

PART 8

Karar Verme

8.1 “if” yapısı

if komutu, bir veya daha çok komut tarafından takip edilen bir Boolean ifadeden oluşmaktadır.

```
if(Boolean_expression)
{
    //Statements will execute if the Boolean expression is true
}
```

Eğer boolean ifade true olarak değerlendirilirse, o halde if komutu içindeki kod bloğu çalıştırılacaktır.

Örnek:

```
public class Test {

    public static void main(String args[]){
        int x = 10;

        if( x < 20 ){
            System.out.print("This is if statement");
        }
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
This is if statement
```

8.2 “if...else” yapısı

Bir if komutunun ardından, Boolean ifade false olduğu zaman çalışan isteğe bağlı bir else komutu gelebilir.

```
if(Boolean_expression){
    //Executes when the Boolean expression is true
}else{
    //Executes when the Boolean expression is false
}
```

Örnek:

```
public class Test {

    public static void main(String args[]){
        int x = 30;
```

```
if( x < 20 ){
    System.out.print("This is if statement");
}else{
    System.out.print("This is else statement");
}
}
```

Bu aşağıdaki sonucu üretecektir:

```
This is else statement
```

8.3 “if--else if--else” yapısı

Bir if komutunun ardından, isteğe bağlı bir else if...else komutu gelebilir. Çeşitli koşulları test etmek için, tek bir if..else if komutu kullanmak çok yararlıdır.

If...else ‘in syntax’i:

```
if(Boolean_expression 1){
    //Executes when the Boolean expression 1 is true
}else if(Boolean_expression 2){
    //Executes when the Boolean expression 2 is true
}else if(Boolean_expression 3){
    //Executes when the Boolean expression 3 is true
}else {
    //Executes when the none of the above condition is true.
}
```

Örnek

```
public class Test {
    public static void main(String args[]){
        int x = 30;

        if( x == 10 ){
            System.out.print("Value of X is 10");
        }else if( x == 20 ){
            System.out.print("Value of X is 20");
        }else if( x == 30 ){
            System.out.print("Value of X is 30");
        }else{
            System.out.print("This is else statement");
        }
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
Value of X is 30
```

8.4 İç içe “if...else”

If-else komutlarını iç içe koymak her zaman geçerlidir, yani; bir if veya else if komutunu başka bir if veya else if komutu içinde kullanabilirsiniz.

```
if(Boolean_expression 1){  
    //Executes when the Boolean expression 1 is true  
    if(Boolean_expression 2){  
        //Executes when the Boolean expression 2 is true  
    }  
}
```

If komutunu içi içe koyduğumuz gibi, else if...else komutunu da iç içe koyabilirsiniz.

Örnek:

```
public class Test {  
  
    public static void main(String args[]){  
        int x = 30;  
        int y = 10;  
  
        if( x == 30 ){  
            if( y == 10 ){  
                System.out.print("X = 30 and Y = 10");  
            }  
        }  
    }  
}
```

Bu aşağıdaki sonucu üretecektir:

```
X = 30 and Y = 10
```

8.5 “switch” yapısı

switch ifadesi, bir değişkenin değerler listesi karşısında eşitliğinin test edilmesini sağlar. Her bir değer “case” olarak adlandırılır ve değişken her bir case ile denetlendikten sonra bir ötekine geçer.

```
switch(expression){  
    case value :  
        //Statements  
        break; //optional  
    case value :  
        //Statements  
        break; //optional  
    //You can have any number of case statements.  
    default : //Optional  
        //Statements  
}
```

switch komutu için aşağıdaki kurallar uygulanır.

- Switch komutunda kullanılan değişken sadece byte, short, int veya char türünde olabilir.
- Bir switch içinde herhangi sayıda case komutu olabilir. Her bir case'in ardından karşılaştırılacak değer ve iki nokta üst üste gelir.
- Case için seçilen değer, switch içindeki değişkenle aynı veri tipinde olmalıdır.
- Switch içindeki değer ile case değeri birbirini tutuyorsa, case komutunun ardından gelen ifadeler, break komutuna kadar çalıştırılacaktır.
- Break komutuna ulaşıldığında, switch sonlanır, ve kontrol akışı switch komutunun ardından gelen satıra atlar.
- Bir switch komutunun isteğe bağlı bir default case'i olabilir ve bu case switch'in sonunda kullanılmalıdır. Default case, hiç bir case'in true olmama durumunda görevi yerine getirmek için kullanılır. Default case içinde break kullanımı gerekmez.

Örnek

```
public class Test {  
  
    public static void main(String args[]){  
        char grade = args[0].charAt(0);  
  
        switch(grade)  
        {  
            case 'A' :  
                System.out.println("Excellent!");  
                break;  
            case 'B' :  
            case 'C' :  
                System.out.println("Well done");  
                break;  
            case 'D' :  
                System.out.println("You passed");  
            case 'F' :  
                System.out.println("Better try again");  
                break;  
            default :  
                System.out.println("Invalid grade");  
        }  
        System.out.println("Your grade is " + grade);  
    }  
}
```

Yukardaki programı, çeşitli satır argümanları kullanarak derleyip, çalıştırın. Bu aşağıdaki sonucu üretecektir:

```
$ java Test a  
Invalid grade  
Your grade is a a
```

```
$ java Test A
Excellent!
Your grade is a A
$ java Test C
Well done
Your grade is a C
$
```