

به نام خدا

مبانی کامپیوتر و برنامه سازی

❖ حلقه های متداخل یا تودرتو:

یک یا چند حلقه می توانند داخل یک حلقه باشند. به این نوع از حلقه ها حلقه های تودرتو گفته می شود. این حلقه ها می توانند از یک نوع باشند یا متفاوت باشند. مثلاً " همگی for باشند یا یکی for، داخلی while و غیره باشند.

oo

مثال:

برنامه زیر جدول ضرب اعداد از یک تا نه را با استفاده از دو حلقه for تودرتو چاپ می کند.

```
1 public class MultiplicationTable {
2     /** Main method */
3     public static void main(String[] args) {
4         // Display the table heading
5         System.out.println("          Multiplication Table");
6
7         // Display the number title
8         System.out.print("          ");
9         for (int j = 1; j <= 9; j++)
10            System.out.print("    " + j);
11
12        System.out.println("\n-----");
13
14        // Display table body
15        for (int i = 1; i <= 9; i++) {
16            System.out.print(i + " | ");
17            for (int j = 1; j <= 9; j++) {
18                // Display the product and align properly
19                System.out.printf("%4d", i * j);
20            }
21            System.out.println();
22        }
23    }
24 }
```

Multiplication Table									
	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

مثال: برنامه‌ای بنویسید که n عدد صحیح از ورودی بخواند، برای هر عدد خوانده شده، تعداد مقسوم علیه‌های آن (به غیر از خود عدد) را چاپ کند. **6: 1, 2, 3 مقسوم علیه‌ها**

```
import java.util.Scanner;
public class Divisor {
    public static void main (String[] args) {
        Scanner input = new Scanner (System.in);
        System.out.print ("Enter n: ");
        int n = input.nextInt();

        for (int i = 1; i <= n ; i++) {

            int count = 0; // شمارنده تعداد مقسوم علیه ها
            System.out.print ("Enter x: ");
            int x = input.nextInt();
            int j = 1;

            while (j <= x/2){

                if (x % j ==0)
                    count++;

                j++;
            } // end while

            System.out.println("x =" + x + "\t count="+count);
        } // end for

    } // end main
} // end class
```

```
Enter n: 2
Enter x: 4
x =4   count=2
Enter x: 245
x =245   count=5
```

چاپ

مثال: برنامه‌ای بنویسید که n عدد صحیح از ورودی بخواند، فاکتوریل هر عدد خوانده شده را چاپ کند.

$$4! = 1*2*3*4 = 24$$

```
import java.util.Scanner;
public class Factorial {
    public static void main(String[] args) {
        Scanner input=new Scanner (System.in);
        System.out.print("Enter n: ");
        int n = input.nextInt();
        int i = 1;

        while (i <= n) {
            long fact = 1;
            System.out.print("Enter x: ");
            int x = input.nextInt();

            for (int j = 2 ; j <= x ; j++)
                fact *= j;

            System.out.println("x =" +x+" \t factorial="+fact);
            i++;
        } // end while

    } // end main
} // end class
```

```
Enter n: 3
Enter x: 4
x =4  factorial=24
Enter x: 12
x =12 factorial=479001600
Enter x: 34
x =34 factorial=4926277576697053184
```

چاپ

تمرین:

خروجی های چهار برنامه زیر را مشخص کنید:

```
public class Test {
    public static void main(String[] args) {
        for (int i = 1; i < 5; i++) {
            int j = 0;
            while (j < i) {
                System.out.print(j + " ");
                j++;
            }
        }
    }
}
```

(a)

```
public class Test {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            for (int j = i; j > 1; j--)
                System.out.print(j + " ");
            System.out.println("****");
            i++;
        }
    }
}
```

(b)

```
public class Test {
    public static void main(String[] args) {
        int i = 5;
        while (i >= 1) {
            int num = 1;
            for (int j = 1; j <= i; j++) {
                System.out.print(num + "xxx");
                num *= 2;
            }

            System.out.println();
            i--;
        }
    }
}
```

(c)

```
public class Test {
    public static void main(String[] args) {
        int i = 1;
        do {
            int num = 1;
            for (int j = 1; j <= i; j++) {
                System.out.print(num + "G");
                num += 2;
            }

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

(d)

OO

❖ دستورات break و continue در حلقه ها:

دستور break:

هنگامی که کنترل برنامه به دستور break در حلقه می‌رسد، کنترل برنامه از داخل حلقه خارج می‌شود و به دستور بعد از حلقه منتقل می‌گردد. یعنی حلقه زودتر خاتمه می‌یابد.



مثال: از دستور break

```
1 public class TestBreak {
2     public static void main(String[] args) {
3         int sum = 0;
4         int number = 0;
5
6         while (number < 20) {
7             number++;
8             sum += number;
9             if (sum >= 100)
10                break;
11        }
12
13        System.out.println("The number is " + number);
14        System.out.println("The sum is " + sum);
15    }
16 }
```

```
The number is 14
The sum is 105
```



دستور `continue`:

هنگامی که کنترل برنامه به دستور `continue` در حلقه می‌رسد، کنترل برنامه به انتهای حلقه منتقل می‌شود.

OO

مثال: از دستور `continue`

```
1 public class TestContinue {
2     public static void main(String[] args) {
3         int sum = 0;
4         int number = 0;
5
6         while (number < 20) {
7             number++;
8             if (number == 10 || number == 11)
9                 continue;
10            sum += number;
11        }
12
13        System.out.println("The sum is " + sum);
14    }
15 }
```

The sum is 189

OO