

Nesne Yönelimli Programlama

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Not: Bu dersin sunumları, "Java Programlama Dili ve Yazılım Tasarımı, Altuğ B. Altıntaş, Papatya Yayıncılık, 2016" kitabı kullanılarak hazırlanmıştır.

Konular

- **Kontrol Deyimleri**
- **Döngü Deyimleri**
 - **while**
 - **do-while**
 - **for**
- **Karşılaştırma Deyimleri**
 - **if-else**
 - **switch**
- **Dallanma Deyimleri**
 - **break**
 - **continue**
 - **return**
 - **Varargs**

Kontrol Deyimleri

- Kontrol deyimleri bir uygulamanın yürütülmesi sırasında program akışını yönlendiren yapılardır.
- Java programlama dilinde farklı işlevlere sahip çok sayıda kontrol deyimi vardır.
- Döngü deyimleri:
`while, do-while, for`
- Karşılaştırma deyimleri:
`if-else, switch-case`
- Dallanma deyimleri:
`break, continue, label:, return`
- İstisna deyimleri:
`try-catch-finally, throw`

3

Konular

- Kontrol Deyimleri
- **Döngü Deyimleri**
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

Döngü Deyimleri

- Döngü deyimleri aynı işlemin tekrarı için kullanılır.
- Tekrarlı çalışmalar aynı veya farklı parametre değerleri üzerinde gerçekleştirilir.
- Java programlama dilinde `while`, `do-while` ve `for` olarak üç farklı döngü deyimi vardır.

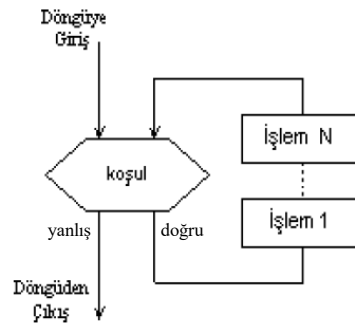
5

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

while

- **while** deyimi belirli bir grup kod öbeğini döngü koşulu doğru (true) olduğu sürece tekrarlar.



```
while (koşul)
{
    çalışması istenen kod bloğu
}
```

7

while

Örnek

```
package DonguOrnekler;

public class DonguOrnekler {

    public static void main(String [] args) {
        int i = 0 ;    //döngü kontrol değişkeni

        while (i < 5 ) {
            System.out.println("i = " + i);
            i++ ;
        }

        System.out.println("Sayma islemi tamamlandi.");
    }
}
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
Sayma islemi tamamlandi.
```

8

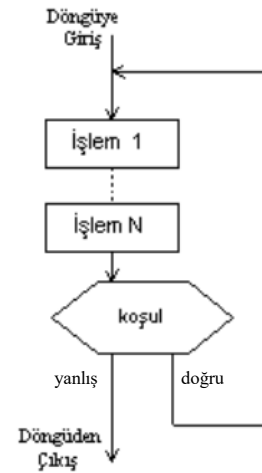
Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - **while**
 - **do-while**
 - **for**
- Karşılaştırma Deyimleri
 - **if-else**
 - **switch**
- Dallanma Deyimleri
 - **break**
 - **continue**
 - **return**
 - **Varargs**

do-while

- **do-while** deyimi belirli bir grup kod öbeğini döngü koşulu doğru (true) olduğu sürece tekrarlar.
- Döngü koşulu döngü sonunda yer alır.
- Döngü deyimleri en az bir kez çalıştırılır.

```
do
{
    çalışması istenen kod bloğu
} while (koşul);
```



do-while

Örnek

```
package DonguOrnekler;

public class DonguOrnekler {

    public static void main(String [] args) {
        int i = 0;        //döngü koşul değişkeni

        do {
            System.out.println("i = " + i);
            i++;
        } while ( i < 0 );

        System.out.println("Sayma islemi tamamlandi.");
    }
}
```

```
i = 0
Sayma islemi tamamlandi.
```

11

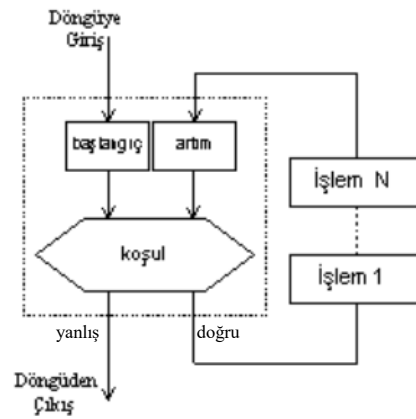
Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - while
 - do-while
 - for
- Karşılaştırma Deyimleri
 - if-else
 - switch
- Dallanma Deyimleri
 - break
 - continue
 - return
 - Varargs

for

- **for** deyiminde, koşulda kullanılan çevrim değişkeni, koşul ifadesi ve çevrim sayacı değişimi for ifadesi içerisinde verilir.
- **for** deyimi içerisinde ";" ile ayrılmış 3 kısım vardır:
 - çevrim sayacı
 - koşul ifadesi
 - sayaç değişimi

```
for (başlangıç; koşul; artış)
{
    çalışması istenen kod bloğu
}
```



13

for

Örnek

```
package DonguOrnekler;

public class DonguOrnekler {

    public static void main(String [] args) {

        for (int i=0; i < 5; i ++){
            System.out.println("i = " + i);
        }

    }

}
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
```

14

for

- **for** deyiminde, birden fazla parametre ile işlem yapılabilir.

```
package DonguOrnekler;

public class DonguOrnekler {

    public static void main(String [] args) {

        for ( int i = 0, j = 0 ; i < 20 ; i++, j++ ) {
            i *= j ;
            System.out.println("i = " + i + " j = " + j);
        }
    }
}
```

```
i = 0 j = 0
i = 1 j = 1
i = 4 j = 2
i = 15 j = 3
i = 64 j = 4
```

15

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - while
 - do-while
 - for
- Karşılaştırma Deyimleri
 - if-else
 - switch
- Dallanma Deyimleri
 - break
 - continue
 - return
 - Varargs

Karşılaştırma Deyimleri

- Karşılaştırma deyimleri belirli bir koşula göre farklı işlemler yaptırılacağı zaman kullanılır.
- Java'da **if-else** ve **switch-case** olmak üzere iki farklı karşılaştırma deyimi vardır.

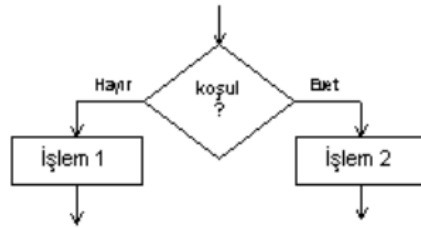
17

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - **while**
 - **do-while**
 - **for**
- Karşılaştırma Deyimleri
 - **if-else**
 - **switch**
- Dallanma Deyimleri
 - **break**
 - **continue**
 - **return**
 - **Varargs**

if-else

- Koşula göre program akışı değiştirilmek isteniyorsa if kullanılabilir.



```
if (koşul)
{
    durum true olduğunda çalışması istenen kod bloğu
} else
{
    durum false olduğunda çalışması istenen kod bloğu
}
```

19

if-else

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {
    public static void main(String [] args) {
        int puan = 76;
        char sonuc;

        if (puan >= 90) {
            sonuc = 'A';
        } else if (puan >= 80) {
            sonuc = 'B';
        } else if (puan >= 70) {
            sonuc = 'C';
        } else if (puan >= 60) {
            sonuc = 'D';
        } else {
            sonuc = 'F';
        }
        System.out.println("Sonuc = " + sonuc);
    }
}
```

Sonuc = C

20

if-else

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {
    public static void main(String [] args) {
        int puan = 80;
        char sonuc;

        sonuc = (puan >= 50? 'B': 'K');

        System.out.println("Sonuc = " + sonuc);
    }
}
```

Sonuc = B

21

if-else

Örnek

```
import java.util.Scanner;

public class KontrolOrnekler {
    public static boolean pozitif(int sayi) {
        return (sayi > 0);
    }

    public static boolean negatif(int sayi) {
        return (sayi < 0);
    }

    public static void main(String [] args) {
        Scanner klavye = new Scanner(System.in);
        System.out.print("Birinci sayıyı giriniz =");
        int sayi1 = klavye.nextInt();
        System.out.print("İkinci sayıyı giriniz =");
        int sayi2 = klavye.nextInt();

        System.out.println("*****");
        System.out.println("Başlangıç");

        if (pozitif(sayi1) && pozitif(sayi2)) {
            System.out.println("İkisi de pozitif");
        }
        else if (pozitif(sayi1) || pozitif(sayi2)) {
            System.out.println("İkisinden birisi pozitif");
        }
        else if (negatif(sayi1) && negatif(sayi2)) {
            System.out.println("İkisi de negatif");
        }
        else if (negatif(sayi1) || negatif(sayi2)) {
            System.out.println("İkisinden birisi negatif");
        }
        System.out.println("Son");
        System.out.println("*****");
    }
}
```

Birinci sayıyı giriniz =-10
İkinci sayıyı giriniz =-10

Başlangıç
İkisi de negatif
Son

22

if-else

Örnek

```
import java.util.Scanner;

public class KontrolOrnekler {
    public static boolean pozitif(int sayi) {
        return (sayi > 0);
    }

    public static boolean negatif(int sayi) {
        return (sayi < 0);
    }

    public static void main(String [] args) {
        Scanner klavye = new Scanner(System.in);
        System.out.print("Birinci sayıyı giriniz =");
        int sayi1 = klavye.nextInt();
        System.out.print("İkinci sayıyı giriniz =");
        int sayi2 = klavye.nextInt();

        System.out.println("*****");
        System.out.println("Başlangıç");

        if (pozitif(sayi1) && pozitif(sayi2)) {
            System.out.println("İkisi de pozitif");
        }
        else if (negatif(sayi1) && negatif(sayi2)) {
            System.out.println("İkisi de negatif");
        }
        else {
            System.out.println("İkisinden birisi pozitif");
        }
        System.out.println("Son");
        System.out.println("*****");
    }
}
```

```
Birinci sayıyı giriniz ==-10
İkinci sayıyı giriniz ==-10
*****
Başlangıç
İkisi de negatif
Son
*****
```

23

if-else

Örnek

```
import java.util.Scanner;

public class KontrolOrnekler {
    public static boolean pozitif(int sayi) {
        return (sayi > 0);
    }

    public static boolean negatif(int sayi) {
        return (sayi < 0);
    }

    public static void main(String [] args) {
        Scanner klavye = new Scanner(System.in);
        System.out.print("Birinci sayıyı giriniz =");
        int sayi1 = klavye.nextInt();
        System.out.print("İkinci sayıyı giriniz =");
        int sayi2 = klavye.nextInt();

        System.out.println("*****");
        System.out.println("Başlangıç");

        if (pozitif(sayi1) ^ pozitif(sayi2)) {
            System.out.println("Birisi pozitif birisi negatif");
        }
        else {
            System.out.println("İkisinin de işareti aynı");
        }
        System.out.println("Son");
        System.out.println("*****");
    }
}
```

```
Birinci sayıyı giriniz ==-10
İkinci sayıyı giriniz ==10
*****
Başlangıç
Birisi pozitif birisi negatif
Son
*****
```

```
Birinci sayıyı giriniz ==-10
İkinci sayıyı giriniz ==-10
*****
Başlangıç
İkisinin de işareti aynı
Son
*****
```

24

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

switch

- `switch` deyimi tamsayıların karşılaştırılması ile doğru koşulların elde edilmesini sağlayan mekanizmadır.

```
switch (tamsayı) {  
    case deger1: çalışması istenen kod bloğu;  
        break;  
    case deger2: çalışması istenen kod bloğu;  
        break;  
    case deger3: çalışması istenen kod bloğu;  
        break;  
    //...  
    default: çalışması istenen kod bloğu;  
}
```

switch

Örnek

```
public class KontrolOrnekler {  
    public static void main(String [] args) {  
        int ay = 8;  
        switch (ay) {  
            case 1: System.out.println("Ocak");  
                break;  
            case 2: System.out.println("Subat");  
                break;  
            case 3: System.out.println("Mart");  
                break;  
            case 4: System.out.println("Nisan");  
                break;  
            case 5: System.out.println("Mayis");  
                break;  
            case 6: System.out.println("Haziran");  
                break;  
            case 7: System.out.println("Temmuz");  
                break;  
            case 8: System.out.println("Agustos");  
                break;  
            case 9: System.out.println("Eylul");  
                break;  
            case 10: System.out.println("Ekim");  
                break;  
            case 11: System.out.println("Kasim");  
                break;  
            default: System.out.println("Aralik");  
        }  
    }  
}
```

Agustos

27

switch

Örnek

```
package KontrolOrnekler;  
  
public class KontrolOrnekler {  
    public static void main(String [] args) {  
        int ay = 8;  
        switch (ay) {  
            case 1: System.out.println("Ocak");  
            case 2: System.out.println("Subat");  
            case 3: System.out.println("Mart");  
            case 4: System.out.println("Nisan");  
            case 5: System.out.println("Mayis");  
            case 6: System.out.println("Haziran");  
            case 7: System.out.println("Temmuz");  
            case 8: System.out.println("Agustos");  
            case 9: System.out.println("Eylul");  
            case 10: System.out.println("Ekim");  
            case 11: System.out.println("Kasim");  
            default: System.out.println("Aralik");  
        }  
    }  
}
```

Agustos
Eylul
Ekim
Kasim
Aralik

28

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

Dallanma Deyimleri

- Dallanma deyimleri programın devam eden akışını atlama yaparak değiştirir.
- Java'da `break`, `continue` ve `return` olmak üzere üç farklı dallanma deyimi vardır.

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

break

- `break` deyimi içinde bulunduğu döngüyü tümüyle sonlandırarak çıkılmasını sağlar.
- İççe döngülerde `break` bulunduğu seviyedeki döngüyü sonlandırır.
- `break` deyiminin **etiketli (labeled)** ve **etiketsiz (unlabeled)** olarak iki türü vardır.
- **Etiketsiz** `break`, `switch` deyimindeki gibi kullanılır.
- **Etiketli** `break`, döngü deyimlerinde kullanılır ve bir etikete dallanma yapar.

break

Örnek - etiketsiz

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static void main(String [] args) {
        for ( int i = 0; i < 100; i++ ) {
            if ( i == 9 ) { // for döngüsünü kırıyor
                break;
            }
            System.out.println("i =" +i);
        }
        System.out.println("Donguden cikti");
    }
}
```

```
i =0
i =1
i =2
i =3
i =4
i =5
i =6
i =7
i =8
Donguden cikti
```

33

break

Örnek - etiketli

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static void main(String [] args) {

        kilil:
        for ( int i = 0; i < 100; i++ ) {
            if ( i == 6 ) { // for döngüsünü kırıyor
                break kilil;
            }
            System.out.println("i =" +i);
        }
        System.out.println("Donguden cikti");
    }
}
```

```
i =0
i =1
i =2
i =3
i =4
i =5
Donguden cikti
```

34

break

Örnek - etiketli

```
package Deneme;

public class Deneme {
    public static void main(String[] args) {
        int counter = 0;
        for (int i = 0; i <= 10; i++) {
            start:
            for (int j = 0; j <= 10; j++) {
                if (i == 5)
                    break start;
            }
            counter++;
        }
        System.out.println(counter);
    }
}
```

Son değer = 11

```
package Deneme;

public class Deneme {
    public static void main(String[] args) {
        int counter = 0;
        for (int i = 0; i <= 10; i++) {
            start:
            for (int j = 0; j <= 10; j++) {
                if (i == 5)
                    break start;
            }
            counter++;
        }
        System.out.println("Son değer = " + counter);
    }
}
```

Son değer = 5

35

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - while
 - do-while
 - for
- Karşılaştırma Deyimleri
 - if-else
 - switch
- Dallanma Deyimleri
 - break
 - continue
 - return
 - Varargs

continue

- **continue** deyiminin **etiketli (labeled)** ve **etiketsiz (unlabeled)** olarak iki türü vardır.
- Döngüde kullanılan **continue** deyimi devam etmekte olan iterasyonun kalan kısmını iptal eder ve bir sonraki iterasyona geçer.
- **Etiketsiz continue**, **switch** deyimindeki gibi kullanılır.
- **Etiketli continue**, döngü deyimlerinde kullanılır ve bir etikete dallanma yapar.

37

continue

Örnek - etiketsiz

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static void main(String [] args) {

        for ( int i = 0; i < 10; i++ ) {
            if ( i == 5 ) { // for döngüsünü kırıyor
                continue;
            }
            System.out.println("i =" + i);
        }
        System.out.println("Donguden cikti");
    }
}
```

```
i =0
i =1
i =2
i =3
i =4
i =6
i =7
i =8
i =9
Donguden cikti
```

38

continue

Örnek - etiketli

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static void main(String [] args) {
        pas:
        for (int j = 0 ; j < 5 ; j ++ ) {
            for (int i = 0; i < 5; i++) {
                if (i == 3) { // for döngüsünü kırıyor
                    continue pas;
                }
                System.out.println("j = " + j + ", i = " + i);
            }
            System.out.println("Donguden cikti");
            System.out.println("j = " + j);
        }
    }
}
```

```
j = 0, i = 0
j = 0, i = 1
j = 0, i = 2
j = 1, i = 0
j = 1, i = 1
j = 1, i = 2
j = 2, i = 0
j = 2, i = 1
j = 2, i = 2
j = 3, i = 0
j = 3, i = 1
j = 3, i = 2
j = 4, i = 0
j = 4, i = 1
j = 4, i = 2
```

39

continue

Örnek - etiketli

```
public class Deneme {
    public static void main(String[] args) {
        int counter = 0;
        for (int i = 0; i <= 3; i++) {
            start:
            for (int j = 0; j <= 3; j++) {
                if (i == 2)
                    continue start;
                System.out.println("i = " + i + ", " + "j = " + j);
            }
            counter++;
            System.out.println("Counter = " + counter);
        }
    }
}
```

```
i = 0, j = 0
i = 0, j = 1
i = 0, j = 2
i = 0, j = 3
Counter = 1
i = 1, j = 0
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
Counter = 2
i = 2, j = 0
i = 2, j = 1
i = 2, j = 2
i = 2, j = 3
Counter = 3
i = 3, j = 0
i = 3, j = 1
i = 3, j = 2
i = 3, j = 3
Counter = 4
```

```
package Deneme;

public class Deneme {
    public static void main(String[] args) {
        int counter = 0;
        start:
        for (int i = 0; i <= 3; i++) {
            for (int j = 0; j <= 3; j++) {
                if (i == 2)
                    continue start;
                System.out.println("i = " + i + ", " + "j = " + j);
            }
            counter++;
            System.out.println("Counter = " + counter);
        }
    }
}
```

```
i = 0, j = 0
i = 0, j = 1
i = 0, j = 2
i = 0, j = 3
Counter = 1
i = 1, j = 0
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
Counter = 2
i = 2, j = 0
i = 2, j = 1
i = 2, j = 2
i = 2, j = 3
Counter = 3
```

40

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - `while`
 - `do-while`
 - `for`
- Karşılaştırma Deyimleri
 - `if-else`
 - `switch`
- Dallanma Deyimleri
 - `break`
 - `continue`
 - `return`
 - `Varargs`

`return`

- `return` deyiminin 2 tür kullanım şekli vardır:
 - Yordamlardan üretilen değer döndürülür.
 - Yordamlardan herhangi bir noktadan `return` ile çıkılır.

return

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static double toplamaYap(double a, double b) {
        double sonuc = a + b ;
        return sonuc ; // normal return kullanımı
    }

    public static void birseyYapma(double a) {
        if (a == 0) {
            return ; // sıfır ise yordamı terk et
        } else {
            System.out.println("Gelen değer = " + a);
        }
    }

    public static void main(String [] args) {
        System.out.println("Toplama sonucu = " + toplamaYap(10.5, 11.5));
        birseyYapma(10.5);
    }
}
```

Toplama sonucu = 22.0
Gelen değer = 10.5

43

return

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static double toplamaYap(double a, double b) {
        double sonuc = a + b ;
        return sonuc ; // normal return kullanımı
    }

    public static void birseyYapma(double a) {
        if (a == 0) {
            return ; // sıfır ise yordamı terk et
        } else {
            System.out.println("Gelen değer = " + a);
        }
    }

    public static void main(String [] args) {
        System.out.println("Toplama sonucu = " + toplamaYap(10.5, 11.5));
        birseyYapma(0);
    }
}
```

Toplama sonucu = 22.0

44

Konular

- Kontrol Deyimleri
- Döngü Deyimleri
 - while
 - do-while
 - for
- Karşılaştırma Deyimleri
 - if-else
 - switch
- Dallanma Deyimleri
 - break
 - continue
 - return
 - **Varargs**

Varargs

- **Varargs** deyimi çok sayıda parametrenin sıralı ifade edilmesi için kullanılır.
- Bu değerlere döngü kullanarak erişilebilir.
- Fonksiyon sadece bir tane varargs alabilir.
- Fonksiyondaki varargs son değişken olmalıdır.

```
void method(String... a, int... b){} //Compile time error  
void method(int... a, String b){} //Compile time error
```

Varargs

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {

    public static void ekranaYaz(String... parametreler) {
        System.out.println("-----");
        for (String param: parametreler)
            System.out.println(" " + param);
    }

    public static void main(String [] args) {
        ekranaYaz("selamlar");
        ekranaYaz("merhaba", "nasılsın?");
        ekranaYaz("Bugün", "hava", "çok", "güzel");
    }
}
```

```
selamlar
-----
merhaba
nasılsın?
-----
Bugün
hava
çok
güzel
```

47

Varargs

Örnek

```
package KontrolOrnekler;

public class KontrolOrnekler {

    static void display(int num, String... values){
        System.out.println("number is "+num);
        for(String s:values){
            System.out.println(s);
        }
    }

    public static void main(String args[]){
        display(500,"hello");//one argument
        display(1000,"my","name","is","varargs");//four arguments
    }
}
```

```
number is 500
hello
number is 1000
my
name
is
varargs
```

48