür.

Turtle. Turn (45) : Turn komutu kaplumbağa nesnesinin yönünü parantezin içine yazılan açı değeri kadar sağa döndürür. Eğer kaplumbağayı sola doğru döndürmek istiyorsanız Turtle.Turn(-30) gibi, parantezin içine eksi – işareti koyarak bir açı değeri yazmalısınız.

Örnek:

```
Turtle.Move(100)
Turtle.TurnRight()
Turtle.Move(50)
Turtle.Turn(-30)
Turtle.Move(100)
```



Rastgele Renk Üretme: GraphicsWindows nesnesinin **GetRandomColor()** metodu her çalıştırıldığında **rastgele bir renk** üretilir.

Alt Programlar (Subroutine)

İngilizcesi **Subroutine** olan **Altprogramlar**, program içerisinde tekrar tekrar yazmak zorunda kaldığımız kodları sadece bir defa yazarak, birden fazla kullanmamıza izin verirler. Altprogramlar kullanmanın avantajı yazdığımız kodları daha rahat okunabilir hale getirmesidir. **Sub** anahtar kelimesiyle başlar **EndSub** anahtar kelimesiyle biter. Alt programların sonunda () parantezler vardır.

Örnek:

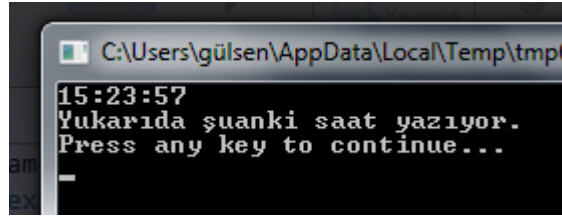
ZamaniEkranayaYaz()

```
Textwindow.writeline("Yukarıda şuanki saat yazıyor.")
```

Sub ZamaniEkranayaYaz

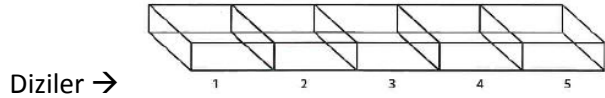
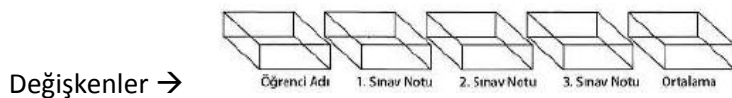
```
Textwindow.Writeline( clock.time )
```

EndSub



Diziler (Array)

Birden fazla değişkende tutulabilecek değerleri tek bir değişken içinde depolamamızı sağlar. Dizileri, sadece ilgili ve benzer değerleri, aynı değişken ismi altında depolamak için kullanırız. Değişken adından sonra köşeli parantez [] içinde verdiğimiz sayı, bunun dizi türünde bir değişken olduğunu gösterir.



Kullanımı →

```
ogrenci_bilgisi [0] ="onur"
ogrenci_bilgisi [1] =90
ogrenci_bilgisi [2] =80
```

} **ogrenci_bilgisi** dizi değişkenin 0,1 ve 2 bölümlerine gerekli bilgileri kaydediyoruz.

Bilgisayarla Etkileşim

Event (Olay): Kullanıcının bilgisayarla etkileşim kurmasını sağlayan fareyi tıklama, çift tıklama, sağ tıklama, klavyeden tuşa basma gibi olaylara event (olay) denir.

Grafiksel Kullanıcı Arayüzü: Windows benzeri, kullanıcının bilgisayarla etkileşiminin görsel nesnelere ile sağlandığı ortamlara grafiksel kullanıcı arayüzü (**Graphical User Interface – GUI**) adı verilir.

GraphicsWindow.MouseDown

GraphicsWindow.MouseUp

GraphicsWindow.MouseMove

GraphicsWindow.KeyDown

GraphicsWindow.KeyUP

Örnek:

GraphicsWindow.MouseDown= FareTiklandiginda //fareye basıldığında altprogram çalışıyor

Sub FareTiklandiginda

GraphicsWindow.BackgroundColor=GraphicsWindow.GetRandomColor()

EndSub

5

BİR BİLGİSAYAR PROGRAMINI ADIM ADIM PLANLAMAK

Tüm üretim süreçleri 4 ana aşamadan oluşur;

1-Analiz

2-Tasarım

3-Geliştirme (Üretim)

4-Test

1-Analiz: Yazılım geliştirmenin ilk adımı, çözüm geliştirmek istediğimiz problemi tüm yönleriyle incelemektir.

Bu süreçte;

- Geliştirilecek yazılımı kullanacak kişilerle görüşme yapılarak istekleri ve problemleri ortaya çıkarılır,
- Geliştirilecek yazılımı kullanacak olan şirketin veya kurumun yetkililerinden programla ilgili beklentileri öğrenilir,
- Yazılımın hangi işleri yapacağı ortaya çıkarılır,
- Geliştirilecek yazılımın çalışacağı bilgisayarların işlem yapma ve hafıza kapasiteleri incelenir.

2-Tasarım: Analiz aşamasında elde edilen bilgiler doğrultusunda nasıl bir yazılım geliştireceğimize karar veririz.

- Bittiğinde nasıl bir program olacağını bu aşamada kağıt üzerinde görebiliriz,
- Tüm metot ve özellikler listelendiğinden atlanılan herhangi bir özellik olup olmadığını kontrol edebiliriz,
- Birbirini tekrarlayan metot ve özellikler olup olmadığını görebiliriz,
- Benzer kod içerecek metot ve özellikler için alt programlar kullanmaya karar verebiliriz.

Akış Şeması: Yazılımın, kodlamaya başlamadan önce nasıl çalışacağını belirleyebilmek için Akış Şemaları kullanılır.

3-Geliştirme: Bu aşamada bilgisayar programcıları yazılımı kodlarla geliştirmeye başlarlar.

4-Test: Programcılardan başka birilerinin de yazılımı kullanması ve kurcalamasıdır. Test aşaması, büyük yazılım projelerinde temelde 2 aşamada gerçekleştirilir;

-Alfa Test

-Beta Test

Alfa Testi: Programı yapan ekibin dışında başka şirket veya kurumlarda çalışan uzmanların, yazılımı amacına uygun olarak kullanarak, ortaya çıkabilecek hataları ve eksikleri programı geliştiren ekibe bildirmesi çalışmasıdır.

Beta Testi: Bilgisay programını satın alabilecek kişilere, yazılımın ücretsiz olarak verilmesi ve bu kişilerin programı belirli bir süre kullanmalarını sağlamaktır.

2.DÖNEM 1.YAZILISI 22-03-2019 TARİHİNDE CUMA GÜNÜ 3.DERS TÜM 6.SINIFLAR ORTAK SINAV OLACAKTIR.

SINAVDA SADECE SMALL BASİCTEN SORULAR OLACAKTIR. SINAV SORULARI TEST OLACAKTIR.
BU ÇALIŞMA NOTLARI OKULUMUZUN DUYURULAR BÖLÜMÜNDE BİLİŞİM TEKNOLOJİLERİ
ÇALIŞMA NOTLARI BÖLÜMÜNDE VE EBA DERS SİSTEMİNİZDE BULABİLİRSİNİZ.