

دستور if در جاوا

نحو دستور if در جاوا به شکل زیر است:

```
if (expression) {  
  // statements  
}
```

در اینجا expression یک عبارت از نوع boolean است (درست یا نادرست است).

اگر expression صحیح ارزیابی شود، دستورات داخل بدنه ی if (عبارات داخل پرانتز) اجرا می شوند.

اگر expression نادرست ارزیابی شود، دستورات داخل بدنه ی if اجرا نخواهند شد.

مثال ۱: دستور if در جاوا

```
1. class IfStatement {  
2.   public static void main(String[] args) {  
3.     int number = 10;  
4.     if (number > 0) {  
5.       System.out.println("Number is positive.");  
6.     }  
7.     System.out.println("This statement is always executed.");  
8.   }  
9. }
```

خروجی

```
Number is positive.
```

```
This statement is always executed.
```

وقتی number برابر با ۱۰ باشد، عبارت `number > 0` صحیح ارزیابی می شود. از این رو، کدهای داخل بدنه ی if اجرا می شوند.

حال مقدار number را به یک عدد صحیح منفی تغییر دهید. مثلاً ۵- . خروجی در این حالت برابر خواهد بود با:

```
This statement is always executed.
```

وقتی number برابر با ۵- است، عبارت `number > 0` نادرست ارزیابی می شود. از این رو، کامپایلر جاوا از اجرای دستورات بدنه ی if می گذرد.

دستور (if-else) در جاوا

در if در صورتی که شرط صحیح ارزیابی شود، بخش خاصی از کد اجرا می شود. دستور if ممکن است بلوک else هم داشته باشد. در صورتی که شرط نادرست ارزیابی شود، دستورات موجود در بدنه ی else اجرا می شوند.

نحو دستور if-else به شکل زیر است:

```
if (expression) {  
  // codes  
}  
else {  
  // some other code  
}
```

مثال ۲: دستور else ... if در جاوا

```
1. class IfElse {  
2.   public static void main(String[] args) {  
3.     int number = 10;  
4.  
5.     if (number > 0) {  
6.       System.out.println("Number is positive.");  
7.     }  
8.     else {  
9.       System.out.println("Number is not positive.");  
10.    }  
11.  
12.    System.out.println("This statement is always executed.");  
13.  }  
14. }
```

خروجی

Number is positive.

This statement is always executed.

وقتی number برابر با ۱۰ باشد، عبارت $number > 0$ به صورت صحیح ارزیابی می شود. در این حالت، کدهای داخل بدنه ی if اجرا می شوند و کدهای داخل بدنه ی else اجرا نخواهند شد.

حال مقدار number را به عدد منفی تغییر دهید. مثلاً ۵- . خروجی در این حالت برابر خواهد بود با:

Number is not positive.

This statement is always executed.

وقتی عدد ۵- است ، عبارت `number>0` به صورت نادرست ارزیابی می شود. در این حالت ، کدهای داخل بدنه ی `else` اجرا می شوند و کدهای داخل بدنه ی `if` اجرا نخواهند شد.

دستور `if ... else ... if` در جاوا

در جاوا ، اجرای یک قطعه کد از بین کدهای زیاد، امکان پذیر است. برای این کار می توانید از `if ... else` استفاده کنید.

```
if (expression1)
{
// codes
}
else if(expression2)
{
// codes
}
else if (expression3)
{
// codes
}
.
.
else
{
// codes
}
```

دستورات `if` از بالا به پایین اجرا می شوند. به محض درست بودن شرط ، کد مربوط به آن اجرا می شود.. پس از آن ، کنترل برنامه به خارج از `if ... else ... if` پرش می کند.

اگر همه عبارات تست نادرست باشند ، کدهای داخل بدنه ی `else` اجرا می شوند.

مثال ۳: دستور `if ... else ... if` در جاوا

```
1. class Ladder {
2.     public static void main(String[] args) {
3.         int number = 0;
4.
5.         if (number > 0) {
6.             System.out.println("Number is positive.");
7.         }
8.         else if (number < 0) {
9.             System.out.println("Number is negative.");
10.        }
```

```
11. else {
12. System.out.println("Number is 0.");
13. }
14. }
15. }
```

خروجی

Number is 0.

وقتی `number` برابر با ۰ باشد، هر دو شرط `number > 0` و `number < 0` به غلط ارزیابی می شوند. از این رو، کد درون بدنه ی `else` اجرا می شود.

برنامه بالا بررسی می کند که آیا عدد مثبت، منفی یا ۰ است.

دستور `if ... else` تو در تو در جاوا

امکان این که دستورات `if..else` تو در تو در جاوا داشته باشیم، وجود دارد.

در اینجا برنامه ای برای یافتن بزرگترین عدد از بین ۳ عدد وجود دارد:

مثال ۴: دستور `if ... else` تو در تو

```
1. class Number {
2. public static void main(String[] args) {
3. Double n1 = -1.0, n2 = 4.5, n3 = -5.3, largestNumber;
4. if (n1 >= n2) {
5. if (n1 >= n3) {
6. largestNumber = n1;
7. } else {
8. largestNumber = n3;
9. }
10. } else {
11. if (n2 >= n3) {
12. largestNumber = n2;
13. } else {
14. largestNumber = n3;
15. }
16. }
17. System.out.println("Largest number is " + largestNumber);
18. }
19. }
```

خروجی

Largest number is 4.5

نکته: در برنامه های بالا ، ما خودمان به متغیرها مقدار اختصاص داده ایم تا این امر ساده تر شود. اما ، در اپلیکیشن های دنیای واقعی ، این مقادیر ممکن است از داده های ورودی کاربر ، فایل های لاگ ، فرم های ارسالی و غیره حاصل شود.

تمرین:

۱- برنامه ای بنویسید که دو عدد را از ورودی دریافت کند و بزرگتر و کوچکتر بودن آنها را با پیام مناسب در خروجی نشان دهد.

۲- برنامه ای بنویسید که سه عدد از ورودی دریافت کند و مشخص کند که آیا این سه عدد اضلاع یک مثلث هست یا نه؟

۳- برنامه ای بنویسید که ضرایب یک معادله درجه ۲ را دریافت کند و ریشه های آن را در صورت وجود در خروجی نمایش دهد. (معادله درجه ۲ $Ax^2+Bx+C=0$)

۴- برنامه ای بنویسید که اضلاع یک مثلث را دریافت کند و مشخص کند مثلث قائم الزایه هست یا خیر؟

جواب سوال ۲

```
package javalike.Example;
import java.util.Scanner;
public class Mosalas {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int x, y, z;
        System.out. print ln("Enter x:");
        x = input.nextInt();
        System.out. print ln("Enter y:");
        y = input.nextInt();
        System.out. print ln("Enter z:");
        z = input.nextInt();
        if ((x + y > z) && (x + z > y) && (y + z > x))
            System.out. print ln("Yes");
        else
            System.out. print ln("No");
    }
}
```