



Gelişen Öğretmen Akademisi

Bilgisayar Programlamaya Giriş

Doç. Dr. Özgür ZEYDAN

<https://ozgurzeydan.com.tr/>



Öğrenme Çıktıları

Ders sonunda katılımcılar;

- Bilgisayar programlama ile ilgili temel kavramlar (algoritma, sözde kod, akış şeması) hakkında bilgi sahibi olacaklardır.
- Program geliştirme sürecini tanıyacaklardır.
- Yapısal programlama elemanlarını (sıralı yapı, karar ve döngü yapıları) bileceklerdir.
- R programlama dilini tanıyacaklardır.
- R programlama dilinde basit kod parçacıkları (script) yazıp çalıştırabileceklerdir.

Bilgisayar Programlama

- **Bilgisayar** programlanabilir bir makinedir. Aritmetik ve mantıksal işlemler yapabilir. Kendisine verilen sıralı komutları çalıştırır ve sonucu çıktı olarak verir.
- **Bilgisayar programlama**, bir programlama diline ait çeşitli komutlar yardımıyla bilgisayara belli bir işi yaptırmaktır.
- **Programlama dili**, Bir programcının bir bilgisayara ne yapmasını istediğini anlatmasının standartlaştırılmış bir yoludur.
- **Programcılar**, programlama dili kullanarak bilgisayarı programlaya kişilerdir. Belirli işleri / görevleri yazdıkları programlar ile bilgisayara yaptırabilirler.

Temel Kavramlar

- **Algoritma**, bir problemi çözmek için kullanılan yol (komut, prosedür ve formüller).
- **Sözde kod (Pseudocode)** bir problemin çözümü için gerekli adımların sözel olarak ifade edilmesidir.
- **Kaynak kod**, programcı tarafından metin editöründe veya görsel editörde yazılmış kodların tümüdür.
- Herhangi bir yazılımın işlenip makine diline çevrilmeden önce insanların okuyup üzerinde çalışabildiği programlama diliyle yazılmış hali.
- Kaynak kodlar **derleyici** tarafından derlenebilir veya **yorumlayıcı** tarafından çalıştırılabilir.

Program Geliştirme Süreci

1. **Problemin analiz edilmesi:** programın girdileri, çıktıları ve bunlar arasındaki ilişkilerin belirlenmesi.
2. **Problem çözümünün tasarlanması:** problem çözümü için mantıklı adımlar tanımlanmalıdır. Bu adımlar algoritma olarak adlandırılır.
3. **Arayüzün belirlenmesi:** girdilerin kullanıcıdan nasıl isteneceği (ekrana yazı yazdırılarak sorulacak veya dosyadan okunacak vb.) ve çıktının nasıl gösterileceği belirlenir.

Program Geliştirme Süreci

4. **Kodlama işlemi:** algoritması yazılan problem kodlama kullanılarak programlama diline dönüştürülür.
5. **Programın test edilmesi ve hata tespiti:** Program farklı değişkenler ile kontrol edilir, hatasız çalıştığından emin olunur.
6. **Dokümantasyonun hazırlanması:** Program, başkaları tarafından kullanılacaksa programın çalışma mantığı, yapısı, girdi ve çıktıları diğer kullanıcıların anlayacağı şekilde açıklanmalıdır.

Algoritma Geliştirme Mantığı

- Program sonucunda elde edilmesi planlanan **çıkıtı** nedir?
- Programın çalışması için hangi **girdiler** kullanıcıdan istenmelidir?
- Program girdileri kullanarak çıkıtıyı hangi **işlemler** yoluyla hesaplamalıdır?

➤ **Örnek:** Bilgisayarın, bir dikdörtgenin alanını ve çevresini hesaplamasını istersek;

➤ Girdiler: **a** ve **b** uzunlukları

➤ Çıktılar: **alan** ve **çevre**





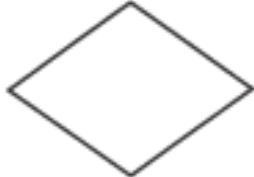
➤ İşlemler: alan ve çevrenin hesaplanması için kullanılan **formüllerdir** (**$alan = a * b$** , **$çevre = 2 * (a + b)$**).

Temel Kavramlar - Söзде Kod

- Günlük konuşma dilindedir.
- Gerçek programlama dili değildir.
- Bilgisayar tarafından anlaşılmaz (derlenemez veya çalıştırılmaz).
- Programcıların algoritma geliştirmesine yardımcı olur.

Temel Kavramlar - Akış Şeması

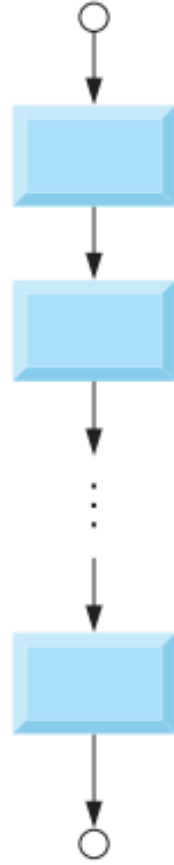
- Akış şemaları ilişkileri tanımlamak için belirli geometrik şekiller ve oklar kullanır.
- **Akış çizgisi**, programın akış yönünü gösterir.
- Terminal, programın **başlangıç** ve **bitişini** belirtmek için kullanılır.
- **Girdi / Çıktı**, programın okuma yaptığı ve hesaplanan veriyi yazdırdığı yerlerde kullanılır.
- Aritmetik ve mantıksal **işlemler** için kullanılır.
- **Karar verme işlemleri** için kullanılır.

| Symbol, | Name, |
|---|---------------------|
|  | <i>Flowline</i> |
|  | <i>Terminal</i> |
|  | <i>Input/Output</i> |
|  | <i>Processing</i> |
|  | <i>Decision</i> |

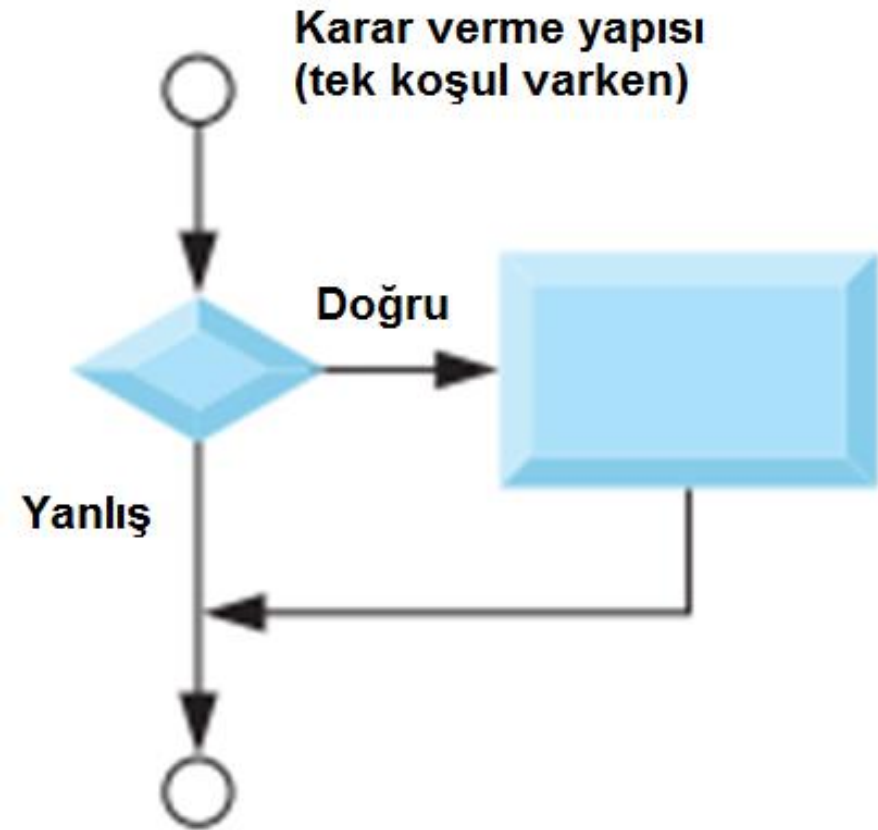
Yapısal Programlama

- Yapısal programlama ilk olarak **Corrado Böhm** ve **Giuseppe Jacopini** tarafından önerilmiştir. Bu iki matematikçi herhangi bir bilgisayar programının şu 3 temel yapı ile yazılabileceğini belirtmişlerdir:
- **Sıralama**: komutların hangi sırayı takip ederek çalıştırılacağını gösterir.
- **Karar verme (seçim) yapıları**: komutlar belirli bir **durumun** DOĞRU (TRUE) veya YANLIŞ (FALSE) olmasına göre çalıştırılır.
- **Döngüler (tekrarlar)**: komutlar belirli bir **durumun** DOĞRU (TRUE) veya YANLIŞ (FALSE) olmasına göre **sürekli olarak** çalıştırılır.
- *Böhm, C., Jacopini, G. (1966). "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules". Communications of the ACM, 9 (5): 366–371.*

Sıralama

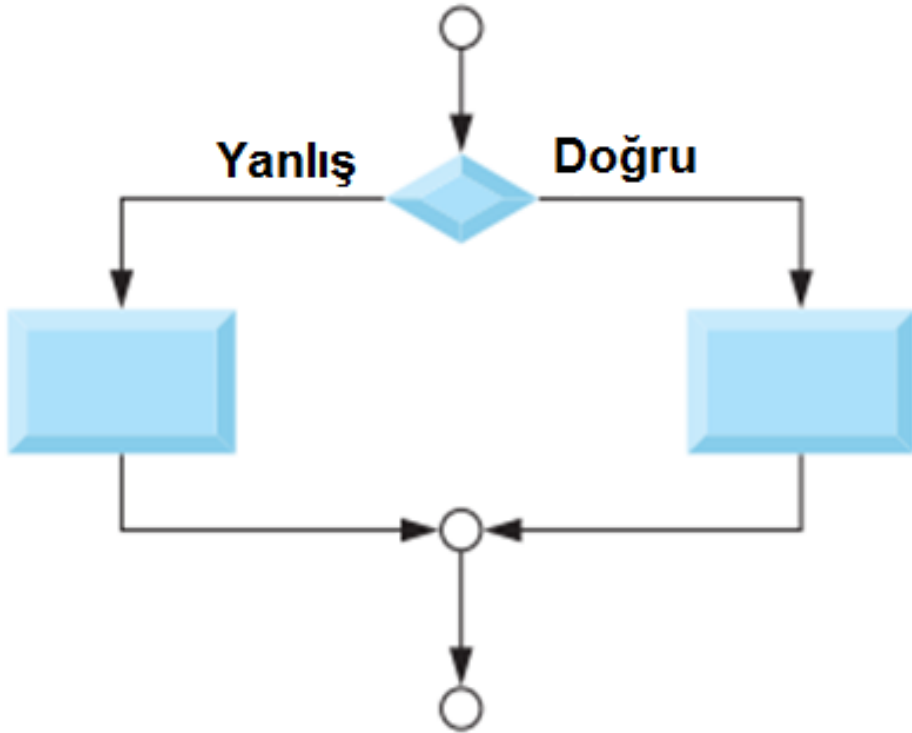


Karar Verme

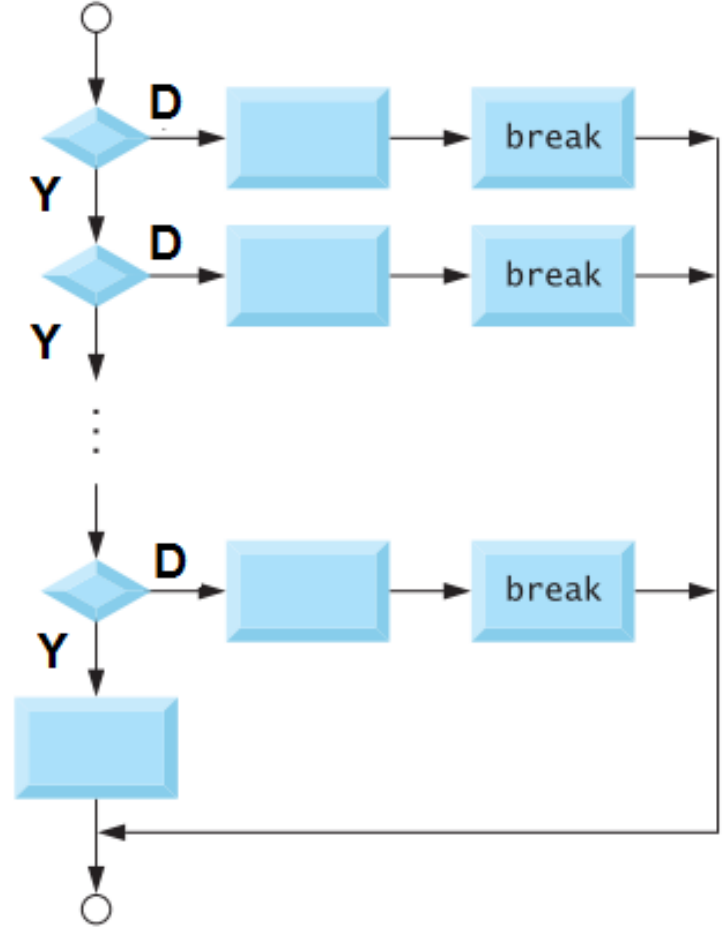


Karar Verme

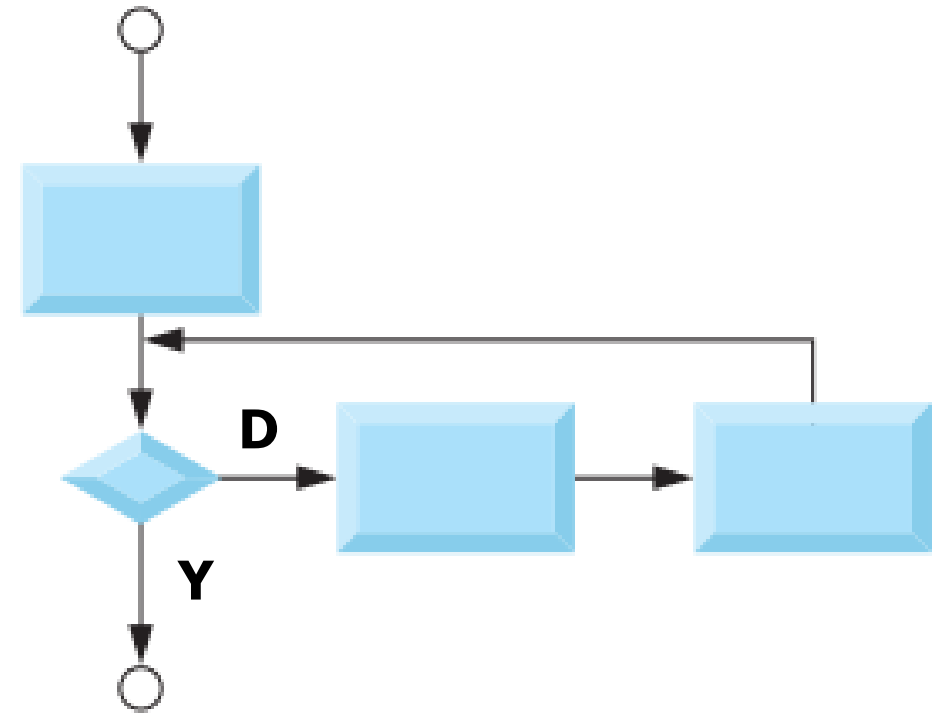
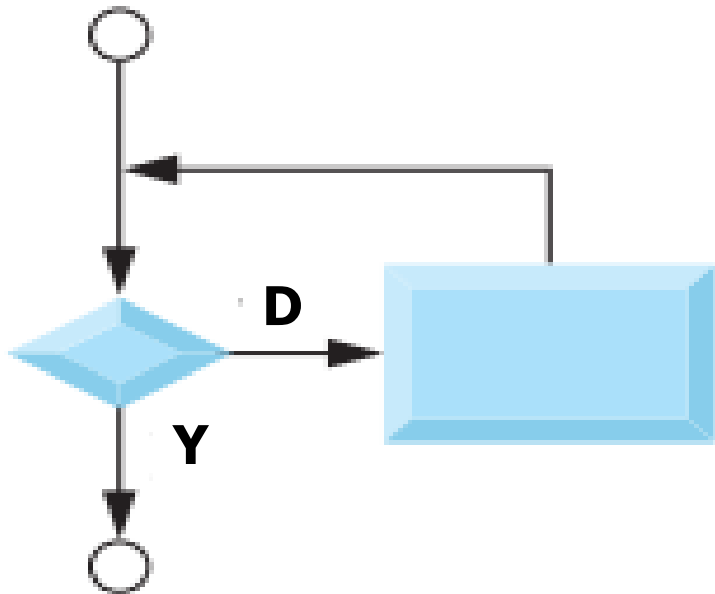
Karar verme yapısı
(iki koşul varken)



Karar verme yapısı
(çok koşul varsa)



Döngüler (Tekrarlar)



Programlama Dilleri

Programming languages you should **learn** to become

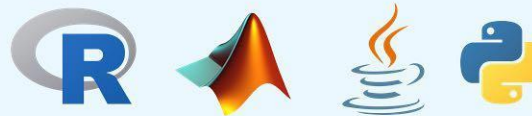
WEB DEVELOPER



GAME DEVELOPER



DATA ANALYSIS



DESKTOP DEVELOPER



MOBILE APP DEVELOPER



Kaynak Kod Örnekleri

1'den 100'e kadar olan sayıların toplamını hesaplamak

➤ C Programlama Dili:

```
#include <stdio.h>
int main()
{
    int i, toplam=0;
    for (i=1; i<=100; i++)
        { toplam = toplam + i; }
    printf(toplam);
    getch();
    return 0;
}
```

➤ Python Programlama Dili:

```
print(sum(range(1,101)))
```

➤ C++ Programlama Dili:

```
#include <iostream>
using namespace std;
int main()
{
    int i, toplam = 0;
    for(i=1; i<=100; i++)
        { toplam = toplam + i; }
    cout << toplam;
    return 0;
}
```

➤ R Programlama Dili:

```
sum(1:100)
```



R Programlama Diline Giriş



R Programlama Dili

- R, istatistiksel hesaplama, grafik çizim programı ve programlama dilidir.
- 1993 yılında ilk kez Yeni Zelanda Auckland Üniversitesi'nden **Ross Ihaka** ve **Robert Gentleman** tarafından geliştirilmiştir.
- R Geliştirme Çekirdek Ekibi tarafından geliştirilmeye devam etmektedir.
- Kullanıcılar tarafından geliştirilen paketler sayesinde oldukça popüler olmuştur.
- Açık kaynak kodlu ve ücretsizdir.
- Windows, Linux ve Mac OS X işletim sistemlerinde çalışmaktadır.
- Veri biliminde ve istatistikte en çok kullanılan programlama dilidir.

R ile Neler Yapılabilir?

- Kodlama
- Veri analizi
- İstatistiksel analiz
- Görselleştirme



R yazılımının indirilmesi



➤ <https://cran.r-project.org/>

Download and Install R

Precompiled binary distributions of the base versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

1

Subdirectories:

[base](#)

2

[contrib](#)

[old contrib](#)

[Rtools](#)

3

[Download R-4.4.0 for Windows](#) (82 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)

RStudio IDE



➤ <https://rstudio.com/products/rstudio/download/>

1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

[DOWNLOAD AND INSTALL R](#)

2: Install RStudio

[DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS](#)

Size: 215.66 MB | [SHA-256: D3C03C42](#) | Version: 2023.12.1+402 |
Released: 2024-01-29

RStudio Ekranı

Editör
(script veya fonksiyon yazmak için)

Terminal

Geçmiş

Ortam

Dosyalar

| Name | Type | Len... | Size | Value |
|--------|--------|--------|-------|---------------|
| age | num... | 10 | 17... | num [1:10]... |
| weight | num... | 10 | 17... | num [1:10]... |
| x | num... | 20 | 20... | num [1:20]... |

| Name | Size | Mo |
|-------------------|--------|----|
| .. | | |
| .RData | 2.8 KB | Nc |
| .Rhistory | 3.6 KB | De |
| booksales.csv | 240 B | Nc |
| booksales.xls | 8.3 KB | Nc |
| dosyam.Rdata | 2.6 KB | Nc |
| filename.pdf | 3.4 KB | Nc |
| openair_2.6-3.zip | 3.2 MB | De |

Girdi - Çıktı Komutları

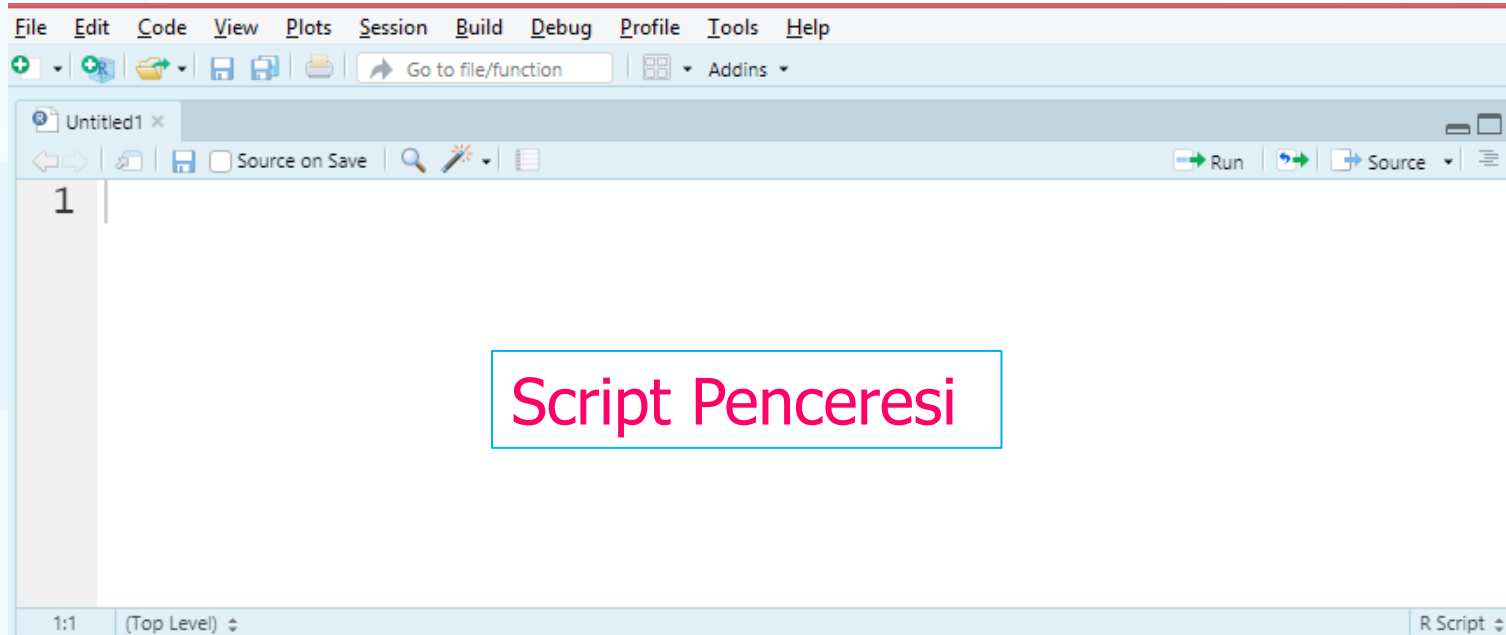
- `readline()` fonksiyonu kullanıcı tarafından **veri girilmesi** amacıyla kullanılır.
- `x=readline(prompt = "Bir sayı yazınız: ")`
- `as.numeric()` fonksiyonu girilen karakteri sayıya çevirir.
- `x=as.numeric(readline(prompt = "Bir sayı yazınız: "))`
- `print()` fonksiyonu bir değişkenin değerini **ekrana yazdırmak** için kullanılır.
- `print(x)`
- Uzun ve birleştirilmiş metinler yazdırılacaksa `paste()` fonksiyonu ile birleştirme yapılır.
- `print(paste("yazılan sayı: ", x))`

Girdi - Çıktı Komutları

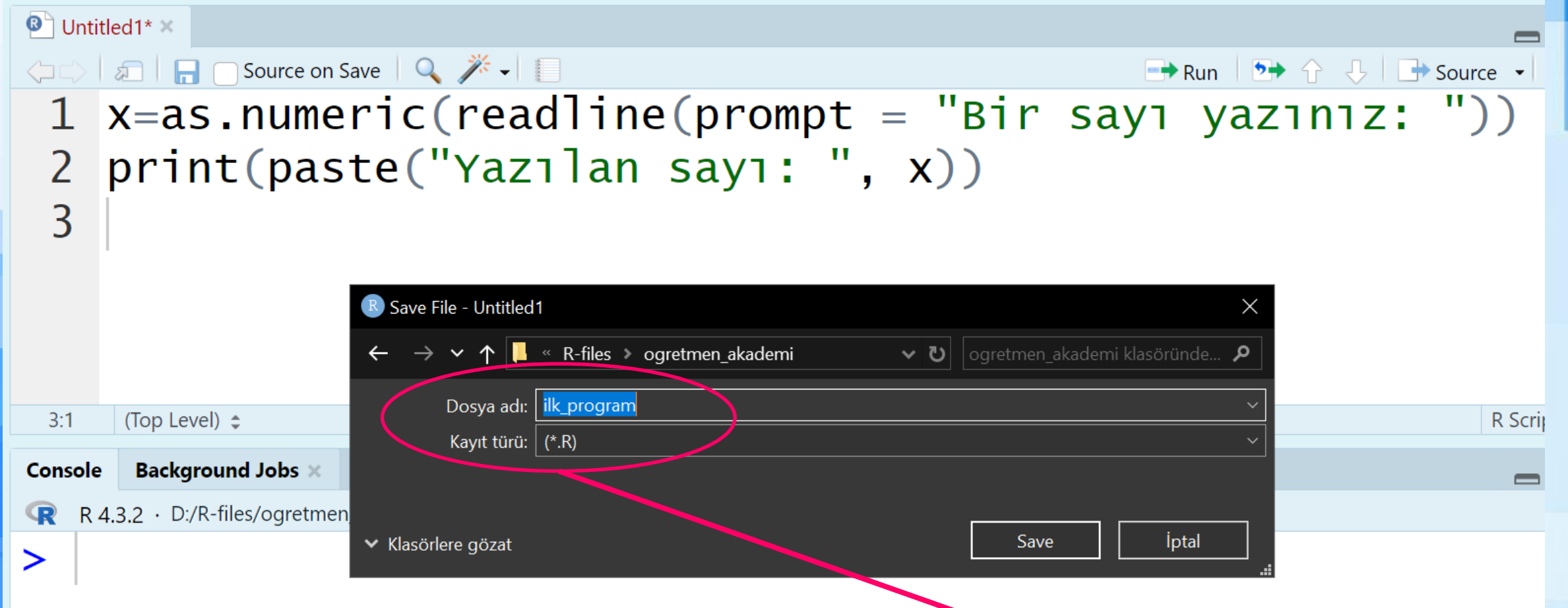
```
> x=as.numeric(readline(prompt = "Bir sayı yazınız: "))
Bir sayı yazınız: 5
> print(x)
[1] 5
> print(paste("Yazılan sayı: ", x))
[1] "Yazılan sayı: 5"
```

Script Oluřturma

- Daha sonra ihtiya duyacađımız kodları **script** yazarak kaydedebiliriz.
- R'de yeni script oluřturmak iin **File > New File > R Script** yolu izlenir veya **CTRL + SHIFT + N** kısıyolu kullanılır.



Scriptin Kaydedilmesi



```
1 x=as.numeric(readline(prompt = "Bir sayı yazınız: "))
2 print(paste("Yazılan sayı: ", x))
3
```

Save File - Untitled1

Dosya adı: ilk_program

Kayıt türü: (*.R)

Save İptal

Script adı yazılır. Dosya *.R uzantısı ile kaydedilir.

Script'in Çalıştırılması

- Scripti çalıştırmak için konsolda `source("ilk_program.R")` yazılır.

```
> source("ilk_program.R")  
Bir sayı yazınız: 10  
[1] "Yazılan sayı: 10"
```

Programlama Örneği – 1

İki Sayının Toplamını Hesaplayan Program

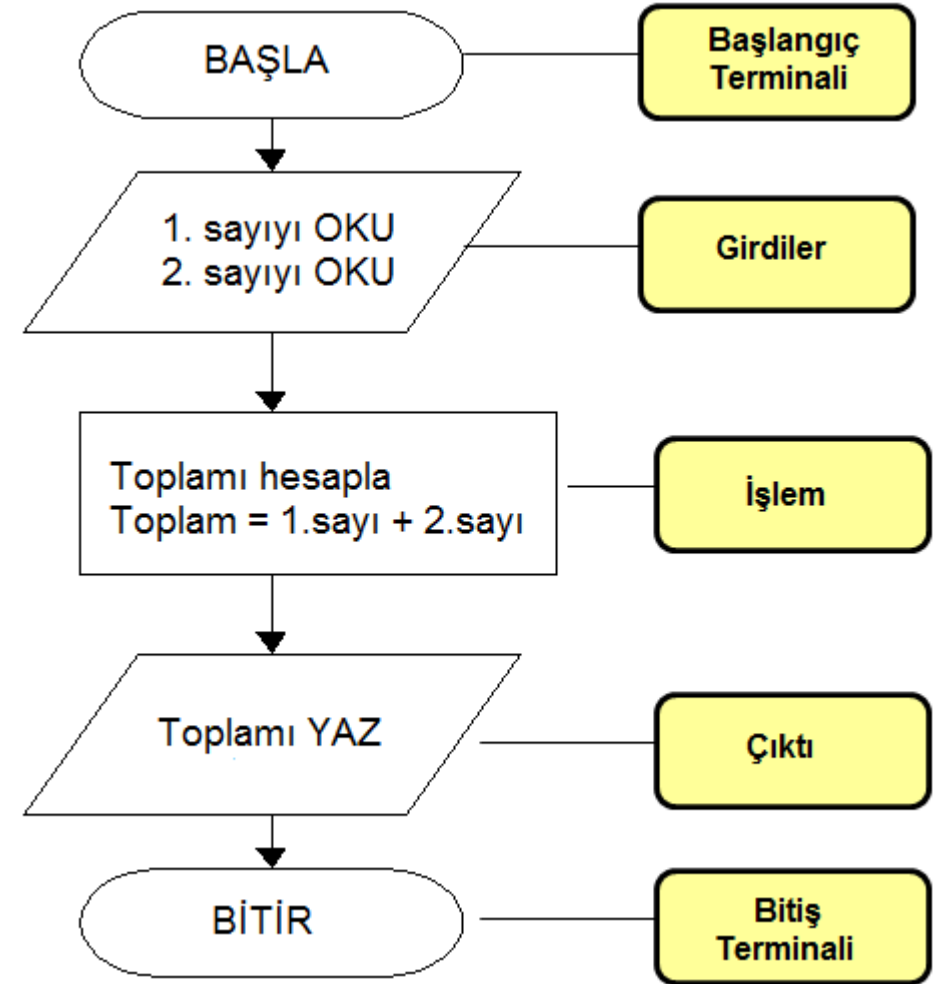
- İki sayının toplamını hesaplayıp ekrana yazdıran bir program yazınız.
- **Programın analizi**
- Girdiler: `sayi1` ve `sayi2`
- İşlem: $\text{toplam} = \text{sayi1} + \text{sayi2}$
- Çıktılar: `toplam`

Programlama Örneđi – 1

Sözde Kod ve Akış Şeması

Sözde Kod

- BAŞLA
- 1. sayıyı OKU (sayi1)
- 2. sayıyı OKU (sayi2)
- toplam = sayi1 + sayi2
- toplam deđerini ekrana YAZ
- BİTİR



Programlama Örneği – 1

R Kaynak Kodu ve Ekran Çıktısı

- `sayi1=as.numeric(readline(prompt = "Birinci sayıyı yazınız: "))`
- `sayi2=as.numeric(readline(prompt = "İkinci sayıyı yazınız: "))`
- `toplam = sayi1 + sayi2`
- `print(paste("Toplam = ", toplam))`

```
> source('iki_sayi_topla.R')
Birinci sayıyı yazınız: 15
İkinci sayıyı yazınız: 20
[1] "Toplam = 35"
```

Programlama Örneği – 2

İki Sayının Toplamını Hesaplayan Program

- Uzunlukları verilen bir dikdörtgenin alanını ve çevresini hesaplayan bir program yazınız.
- **Programın analizi**
- Girdiler: Dikdörtgenin **a** ve **b** kenar uzunlukları
- İşlemler: $\text{alan} = a * b$ $\text{cevre} = 2 * (a + b)$
- Çıktılar: **alan** ve **cevre**

Programlama Örneği – 2

R Kaynak Kodu ve Ekran Çıktısı

- `a=as.numeric(readline(prompt = "Birinci kenar uzunluğu: "))`
- `b=as.numeric(readline(prompt = "İkinci kenar uzunluğu: "))`
- `alan=a*b`
- `cevre=2*(a+b)`
- `print(paste("Dikdörtgen Alanı = ", alan))`
- `print(paste("Dikdörtgen Cevresi = ", cevre))`

```
> source('dikdortgen.R')
Birinci kenar uzunluğu: 5
İkinci kenar uzunluğu: 4
[1] "Dikdörtgen Alanı = 20"
[1] "Dikdörtgen Cevresi = 18"
```

Programlama Örneği – 3

İki Sayının Toplamını Hesaplayan Program

- Uzunlukları verilen bir silindirin yüzey alanını ve hacmini hesaplayan bir program yazınız.
- π sayısı için `pi` fonksiyonunu kullanabilirsiniz.
- Ondalıklı basamakları yuvarlamak için: `round(..., digits = 1)`
- **Programın analizi**
- Girdiler: Silindirin `r` ve `h` uzunlukları
- İşlemler: $yuze_alani = 2 * pi * r * h + 2 * pi * r^2$
 $hacim = pi * r^2 * h$
- Çıktılar: `yuze_alani` ve `hacim`

Programlama Örneği – 3

R Kaynak Kodu ve Ekran Çıktısı

```
> r=as.numeric(readline(prompt = "Silindir yarıçapı: "))
> h=as.numeric(readline(prompt = "Silindir yüksekliği: "))
> yuzey_alani = round(2*pi*r*h + 2*pi*r^2, digits = 1)
> hacim = round(pi*r^2*h, digits = 1)
> print(paste("Silindir Yüzey Alanı = ", yuzey_alani))
> print(paste("Silindir Hacmi = ", hacim))
```

```
> source('silindir.R')
Silindir yarıçapı: 5
Silindir yüksekliği: 10
[1] "Silindir Yüzey Alanı = 471.2"
[1] "Silindir Hacmi = 785.4"
```

Programlama Örneđi – 4

Geçme Kalma Durumunu Hesaplayan Program

- Bir öğrencinin notu ve geçme kalma durumunu hesaplayan program yazınız. Program öğrencinin **vize** ve **final** notlarını okuyarak **ortalamasını** hesaplayacaktır. Ortalamanın 60 veya daha fazla olması durumunda öğrenci dersi geçecek aksi halde kalacaktır.
- **Programın analizi**
- Girdiler: Öğrencinin **vize** ve **final** notları
- İşlemler: $ortalama = (vize + final) / 2$
- **Karar verme yapısı** ile ortalama kontrol edilecek.
- Çıktılar: **Geçti** veya **Kaldı** ifadesi yazdırılacak.

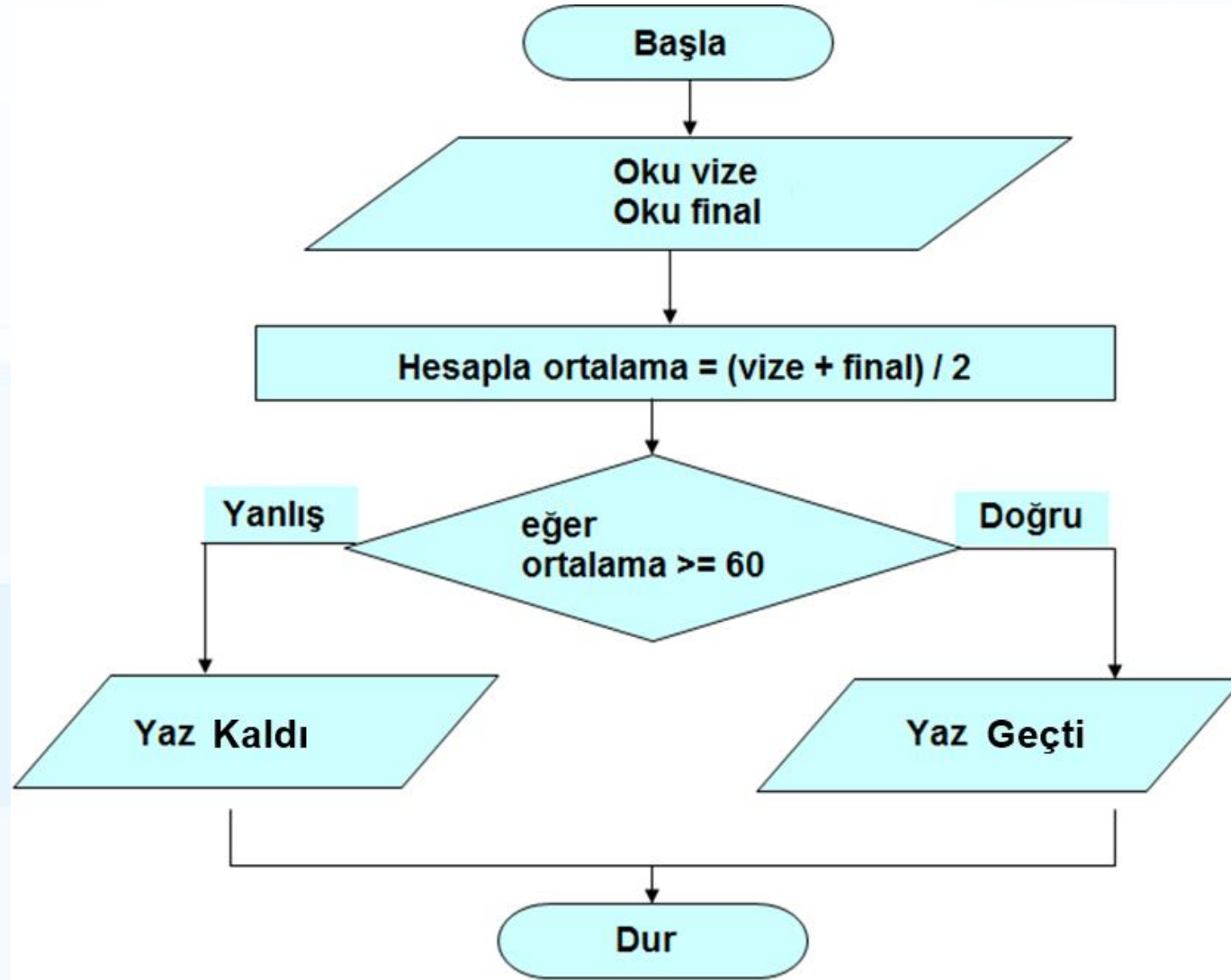
Programlama Örneği – 4

Sözde Kod

- BAŞLA
- YAZ "öğrencinin vize notunu yazınız: "
- OKU *vize*
- YAZ "öğrencinin final notunu yazınız: "
- OKU *final*
- Hesapla $\text{ortalama} = (\text{vize} + \text{final})/2$
- Eğer *ortalama* ≥ 60 ise YAZ "Geçti."
- Eğer *ortalama* < 60 ise YAZ "Kaldı."
- BİTİR

Programlama Örneği – 4

Akış Şeması



Programlama Örneği – 4

R Kaynak Kodu ve Ekran Çıktısı

```
➤ vize=as.numeric(readline(prompt = "Vize notu: "))
➤ final=as.numeric(readline(prompt = "Final notu: "))
➤ ortalama = (vize + final)/2
➤ if (ortalama>=60) {
➤ print("Geçti")
➤ } else {
➤ print("Kaldı")
➤ }
```

```
> source('ortalama.R')
Vize notu: 50
Final notu: 75
[1] "Geçti"
> source('ortalama.R')
Vize notu: 60
Final notu: 42
[1] "Kaldı"
```

Programlama Örneği – 5

İkinci Derece Denklem Çözümü

➤ $ax^2+bx+c=0$ şeklindeki bir 2. derece denklemin köklerini hesaplayan program yazınız. Program kullanıcıdan a , b ve c katsayılarını isteyecek ve Δ değerini hesaplayacaktır. $\Delta \geq 0$ olması durumunda x_1 ve x_2 hesaplanıp ekrana yazdırılacak, aksi halde "reel kök yok" mesajı ekrana yazdırılacaktır. Karekök fonksiyonu: `sqrt(...)`

➤ Programın analizi

➤ Girdiler: a , b ve c katsayıları

➤ İşlemler: $\Delta = b^2-4*a*c$

➤ Karar verme yapısı: $\Delta \geq 0$ ise $x_{1,2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$, $\Delta < 0$ ise reel kök yok

➤ Çıktılar: x_1 ve x_2 veya "reel kök yok" ifadesi yazdırılacak.

Programlama Örneği – 5

R Kaynak Kodu

```
➤ a=as.numeric(readline(prompt = "a: "))
➤ b=as.numeric(readline(prompt = "b: "))
➤ c=as.numeric(readline(prompt = "c: "))
➤ delta=b^2-4*a*c
➤ if (delta>=0) {
➤ x1=(-b-sqrt(delta))/(2*a)
➤ x2=(-b+sqrt(delta))/(2*a)
➤ print(paste("Denklemin kökleri: ",x1," ",x2))
➤ } else {
➤ print("Reel kök yok.")
➤ }
```

Programlama Örneği – 5

Ekran Çıktısı

```
> source('idd.R')  
a: 1  
b: 2  
c: -15  
[1] "Denklemin kökleri:  -5   3"  
> source('idd.R')  
a: 1  
b: 2  
c: 15  
[1] "Reel kök yok."
```


Programlama Örneği – 6

1'den 10'a Kadar Olan Sayıların Kareleri

- 1'den 10'a kadar olan sayıların karelerini hesaplayıp yazdıran bir program yazınız.
- Girdiler: programın girdisi yoktur.
- İşlemler:
 - **Sayaç kontrollü** ile 1'den 10'a kadar sayılacaktır.
 - $kare_sayi = sayac * sayac$
- Çıktılar: Döngünün her adımında **kare_sayi** yazdırılacak.

Programlama Örneği – 6

R Kaynak Kodu ve Ekran Çıktısı

```
➤ for (sayac in 1:10) {  
➤   kare_sayi=sayac*sayac  
➤   print(kare_sayi)  
➤ }
```

```
> source('kare_hesapla.R')  
[1] 1  
[1] 4  
[1] 9  
[1] 16  
[1] 25  
[1] 36  
[1] 49  
[1] 64  
[1] 81  
[1] 100
```

Programlama Örneđi – 7

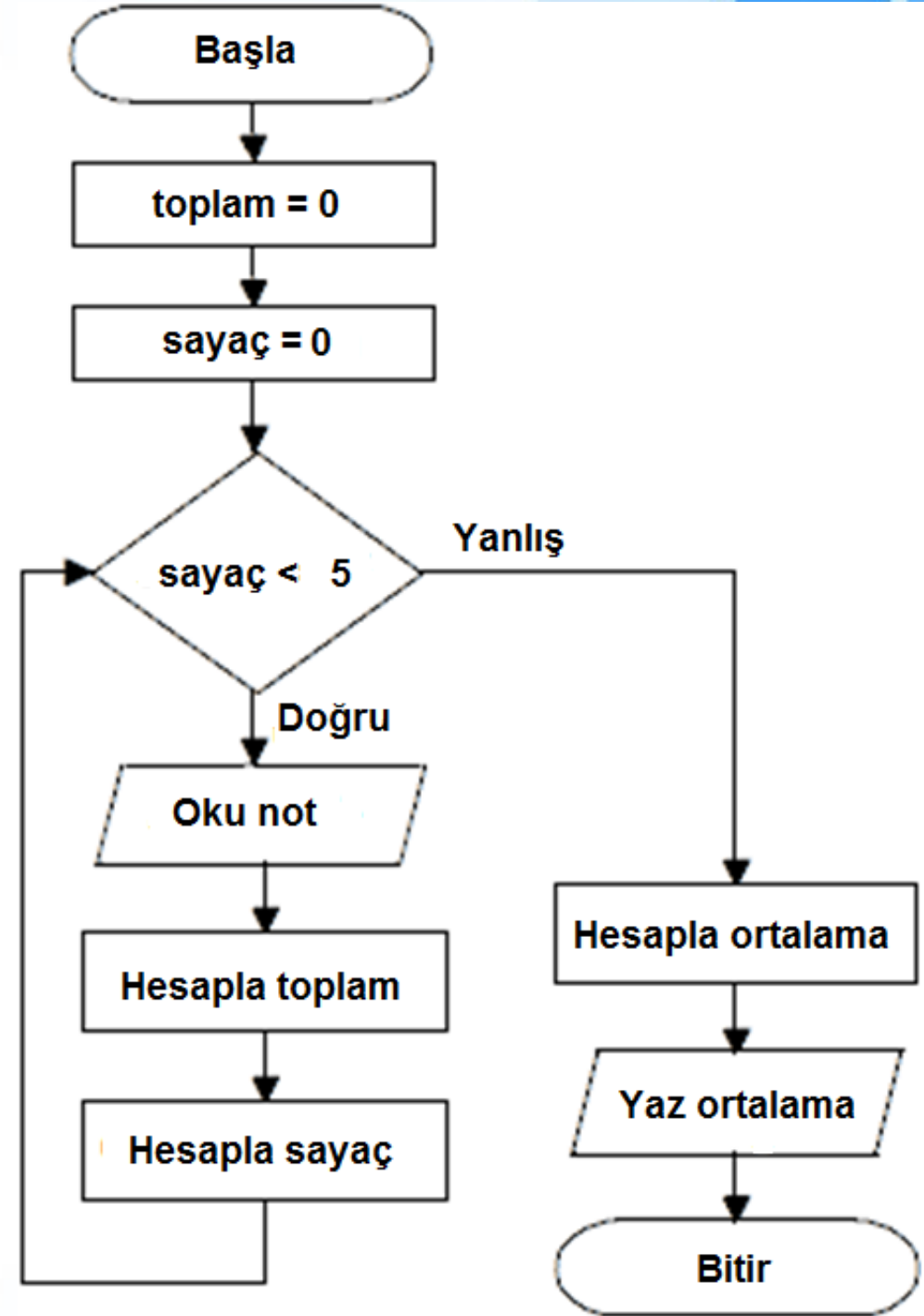
Döngü Kullanımı (Sayaç Kontrollü)

- 5 kişilik bir sınıftaki öğrencilerin notlarını okuyan ve bu sınıfın not ortalamasını hesaplayan bir program yazınız.
- **Programın analizi**
- Girdiler: 5 öğrenciye ait **not**
- İşlemler:
 - Notların toplamını hesapla. (**toplam**)
 - Öğrenci sayısını say (**sayaç kontrollü**)
 - Ortalamayı hesapla, **ortalama** = $\text{toplam} / \text{sayaç}$
- Çıktılar: **ortalama**

Programlama Örneği – 7

Sözde Kod ve Akış Şeması

- Başla
- toplam = 0
- sayaç = 0
- **Döngü** (sayaç < 5 olduğu sürece)
 - Oku not
 - Hesapla toplam = toplam + not
 - Hesapla sayaç = sayaç + 1
- **Döngüyü bitir**
- Hesapla ortalama = toplam / sayaç
- Yaz ortalama
- Bitir



Programlama Örneği – 7

R Kaynak Kodu

```
➤ toplam=0
➤ sayac=0
➤ while (sayac<5) {
➤ not=as.numeric(readline(prompt = "öğrencinin notu: "))
➤ toplam=toplam+not
➤ sayac=sayac+1
➤ }
➤ ortalama=toplam/sayac
➤ print(paste("Sınıf ortalaması:", ortalama))
```

Programlama Örneği – 7

Ekran Çıktısı

```
> source('sinif_ortalamasi.R')  
öğrencinin notu: 45  
öğrencinin notu: 60  
öğrencinin notu: 75  
öğrencinin notu: 100  
öğrencinin notu: 25  
[1] "Sınıf ortalaması: 61"
```

Programlama Örneđi – 6

Bilgisayar Hafızasındaki Deđişkenler

| | not | toplam | sayaç | ortalama |
|------------------|------------|---------------|--------------|-----------------|
| Döngü öncesi | | 0 | 0 | |
| Döngü sırasında | 45 | 45 | 1 | |
| | 60 | 105 | 2 | |
| | 75 | 180 | 3 | |
| | 100 | 280 | 4 | |
| | 25 | 305 | 5 | |
| Döngü sonrasında | | 305 | 5 | 61 |

Notların sırasıyla 45, 60, 75, 100 ve 25 olarak girildiđi düşünöldüğünde.



Dinlediđiniz için teŝekkürler.

Doç. Dr. Özgür ZEYDAN
<https://ozgurzeydan.com.tr/>

