

مقدار برگشتی از یک متد

متدها، می‌توانند مقدار برگشتی از هر نوع داده‌ای داشته باشند. این مقادیر می‌توانند در محاسبات یا به دست آوردن یک داده مورد استفاده قرار بگیرند. در زندگی روزمره فرض کنید که کارمند شما یک متد است و شما او را صدا می‌زنید و از او می‌خواهید که کار یک سند را به پایان برساند. سپس از او می‌خواهید که بعد از اتمام کارش سند را به شما تحویل دهد. سند همان مقدار برگشتی متد است. نکته مهم در مورد یک متد، مقدار برگشتی و نحوه استفاده شما از آن است. برگشت یک مقدار از یک متد آسان است. کافیت در تعریف متد به روش زیر عمل کنید :

```
returnType MethodName()
{
    return value;
}
```

returnType در اینجا نوع داده‌ای مقدار برگشتی را مشخص می‌کند (int, bool و ...). در داخل بدنه متد کلمه کلیدی return و بعد از آن یک مقدار یا عبارتی که نتیجه آن یک مقدار است را می‌نویسیم. نوع این مقدار برگشتی باید از انواع ساده بوده و در هنگام نامگذاری متد و قبل از نام متد ذکر شود. اگر متد ما مقدار برگشتی نداشته باشد باید از کلمه void قبل از نام متد استفاده کنیم. مثال زیر یک متد که دارای مقدار برگشتی است را نشان می‌دهد:

```
1 package myfirstprogram;
2
3 import java.text.MessageFormat;
4
5 public class MyFirstProgram
6 {
7     static int CalculateSum()
8     {
9         int firstNumber = 10;
10        int secondNumber = 5;
11
12        int sum = firstNumber + secondNumber;
13
14        return sum;
15    }
16
17    public static void main(String[] args)
18    {
19        int result = CalculateSum();
20
21        System.out.println(MessageFormat.format("Sum is {0}.", result));
22    }
23 }
```

```
Sum is 15.
```

همانطور که در خط 7 مثال فوق مشاهده می‌کنید هنگام تعریف متد از کلمه int به جای void استفاده کرده‌ایم که نشان دهنده آن است که متد ما دارای مقدار برگشتی از نوع اعداد صحیح است. در خطوط 9 و 10 دو متغیر تعریف و مقدار دهی شده‌اند.

توجه کنید که این متغیرها، متغیرهای محلی هستند. و این بدان معنی است که این متغیرها در سایر متدها مانند متد main() قابل دسترسی نیستند و فقط در متدی که در آن تعریف شده‌اند قابل استفاده هستند. در خط 12 جمع دو متغیر در متغیر sum قرار می‌گیرد. در خط 14 مقدار برگشتی sum توسط دستور return فراخوانی می‌شود. در داخل متد main() یک متغیره نام result در خط 19 تعریف می‌کنیم و متد CalculateSum() را فراخوانی می‌کنیم. متد CalculateSum() مقدار 15 را بر می‌گرداند که در داخل متغیر result ذخیره می‌شود. در خط 21 مقدار ذخیره شده در متغیر result چاپ می‌شود. متدی که در این مثال ذکر شد متد کاربردی و مفیدی نیست. با وجودیکه کدهای زیادی در متد بالا نوشته شده ولی همیشه مقدار برگشتی 15 است، در حالیکه می‌توانستیم به راحتی یک متغیر تعریف کرده و مقدار 15 را به آن اختصاص دهیم. این متد در صورتی کارآمد است که پارامترهایی به آن اضافه شود که در درسهای آینده توضیح خواهیم داد. هنگامی که می‌خواهیم در داخل یک متد از دستور if یا switch استفاده کنیم باید تمام کدها دارای مقدار برگشتی باشند. برای درک بهتر این مطلب به مثال زیر توجه کنید :

```
1 package myfirstprogram;
2
3 import java.util.Scanner;
4 import java.text.MessageFormat;
5
6 public class MyFirstProgram
7 {
8     static int GetNumber()
9     {
10        Scanner input = new Scanner(System.in);
11
12        int number;
13
14        System.out.print("Enter a number greater than 10: ");
15        number = input.nextInt();
16
17        if (number > 10)
18        {
19            return number;
20        }
21        else
22        {
23            return 0;
24        }
25    }
26
27    public static void main(String[] args)
28    {
29        int result = GetNumber();
30
31        System.out.println(MessageFormat.format("Result = {0}.", result));
32    }
33 }
```

```
Enter a number greater than 10: 11
Result = 11
```

```
Enter a number greater than 10: 9
Result = 0
```

در خطوط 25-8 یک متد با نام GetNumber() تعریف شده است که از کاربر یک عدد بزرگتر از 10 را می‌خواهد. اگر عدد وارد شده توسط کاربر درست نباشد متد مقدار صفر را بر می‌گرداند. و اگر قسمت else دستور if و یا دستور return را از آن حذف کنیم در هنگام اجرای برنامه با پیغام خطا مواجه می‌شویم.

چون اگر شرط دستور if نادرست باشد (کاربر مقداری کمتر از 10 را وارد کند) برنامه به قسمت else می‌رود تا مقدار صفر را بر گرداند و چون قسمت else حذف شده است برنامه با خطا مواجه می‌شود و همچنین اگر دستور return حذف شود چون برنامه نیاز به مقدار برگشتی دارد پیغام خطا می‌دهد. و آخرین

مطلبی که در این درس می‌خواهیم به شما آموزش دهیم این است که شما می‌توانید از یک متد که مقدار برگشتی ندارد خارج شوید.

حتی اگر از نوع داده‌ای void در یک متد استفاده می‌کنید باز هم می‌توانید کلمه کلیدی return را در آن به کار ببرید. استفاده از return باعث خروج از بدنه متد و اجرای کدهای بعد از آن می‌شود.

```
1 package myfirstprogram;
2
3 public class MyFirstProgram
4 {
5     static void TestReturnExit()
6     {
7         System.out.println("Line 1 inside the method TestReturnExit()");
8         System.out.println("Line 2 inside the method TestReturnExit()");
9
10        if(true)
11        {
12            return;
13        }
14
15        //The following lines will not execute
16        System.out.println("Line 3 inside the method TestReturnExit()");
17        System.out.println("Line 4 inside the method TestReturnExit()");
18    }
19
20    public static void main(String[] args)
21    {
22        TestReturnExit();
23        System.out.println("Hello World!");
24    }
25 }
```

```
Line 1 inside the method TestReturnExit()
Line 2 inside the method TestReturnExit()
Hello World!
```

در برنامه بالا نحوه خروج از متد با استفاده از کلمه کلیدی return و نادیده گرفتن همه کدهای بعد از این کلمه کلیدی نشان داده شده است. در پایان برنامه متد تعریف شده (TestReturnExit()) در داخل متد main() فراخوانی و اجرا می‌شود.