






KONTROL YAPILARI- I (SEÇİMLİ YAPILAR)

Hedefler

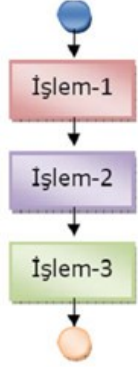
Bu üniteyi çalıştıktan sonra;

	Scanner deyimi ile klavyeden veri girişinin nasıl yapılacağını öğreneceksiniz,
	Diyalog kutuları (showInputDialog) ile veri girişinin nasıl yapılacağını öğreneceksiniz,
	Tek seçimli karar yapısını (if komutunun kullanımını) öğreneceksiniz,
	Çift seçimli karar yapılarını (if/else ve ? : komut yapılarını) öğreneceksiniz,
	Çok seçimli karar yapılarını (iç içe if/else ve switch case) öğreneceksiniz,

5. SEÇİMLİ - KARAR YAPILARI (SELECTION STRUCTURES)

- 5.1. Giriş
- 5.2. Scanner deyiminin kullanımı
- 5.3. Diyalog kutuları (**showInputDialog**, **showMessageDialog**)
- 5.4. Tek seçimli karar yapıları (if deyimi)
- 5.5. Çift seçimli karar yapıları (if/else deyimi)
- 5.6. ? : operatörü
- 5.7. Çok seçimli karar yapıları (iç içe if/else yapıları)
- 5.8. break deyimi
- 5.9. Switch case deyimi
- 5.10. Konunun Özeti
- 5.11. Çoktan Seçmeli Değerlendirme Soruları

GİRİŞ



Program akışının değişmediği sıralı yapıları saymaz isek seçimli ve tekrarlı yapılar olmak üzere iki farklı tipte kontrol yapısı ve komutları mevcuttur. Normalde yazılan bir programın işleyişi, ilk satırdan (işlem-1) son satıra doğru, sırası ile gerçekleşir. Fakat yapısal programlama dillerinde genelde bu işleyiş sırası, şartlı veya şartsız olarak değiştirilebilir.

Programın işleyiş yönünün şartlı veya şartsız olarak değiştirilmesi işlemine **dallanma** (branching) denir. Dallanma işlemini gerçekleştiren komutlara da

kontrol komutları denir.

Dikkat: Bazı programlama dillerinde (Visual Basic, C, C++ gibi) şartsız dallanma komutu (GOTO komutu) kullanılırken, Java dilinde ise kullanılmaz. Yani şartsız bir dallanma gerçekleşmez.

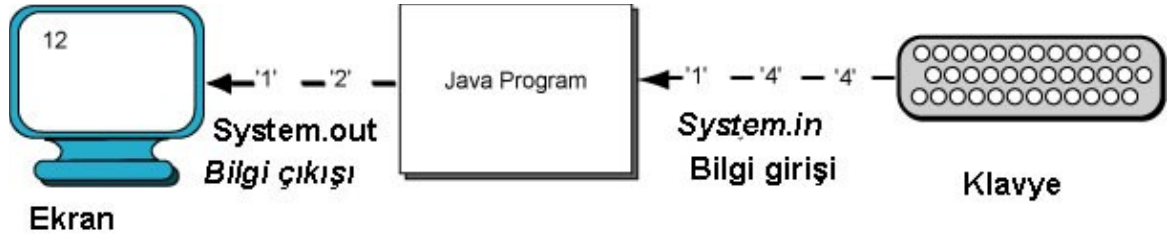
Şimdi seçimli yapıları anlatmadan önce konsol (giriş-çıkış) uygulamalarında kullanacağımız konsoldan veya inputbox'dan veri okuma komutlarını açıklayalım.

Scanner deyimi

Giriş – Çıkış işlemleri için **java.util.Scanner** sınıfının yöntemleri kullanılır. **Scanner**, Java 5.0 ile birlikte gelen yeni bir sınıftır. Daha önceki versiyonlarda (Java 1.4.0 gibi) bulunmaz. **Scanner** sınıfının yöntemlerini kullanabilmek için programın başında “**import java.util.Scanner**” veya “**import java.util.***” ifadesi kullanılır.

Bilgi giriş-çıkışı için genelde üç yöntem kullanılır. Bunlar;

- **System.in** – Klavyeden bilgi girişi için kullanılır
- **System.out** -- İşlem sonuçlarını, mesajları ekranda göstermek için kullanılır.
- **System.err** – Hata mesajlarını verir.



Klavyeden bilgi girişi yapıldıktan sonra o veri üzerinde işlem yaparken **Scanner** sınıfının bazı yöntemleri kullanılır. Bu yöntemler, Tablo1' de gösterilmiştir.

Tablo 1 Bazı Scanner Sınıf Yöntemleri

İşlevi	Yöntem Adı
Klavyeden girilen ifadeyi tamsayı (integer) türüne çevirir	nextInt()
Klavyeden girilen ifadeyi tamsayı (byte) türüne çevirir	nextByte()
Klavyeden girilen boolean türündeki önermeyi okur veya boolean türüne çevirir.	nextBoolean()
Klavyeden girilen ifadeyi kesirli(gerçek) sayı (float) türüne çevirir	nextFloat()
Klavyeden girilen ifadeyi uzun kesirli(gerçek) sayı (double) türüne çevirir	nextDouble()
Klavyeden girilen ifadenin ilk kelimesini okur {boşluk (space) karakterine kadar olan kısmı tarar}	next()
Eğer veri kaynağında veri varsa true yoksa false değerini döndürür. Boolean türüdür. Klavyeden girilen veriler üzerinde işlem yapmak için kullanılır.	hasNext()
Klavyeden girilen tüm satırı okur. String türünde bir satır okur.	nextLine()
Klavyeden girilen tek bir karakteri okur	tara.next().charAt(0)

Scanner deyiminin kullanım şekli;

```
Scanner taraD =new Scanner( System.in );
intDeger=taraDegisken.nextInt();
```

// Öncelikle Scanner sınıfından taraD isimli bir nesne oluşturuyoruz. Bu nesneyi kullanarak
 //klavyeden girilen tamsayıyı başka bir değişkene örneğimizde 'intDeger' isimli değişkene
 //aktarıyoruz.

INPUT (Veri Giriş) İletişim Kutusu

Kullanıcıdan bilgi almak için (String tipinde) kullanılan iletişim kutusudur. Standart kullanım şekli;

```
JOptionPane.showInputDialog(null, "mesajınız", " başlık", "simge");
```

İlk parametre **null** verildiğinde, küçük pencere ekranın ortasında gözükür. 3. ve 4. parametreler (“başlık” ve “simge”) seçimliktir, kullanılmazsa da olur. Ayrıca istenirse iletişim kutusu ilk ekranda gösterildiğinde varsayılan bir değer de gösterilebilir. Örnek tanımlama;

```
JOptionPane.showInputDialog(null, "mesajınız", "varsayılan değer");
```

Şeklinde yapılabilir.

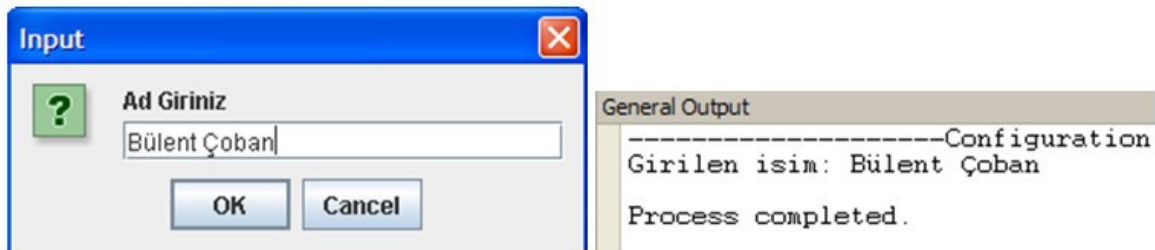
Ayrıca input iletişim kutusunu seçenek (option) iletişim kutusu gibi de kullanabilirsiniz. Yani input dialog iletişim kutusundan birden fazla seçim de yapılabilir. O zaman iletişim kutusunun kullanım şekli aşağıdaki gibi olur;

```
JOptionPane.showInputDialog(null, "Mesajınız", "başlık", "simge", null, "seçenekler", "varsayılan değer");
```

Örnek 1. İletişim kutusuna girilen ismi ekranda gösteren programı aşağıdaki gibi yazabiliriz.

```
import javax.swing.JOptionPane;
class MesajKutulari4 {
    public static void main(String[] args) {
        String ad = JOptionPane.showInputDialog (null,"Ad Giriniz");
        System.out.println("Girilen isim: "+ ad);
    }
}
```

İletişim kutusuna girilen ismi ve konsol / ekran çıktısını aşağıdaki görmekteyiz.



Mesaj İletişim Kutusu

Kullanıcıya bilgi/mesaj veren bir iletişim kutusudur. Bu tür pencerelerde mesaja ek olarak bir de, OK(Tamam) butonu bulunur. Kullanım Şekli;

```
JOptionPane.showMessageDialog(null, "mesajınız", " başlık", "simge");
```

İlk parametre **null** verildiğinde, küçük pencere ekranın ortasında gözükür. 3. ve 4. parametreler (“başlık” ve “simge”) seçimlidir, kullanılmasa da olur.

Örnek 2. Aşağıdaki programın ekran çıktısı ne olur?

```
import javax.swing.JOptionPane;
class Merhaba{
    public static void main(String[] args) {
        JOptionPane.showMessageDialog (null, "Merhaba Java Programcisi");
    }
}
```

Bu programın ekran çıktısı aşağıdaki gibi olur;



SEÇİMLİ - KARAR YAPILARI (SELECTION STRUCTURES)

Seçimli (selection) yapıda bir şarta (koşula) bağlı olarak bir ya da daha fazla seçeneklerden bir tanesine dallanma işlemi gerçekleştirilir. Şartlı dallanma işlemi sayesinde, program şarta bağlı olarak hangi işlemi yapacağına karar verir. Java dilinde üç adet seçme işlemi gerçekleştiren yapı vardır. Bunlar **if**, **if / else (?:)** ve **switch case** yapılarıdır.

Dikkat: Bu yapıların hepsine koşul kısmında karşılaştırma operatörlerinden { == , != , > , < , >= , <= } biri mutlaka kullanılmalıdır.

TEK SEÇİMLİ YAPI (if Deyimi)

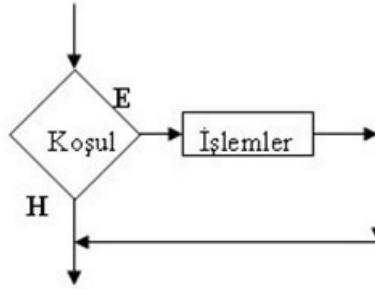
Koşula bağlı olarak tek bir işlemi yerine getiren yani şartlı dallanma işlemi gerçekleştiren yapıdır. Hemen hemen tüm programlama dillerinde olduğu gibi Java’da da bu işlem için **if** deyimi kullanılır. Tek seçimli yapılara örnek verirsek;

- “**Eğer** hava yağmurlu **ise** şemsiyeni al.”
- “**Eğer** yemek pişti **ise** ocağın altını kapat”
- “**Eğer** ortalaman 2.0 ve üzeri **ise** mezun oldun”

Şeklindeki ifadelerde bir koşula bağlı olarak tek bir işlem gerçekleşir. Koşul **doğru** (true) ise işlem yapılır. Koşul **yanlış** (false) ise herhangi bir işlem yapılmaz, program işleyişi bir alt satırdan devam eder.

Tek seçimli yapılarda if deyiminin kullanım şekli;

if (koşul) { < işlemler > }	if (koşul) < işlemler >
---	--



- Tek Seçimli Yapı (**if** deyimi) -

Dikkat: **if** yerine **IF** ya da **If** yazılırsa hata verir. Komutları yazarken büyük / küçük harf ayırımına dikkat ediniz.

Örnek 3. Dışarıdan girilen sayı negatif ise girilen sayının karesini alan programı yazınız.

Çözüm:

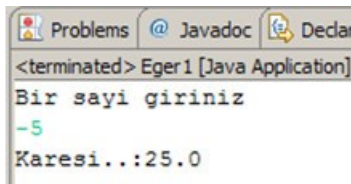
Algoritması;

1. **Başla**
2. *Gir bir sayı*
3. *Eğer sayı negatif (sayı<0) ise sayının karesini al*
4. *Yaz kare'yi*
5. **Dur**

JAVA Programlama Dilinde Kodlanması;

```
import java.util.Scanner;
class Eger1 {
public static void main(String[] args){
    Scanner tara =new Scanner( System.in );
    double sayi, kare=0;
    System.out.println ("Bir sayi giriniz");
    sayi=tara.nextInt();
    if (sayi<0) // negatifse
    {
        kare=sayi*sayi;
    }
    System.out.println("Karesi..:" +kare);
}}
```

Programın ekran çıktısı;



Sıra Sizde Sorusu:

Örnek 4. Aşağıdaki programın ekran çıktısı ne olur?

```
public class test1 {
    public static void main(String[] args) {
        int sonuc = 77;
        if (sonuc >= 80) System.out.print("A");
        if (sonuc >= 70) System.out.print("B");
        if (sonuc >= 60) System.out.print("C");
        else System.out.println("D");
    }
}
```

Programın ekran çıktısı:

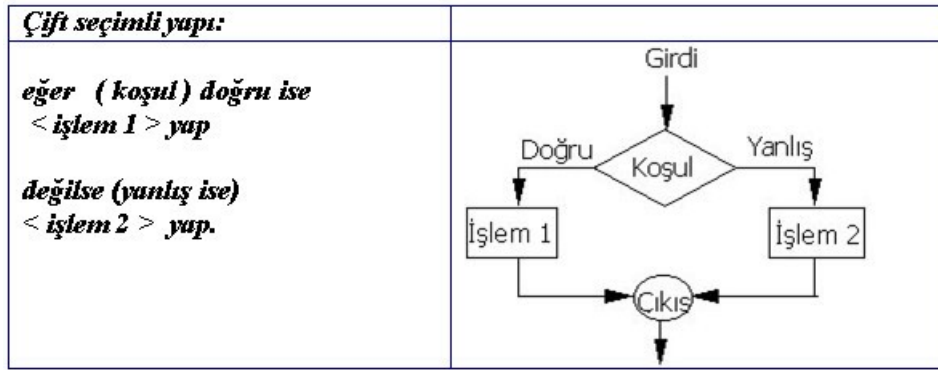
Programa dikkat edersek başlangıçta 'sonuc' isimli değişkene '77' değeri aktarılmıştır. '77' sayısı '70' ve '60' dan büyük olduğu için programın ekrana "BC" değerini yazacaktır.

ÇİFT SEÇİMLİ YAPI (if / else deyimi)

Koşula bağlı olarak iki işlem yerine getiren yani şartlı dallanma işlemini gerçekleştiren yapıdır. Tüm programlama dillerinde olduğu gibi Java'da da bu işlem için **if / else** deyimi kullanılır. Çift seçimli yapılara örnek verirsek;

- “**Eğer** (ortalamanız 2.0 ve üzeri) **ise** mezun oldunuz **değilse** mezun olamadınız.”
- “**Eğer** (hava yağmurlu) **ise** şemsiyeni al **değilse** alma.”

Şeklindeki ifadelerde bir koşula bağlı olarak iki işlem gerçekleşir. Koşul **doğru** (true) ise ilk işlem, **< işlem 1 >** , **yanlış** (false) ise else (değilse) den sonraki işlem, **< işlem 2 >** yapılır / gerçekleştirilir.



Şekil-2: Çift Seçimli Yapı

Çift seçimli yapılarda if deyiminin kullanım şekli;

<pre> if (koşul) { < işlem1 > } else { < işlem2 > } </pre>	<pre> if (koşul) < işlem 1 > else < işlem 2 > </pre>
--	--

Bu iki kullanım şekilleri arasındaki tek fark, işlemlerin { } süslü parantezler arasına alınıp alınmamasıdır.

? : Operatörü

Ayrıca çift seçimli yapı, if /else deyimi kullanılmadan sadece ?: operatörü ile gerçekleştirilebilir. Koşul doğru ise <işlem 1>, yanlışsa < işlem 2 > gerçekleştirilir. Kullanım şekli;

(Koşul) ? <işlem1> : <işlem2> ;

Örnek 5. Dışardan girilen nota göre öğrencinin o dersten geçip, geçmediğini ekranda gösteren programı yazınız.

Çözüm:

Algoritması;

1. **Başla**
2. Gir notu (not)
3. Eğer notunuz 50 ve üzeri ise
 Yaz "Geçtiniz"
 değilse
 Yaz "Kaldınız"
3. **Dur**

JAVA Programlama Dilinde Kodlanması;

a) if / else komutu ile;

```
import java.util.Scanner;
class EgerIse {
public static void main(String[] args){
    Scanner tara =new Scanner( System.in );
    int not;
    System.out.println ("Notunuzu giriniz");
    not=tara.nextInt();
    if (not>=50)
        System.out.println("Geçtiniz");
    else
        System.out.println("Kaldınız");
} }
```

b) ? : Operatörü ile;

Aynı programı if / else yapısını kullanmadan '?:' operatörü ile aşağıdaki gibi gerçekleştirebiliriz.

```
import java.util.Scanner;
class EgerIse {
public static void main(String[] args){
    Scanner tara =new Scanner( System.in );
    System.out.println ("Notunuzu giriniz");
    int not=tara.nextInt();
    System.out.println(not>=50 ? "Geçtiniz": "Kaldınız");
}}
```

Programın ekran çıktısı aşağıdaki gibidir;

```
<terminated> EgerIse [Java Application]
Notunuzu giriniz
55
Geçtiniz
```

```
<terminated> EgerIse [Java Application]
Notunuzu giriniz
45
Kaldınız
```

Örnek 6. Dışardan girilen sayının negatif mi, pozitif mi olduğunu ekranda gösteren programı en kısa kod satırı ile yazınız.

Çözüm:

Algoritması;

1. Başla
2. Gir bir sayı(sayı)
3. Eğer girilen sayı 0 (sıfır) dan büyükse
 Yaz "Sayı Pozitiftir"
 Değilse
 Yaz "Sayı Negatiftir"
4. Dur

JAVA Programlama Dilinde Kodlanması;

```
import java.util.Scanner;
class NP {
public static void main(String[] args){
    Scanner tara =new Scanner( System.in );
    System.out.println ("Bir sayı giriniz");
    int sayi=tara.nextInt();
    System.out.println(sayi>=0 ? "Sayı Pozitif": "Sayı Negatif");
}
}
```

Programın ekran çıktısı;

Sayı değeri olarak, -3 ve +3 girildiğinde programın ekran çıktısı aşağıdaki gibi olur.

<terminated> NP [Java Application] Bir sayı giriniz -3 Sayı Negatiftir	<terminated> NP [Java Application] Bir sayı giriniz 3 Sayı Pozitiftir
---	--

Örnek 7. Dışarıdan girilen sayının tek mi, çift mi olduğuna karar veren programı yazınız.

```
public class TekCift {
    public static void main(String[] args) {
        int sayi = 0;
        Scanner gir = new Scanner(System.in);
        System.out.println("Sayı gir");
        sayi = gir.nextInt();
        if (sayi % 2 == 0) { // Çift mi
            System.out.println("Çift sayı");
        } else {
            System.out.println("Tek sayı");
        }
    }
}
```

Programın ekran çıktısı;

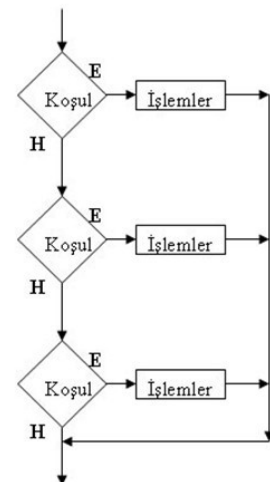
<terminated> TekCift [Java Application] Sayı gir 4 Çift sayı

Sıra Sizde:

Aynı programı ? : operatörlerini kullanarak yeniden yazınız

ÇOK SEÇİMLİ YAPI (İç İçe if ya da switch case yapısı)

Bu yapıda koşula veya koşullara bağlı olarak ikiden fazla seçim varsa **switch case** deyimi veya iç içe **if /else** yapıları kullanılır. Yani bu yapıda birden fazla karar yapısı iç içe kullanılır.



İç İçe İf / Else İf Yapısı (Nested If):

Daha önce belirttiğimiz gibi koşul sayısı ikiden fazla olduğunda kullanılacak yapılardan biridir. İç içe **if** yapısında kontrol edilmesi gereken koşul sayısının bir eksiği kadar **if** komutu kullanılır.

Kullanım Şekli;

```
if (Koşul 1)
    Yapılacak İşlemler 1;
else if (Koşul 2)
    Yapılacak İşlemler 2;
else if (Koşul 3)
    Yapılacak İşlemler 3;
...
...
else
    Yapılacak İşlemler N;
```

switch case yapısı:

Girilen değişkenin belirli değerlerine göre yapılacak farklı işlemlerin seçiminde kullanılır. **switch case** deyimi tek bir değişkenin değerine göre, sınırsız sayıda işlem yapma imkânı veren bir komuttur.

Dikkat: **switch** deyimi ile birlikte kullanılan değişken **char**, **byte**, **short**, veya **int** veri tipinde olabilir ama kesirli sayı değişkeni (float, double) olamaz.

Kullanım Şekli;

```
switch (değişken) {
    case d1:
        // Değişkenin d1 değeri aldığı durumda yapılacak işlemler
        break;
    case d2:
        // Değişkenin d2 değeri aldığı durumda yapılacak işlemler
        break;
    case d3:
        // Değişkenin d3 değeri aldığı durumda yapılacak işlemler
```

```
        break;

        .....

        .....

default:
    // Değişkenin yukarıdaki case kutularından farklı değerler aldığı durumlarda
    // yapılacak işlemler
}
```

Yukarıdaki yapıdan da anlaşılacağı üzere **switch** deyiminden sonra değişken adı gelir. Değişkenin değeri, hangi **case** deyimini izleyen sabitle çakışıyorsa, programın kontrolü o **case**'i izleyen kısma geçer. Eğer değişken değeri, hiçbir **case** bloğu ile çakışmıyorsa, programın akışı **default** ifadesini izleyen kısma geçer. **case ve default** ifadeleri **break** deyimi ile sona erdirilmelidir.

break deyimi;

break deyimi, etiketli(labeled) ve etiketsiz(unlabeled) olmak üzere iki farklı şekilde kullanılabilir. Etiketsiz break ifadesi ile koşul sağlandığında switch ifadesi sonlandırılır.

break deyimi herhangi bir döngü yapısından (**if, for, while** gibi) döngü bitmeden çıkma amacıyla da kullanılabilir. Fakat bu amaçla **break** komutu kullanılarak döngü dışına çıkılması yapısal programlama teknikleri açısından tavsiye edilmez. Döngü yapılarından, döngüden çıkışı sağlayacak koşul ile çıkılması tavsiye edilir.

Örnek 8. Girilen yaş değerine göre kişinin durumunu aşağıdaki kriterlere göre bulan programı yazınız.

Yaş	Durum
0... 2	Bebek
3... 12	Çocuk
13... 24	Yetişkin
25... 39	Genç
40... 54	Orta Yaşlı
55 ve üzeri	Yaşlı

Çözüm:

Algoritması;

1. Başla
2. Gir yaşı (yas)
3. Eğer Girilen yaş 0 ile 2 arasında ise Yaz "Bebek"
4. Değilse-Eğer 3 ile 12 arasında ise Yaz "Çocuk"
5. Değilse-Eğer 13 ile 24 arasında ise Yaz "Yetişkin"
6. Değilse-Eğer 25 ile 39 arasında ise Yaz "Genç"
7. Değilse-Eğer 40 ile 54 arasında ise Yaz "Orta Yaşlı"
- Değilse
Yaz " Yaşlı"
8. Dur

Algoritmasını oluşturduğumuz programı, iç içe **if** yapısını kullanarak Java programlama dilinde kodunu yazarsak;

İç içe if / else deyimi kullanılarak;

```
import java.util.Scanner;
class Yas{
public static void main(String[] args){
Scanner tara =new Scanner( System.in );
int yas;
System.out.println ("Yaşınızı giriniz");
yas=tara.nextInt();
    if (yas >= 0 && yas <= 2)
        System.out.println("Bebek");
    else if (yas > 2 && yas <= 12)
        System.out.println("Çocuk");
    else if (yas > 12 && yas <= 24)
        System.out.println("Yetişkin");
    else if (yas > 24 && yas <= 39)
        System.out.println("Genç");
    else if (yas > 39 && yas <= 54)
        System.out.println("Orta Yaşlı");
    else
        System.out.println("İhtiyar");
}}
```

Programın ekran çıktısı;

Yaş değeri olarak 54 girdiğimizde aşağıdaki ekran çıktısını elde ettik.

```
<terminated> Yas [Java Application]
Yaşınızı giriniz
54
Orta Yaşlı
```

Örnek 9. 100'lük sisteme göre girilen başarı notunu harfli sistemde gösteren programı yazınız.

Kriter: Girilen Not 90 ve üzeri ise AA, 80–89 arası ise BA, 70–79 arası ise BB, 60–69 arası ise CB, 50–59 arası ise CC, 50 den aşağı ise FF şeklinde gösterecek.

Çözüm:

Algoritması;

1. **Başla**
2. Başarı notunu giriniz (not)
3. Eğer notunuz 90 ve üzeri ise "AA"
4. Değilse eğer Notunuz 80 ile 89 arası ise "BA" yaz
5. Değilse eğer Notunuz 70 ile 79 arası ise "BB" yaz
6. Değilse eğer Notunuz 60 ile 69 arası ise "CB" yaz
7. Değilse eğer Notunuz 50 ile 59 arası ise "CC" yaz
8. Değilse eğer Notunuz 50 den küçükse "FF" yaz.
10. **Dur**

JAVA Programlama Dilinde Kodlanması:

Programı hem switch / case ile hem de iç içe if/else yapısını kullanarak yazalım.

a) **Switch / case deyimi ile;**

switch / case komutu ile sadece tek bir sabit değere göre işlem yapılabilirdiğinden 80-89 arasında "BA" sonucunu vermesi için girilen not değerini 10 'a bölerek (not/10) tam kısmını aldık.(83/10, 8 tamsayı sonucunu verdiği gibi 89/10 da 8 tamsayı sonucunu vermektedir)

```
import javax.swing.JOptionPane;
class BasariNotu {
    public static void main(String[] args) {
        int not = Integer.parseInt(JOptionPane
            .showInputDialog("Notunuzu giriniz : "));
        System.out.println("Harf Karşılığı..");
        switch (not / 10) {
            case 10:
            case 9:
                System.out.print("AA");
                break;
            case 8:
                System.out.print("BA");
                break;
            case 7:
                System.out.print("BB");
                break;
```

```

        case 6:
            System.out.print("CB");
            break;
        case 5:
            System.out.print("CC");
            break;

        default:
            System.out.print("FF");
            break;
    }
}
}

```

b). İç içe if Yapısını kullanarak;

Aynı programı iç içe if yapısı ile aşağıdaki gibi yazabiliriz.

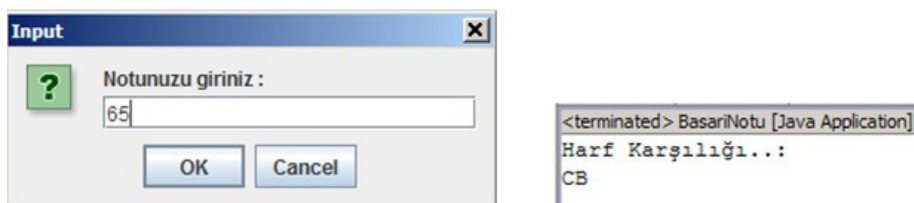
```

import javax.swing.JOptionPane;
class BasariNotu {
    public static void main(String[] args) {
        int not = Integer.parseInt(JOptionPane
            .showInputDialog("Notunuzu giriniz : "));
        System.out.println("Harf Karşılığı...");
        if (not >= 90 && not <= 100)
            System.out.println("AA");
        else if (not >= 80 && not <= 89)
            System.out.println("BA");
        else if (not >= 70 && not <= 79)
            System.out.println("BB");
        else if (not >= 60 && not <= 69)
            System.out.println("CB");
        else if (not >= 50 && not <= 59)
            System.out.println("CC");
        else
            System.out.println("FF");
    }
}

```

Programın ekran çıktısı;

Her iki programı da çalıştırdığımızda gelen iletişim kutusuna 65 girdiğimizde komut isteminde aşağıdaki mesajı aldık



Örnek 10. Girilen gün numarasına göre haftanın gününü ekranda gösteren programı yazınız.

Çözüm:

Algoritması;

1. Başla

2. Gün numarasını giriniz (1 ile 7 arasında)

3. Girilen numaranın gün karşılığını ekranda göster

{ 1: "Pazartesi"; 2: "Salı"; 3: "Çarşamba"; 4: "Perşembe";

5: "Cuma"; 6: "Cumartesi"; 7: "Pazar" }

4. Dur

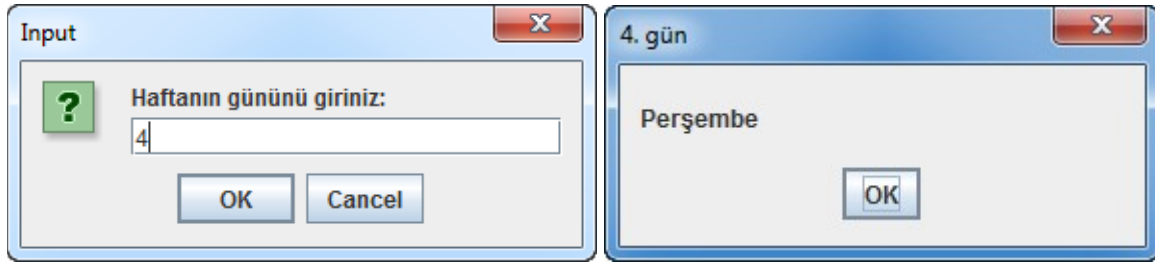
JAVA Programlama Dilinde Kodlanması;

```
import javax.swing.JOptionPane;

class H_Gunu {
    public static void main(String[] args) {
        int gun = Integer.parseInt(JOptionPane
            .showInputDialog("Haftanın gününü giriniz: "));
        String gunad = null;
        switch (gun) {
            case 1:
                gunad = "Pazartesi";
                break;
            case 2:
                gunad = "Salı";
                break;
            case 3:
                gunad = "Çarşamba";
                break;
            case 4:
                gunad = "Perşembe";
                break;
            case 5:
                gunad = "Cuma";
                break;
            case 6:
                gunad = "Cumartesi";
                break;
            case 7:
                gunad = "Pazar";
                break;
            default:
                gunad = "Yanlış gün numarası girdiniz";
                break;
        }
        JOptionPane.showMessageDialog(null, gunad, gun + ". gün",
            JOptionPane.PLAIN_MESSAGE);
    }
}
```

Programın ekran çıktısı;

Gün değeri olarak 4 girdiğimizde aşağıdaki ekran çıktılarını elde ettik.



Kendiniz Uygulayınız: Girilen ay numarasına göre mevsimi söyleyen programı yazınız.

Örnek 11. Klavyeden girilen üç sayı içerisinde en büyüğünü bulan programı yazınız.

Çözüm:

Evvvela şunu belirtelim, bir problemin birden fazla çözüm yolu olabilir. Atalarımız ne güzel demiş, “her yiğidin yoğurt yiyişi farklıdır” diye, aynen öyle de bir problemi farklı kişiler farklı şekillerde çözebilir veya bu örneğimizde olduğu gibi bir problemin birden fazla çözüm yöntemi olabilir.

Bir dizi sayı içerisinde en küçüğünü ya da en büyüğünü bulma problemlerinde, başlangıçta ilk sayının en küçük ya da en büyük (Enb) kabul edilmesi, karşılaştırma sayısını azaltmaktadır. Biz iki farklı algoritmayı da aşağıda gösterdik ve birini java dili ile programladık.

1. Çözüm yolu	2. Çözüm yolu
<ol style="list-style-type: none">1. Başla2. Gir üç sayı { s1, s2, s3 }3. İlk sayıyı en büyük kabul et {Enb=s1}4. Eğer $s2 > Enb$ İse $Enb=s2$ dir5. Eğer $s3 > Enb$ İse $Enb=s3$ dür6. Göster en büyük sayıyı (Enb)7. Dur	<ol style="list-style-type: none">1. Başla2. Gir üç sayı { s1, s2, s3 }3. Eğer $s1 > s2$ ve $s1 > s3$ ise en büyük s1 dir { Enb= s1}4. Eğer $s2 > s1$ ve $s2 > s3$ ise en büyük s2 dir { Enb= s2}5. Eğer $s3 > s1$ ve $s3 > s2$ ise en büyük s3 dür { Enb = s3}6. Göster en büyük sayıyı (Enb)7. Dur

1. Çözüm yolunun Java programlama dilinde kodlanması;

```
import java.util.Scanner;
public class EnBuyuk {
    public static void main(String[] args) {
        Scanner tara = new Scanner(System.in);
        int s1, s2, s3, Enb;
        System.out.println("Birinci sayı:");
        s1 = tara.nextInt();
        System.out.println("İkinci sayı:");
        s2 = tara.nextInt();
        System.out.println("Üçüncü sayı:");
        s3 = tara.nextInt();
        Enb = s1;
        if (s2 > Enb) {
            Enb = s2;
        }
        if (s3 > Enb) {
            Enb = s3;
        }
        System.out.println("En büyük sayı: " + Enb);
    }
}
```

Programın ekran çıktısı;

```
<terminated> EnBuyuk [Java Application]
Birinci sayı:
4
İkinci sayı:
8
Üçüncü sayı:
2
En büyük sayı: 8
```

Sıra Sizde:

Aynı programı ikinci algoritmaya (çözüm yoluna) göre siz yazınız.

Örnek 12. Dört işlem yapan basit bir hesap makinesi uygulamasını yapınız?

Çözüm:

Dr. Bülent ÇOBANOĞLU,

```
import java.util.Scanner;

class HesapMakinesi {
    public static void main(String[] args) {
        Scanner tara = new Scanner(System.in);
        int s1, s2;
        double sonuc = 0;
        char operator;
        System.out.println("Birinci sayı:");
        s1 = tara.nextInt();
        System.out.println("İkinci sayı:");
        s2 = tara.nextInt();
        System.out.println(" İşlem operatörünü giriniz.:");
        operator = tara.next().charAt(0);
        switch (operator) {
            case '+':
                sonuc = s1 + s2;
                break;
            case '-':
                sonuc = s1 - s2;
                break;
            case '*':
                sonuc = s1 * s2;
                break;
            case '/':
                sonuc = s1 / s2;
                break;
        }
        System.out.println("İşlem sonucu: " + sonuc);
    }
}
```

Programın ekran çıktısı:

Birinci sayı olarak 6, ikinci sayı olarak 4 ve işlem operatörü olarak * girildiğinde programın ekran çıktısı aşağıdaki gibi olur.

```
Birinci sayı:
6
İkinci sayı:
4
 İşlem operatörünü giriniz.:
*
İşlem sonucu: 24.0
```

Çoktan Seçmeli Değerlendirme Soruları

1. Aşağıdaki komut satırlarından hangisi ile karakter 'char' türünde veri okunabilir?
a) **next()** b) **nextByte()** c) **tara.next().charAt(0)** d) **nextInt()**
2. Klavyeden girilen tüm satırı okuyan Scanner komutu hangisidir?
a) **nextInt()** b) **nextByte()** c) **nextLine()** d) **next()**
3. Aşağıdaki programı çalıştırdığımızda "tip uyuşmazlığı" hatasını hangi kod satırında alırız?

```
public class test2 {  
    public static void main(String[] args) {  
        byte g=100;  
        switch (g) {  
            case 100: System.out.print("ALİ");  
                break;  
            case 120: System.out.print("OSMAN ");  
            case 140: System.out.print("ÖMER ");  
            default: System.out.print("EBUBEKİR ");  
        }  
    }  
}
```

- a) case 100: b) case 120: c) case 140: d) break;
4. Aşağıdaki programın ekran çıktısı ne olur?

```
public class Test {  
    public static void main(String[] args) {  
        int a = 5;  
        int b = 9;  
        int c = 0;  
        if (a > b) {  
            c = a;  
            a = b;  
            b = c;  
        }  
        System.out.println(b + "," + a);  
    }  
}
```

- a) 5,9 b) 9,5 c) 5,9,0 d) 0,5
5. Aşağıdakilerden hangisinde ekrana "Girilen not 50'den büyükse geçti değilse kaldı" mesajı yazılır?
a) `System.out.println (not>50 ? "Geçti": "Kaldı");`
b) `System.out.println (if (not>50) ; "Geçti": "Kaldı");`
c) `if (not<=50) printf "Geçti" else "Kaldı";`
d) `if (not>50) println "Geçti" else "Kaldı";`