

دا ننگده مصندی و طوم کامنچیوقر

برنامه نویسی پیشرفته زمستان 1398 وحیدی اصل

دستورات كنترلي 2



- •مثالی از switch-case
 - •eurec for
 - •دستور while
 - •دستور do-while
- •دستور break-continue
 - •نوشتن چند برنامه





برنامه ای بنویسید که شماره ماه و سال خوشیدی را از کاربر دریافت کرده، تعداد روزهای ماه را در خروجی چاپ کند. کبیسه یا غیرکبیسه بودن سال را در نظر بگیرید.



تمرین برای switch-case

```
Import java.util.Scanner;
class MonthDays{
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        int month=in.nextInt();
        System.out.println("Please enter a month number:");
        int numDays = 0;
        switch (month) {
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
                numDays = 31;
                break;
            case 7:
            case 8:
            case 9:
            case 10:
            case 11:
                numDays = 30;
                break;
            case 12:
                if ( ((year % 4 == 0) && !(year % 100 == 0))
                     || (year % 400 == 0) )
                    numDays = 30;
                else
                    numDays = 29;
                break;
            default:
                System.out.println("Invalid month.");
                break;
        System.out.println("Number of Days = " + numDays);
```



•فرض کنید می خواهیم یک رشته (مثل <u>Welcome to</u>" را چاپ کنیم. (Java!"

•مسلماً کسالت بار است که بخواهیم دستور زیر را 100 بار تکرار کنیم:

System.out.println("Welcome to Java!");



برای حل این مشکل چه باید کرد؟



```
System.out.println("Welcome to Java!");
```

100 times



معلم خطاب به دانش آموز: 500 بار بنویس"من موشکهای کاغذی را در کلاس پرتاب نخواهم کرد!"

```
# Include (Stato.h)
int main(void)

{
  int count;
  for (count = 1; count <= 500; count++)
    printf ("I will not Throw paper dirplanes in class.");
  return 0;
}

MEAD 16-5
```





```
int count = 0;
while (count < 100) {
   System.out.println("Welcome to Java");
   count++;
}</pre>
```





```
while (loop-continuation-condition) {
 // loop-body;
 Statement(s);
                                        false
                             Continuation
                            Statement(s)
                             (loop body)
                               (A)
```

```
int count = 0;
 while (count < 100) {
  System.out.println("Welcome to Java!");
  count++;
          count = 0;
                      false
         (count < 100)?
          true
System.out.println("Welcome to Java!");
count++;
             (B)
```



Initialize count

```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!");
   count++;
}</pre>
```



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!");
   count++;
}</pre>
```



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!");
   count++;
}</pre>
```

Print Welcome to Java



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!")
   count++;
}</pre>
```

Increase count by 1 count is 1 now



```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}</pre>
```



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!");
   count++;
}</pre>
```

Print Welcome to Java



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!"),
   count++;
}</pre>
```

Increase count by 1 count is 2 now



```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}</pre>
```



```
int count = 0;
while (count < 2) {
   System.out.println("Welcome to Java!"):
   count++;
}</pre>
```

The loop exits. Execute the next statement after the loop.



•برنامه ای بنویسید که به طور تصادفی یک عدد صحیح بین 0 تا 100 تولید کند.

•برنامه به کاربر می گوید تا پشت سر هم اعدادی را وارد کند تا زمانی که عدد وارد شده با عدد تصادفی تولید شده برابر شود.

•به ازای هر ورودی کاربر، برنامه به او می گوید آیا مقدار وارد شده کاربر از عدد مربوطه کوچکتر است یا بزرگتر در نتیجه کاربر می تواند ورودی بعدی را هوشمندانه تر تولید کند.



```
import java.util.Scanner;
public class GuessNumber {
 public static void main(String[] args) {
  // Generate a random number to be guessed
  int number = (int)(Math.random() * 101);
  Scanner input = new Scanner(System.in);
  System.out.println("Guess a magic number between 0 and 100");
  int guess = -1;
  while (guess != number) {
   // Prompt the user to guess the number
   System.out.print("\nEnter your guess: ");
   guess = input.nextInt();
   if (guess == number)
    System.out.println("Yes, the number is " + number);
   else if (guess > number)
    System.out.println("Your guess is too high");
   else
    System.out.println("Your guess is too low");
  } // End of loop
```



ابزار یادگیری تفریق در هر اجرا تنها یک سوال مطرح می کرد.

با استفاده از loop می توانم سوالاتی را پشت سرهم طرح کنیم.

برنامه ای بنویسید که 5 سوال از کاربر بپرسد و تعداد پاسخهای صحیح را پس از پایان سوالات اعلام کند.

همچنین لیست سوالات به همراه صحیح یا غلط بودن پاسخ ارایه شود.



زمان سپری شده برای پاسخ دانش آموز نیز محاسبه شود.
• دستور زیر زمان فعلی را به میلی ثانیه بر می گرداند:

System.currentTimeMillis();

این متد در پکیج java.lang قرار دارد.



مسئله: ابزار پیشرفته آموزش ریاضی

```
import java.util.Scanner;
public class SubtractionQuizLoop {
 public static void main(String[] args) {
  final int NUMBER OF QUESTIONS = 5; // Number of
questions
  int correctCount = 0; // Count the number of correct answers
  int count = 0; // Count the number of questions
  long startTime = System.currentTimeMillis();
  String output = ""; // output string is initially empty
  Scanner input = new Scanner(System.in);
  while (count < NUMBER OF QUESTIONS) {
   // 1. Generate two random single-digit integers
   int number1 = (int)(Math.random() * 10);
   int number2 = (int)(Math.random() * 10);
   // 2. If number1 < number2, swap number1 with number2
   if (number1 < number2) {</pre>
    int temp = number 1;
    number1 = number2;
    number2 = temp;
```

```
// 3. Prompt the student to answer "What is number1 – number2?"
   System.out.print(
    "What is " + number 1 + " - " + number 2 + "?");
   int answer = input.nextInt();
  // 4. Grade the answer and display the result
   if (number1 - number2 == answer) {
    System.out.println("You are correct!");
    correctCount++;
  else
    System.out.println("Your answer is wrong.\n" + number1
     + " - " + number2 + " should be " + (number1 - number2));
  // Increase the count
  count++;
   output += "\n" + number1 + "-" + number2 + "=" + answer +
   ((number1 - number2 == answer)? "correct": "wrong");
 long endTime = System.currentTimeMillis();
 long testTime = endTime - startTime;
 System.out.println("Correct count is " + correctCount +
   "\nTest time is " + testTime / 1000 + " seconds\n" + output);
```



در بیشتر مواقع تعداد دفعات اجرای حلقه از قبل مشخص نمی باشد.

در چنین مواقعی از یک مقدار ورودی برای پایان دادن به حلقه استفاده می کنیم.

به این مقدار اصطلاحاً نگهبان (sentinel) گفته می شود.

حال برنامه ای بنویسید که مجموع تعداد نامشخصی عدد را محاسبه کند. مقدار 0 نگهبان یا پایان دهنده ورودی باشد.



پایان دادن به حلقه با یک مقدار کنترلی

```
import java.util.Scanner;
public class SentinelValue {
/** Main method */
public static void main(String[] args) {
  // Create a Scanner
  Scanner input = new Scanner(System.in);
  // Read an initial data
  System.out.print( "Enter an int value (the program exits if the input is 0): ");
  int data = input.nextInt();
  // Keep reading data until the input is 0
  int sum = 0;
  while (data != 0) {
   sum += data;
   // Read the next data
   System.out.print( "Enter an int value (the program exits if the input is 0): ");
   data = input.nextInt();
  System.out.println("The sum is " + sum);
```



حلقه های while تو در تو

•مى توان چندين حلقه while را به صورت تودرتو نوشت!

•برنامه ای بنویسید که یک هرم از * چاپ کند:

*

**



دقت کنبد

- •از مقادیر ممیز شناور برای بررسی مساوی بودن در شرط کنترلی حلقه استفاده نکنید.
 - •چون برای برخی مقادیر، تقریبی و نه دقیق هستند.
 - •در نتیجه ممکن است منجر به خطا و نتایج نادرست شوند.
 - •فرض کنید می خواهیم عبارت 0.1 + 0.1 + 0.1 + 0.0 + 0.0 + 1را محاسبه نماییم:

```
double item = 1; double sum = 0;
while (item != 0) { // No guarantee item will be 0
  sum += item;
  item -= 0.1;
}
System.out.println(sum);
```

متغیر item از 1 شروع به کم شدن 0.1 در هر تکرار حلقه می شود حلقه باید وقتی مقدار صفر می شود متوقف شود با اینحال، هیچ ضمانتی وجود ندارد که مقدار item دقیقاً 0 شود چون محاسبه ممیز شناور تقریبی است

در عمل ممكن است، حلقه بي نهايت بار تكرار شود!



Statement(s)

(loop body)



```
do {
    // Loop body;
    Statement(s);
} while (loop-continuation-condition);
```



حلقه do-while-مثال

برنامه ای بنویسید که بدون استفاده از آرایه و با استفاده از -0 while

```
int prevPrevVal = 0;
int prevVal = 1;
int currVal;
System.out.println(prevPrevVal);
System.out.println(prevVal);

do
{
    currVal = prevVal + prevPrevVal;
    System.out.println(currVal);

    prevPrevVal = prevVal;
    prevVal = currVal;
    prevVal = currVal;
} while (prevVal <= 100);</pre>
```



```
for (initial-action; loop-
    continuation-condition;
    action-after-each-iteration) {
  // loop body;
  Statement(s);
                          Initial-Action
                            Loop
                                        false
                          Continuation
                           Condition?
                          Statement(s)
                           (loop body)
                      Action-After-Each-Iteration
                             (A)
```

```
int i;
for (i = 0; i < 100; i++)
 System.out.println(
    "Welcome to Java!");
         i = 0
                    false
       (i < 100)?
  System.out.println(
   "Welcome to Java");
          i++
          (B)
```





Declare i

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println(
        "Welcome to Java!");
}</pre>
```





```
int i;
for (i = 0; i < 2; i++) {
    System.out.println(
    "Welcome to Java!");
}</pre>
```

Execute initializer i is now 0





```
int i;
for (i = 0; i < 2; i++) {
   System.out.println( "Welcome to Java!");
}</pre>
```

(i < 2) is true since i is 0





Print Welcome to Java

```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```





```
int i;

i now is 1
```

```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```





```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```

(i < 2) is still true since i is 1





Print Welcome to Java

```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```





```
Execute adjustment statement i now is 2
```

```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```





```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java!");
}</pre>
```

(i < 2) is false since i is 2





```
int i;
for (i = 0; i < 2; i++) {
   System.out.println("Welcome to Java")
}</pre>
```

Exit the loop. Execute the next statement after the loop



- •مقدار دهی اولیه حلقه for ممکن است حاوی لیستی از صفر یا بیشتر عبارت باشد که با کاما از هم جدا شده اند.
- •عملیات بعدی در داخل تعریف حلقه نیز می تواند شامل صفر یا بیشتر عبارت باشد که با کاما از یکدیگر جدا شده اند.
 - •دستورات زیر درست هستند. اما در عمل کمتر استفاده می شوند:

for (int i = 1; i < 100; System.out.println(i++));

```
for (int i = 0, j = 0; (i + j < 10); i++, j++) {

// Do something
```



تعریف چند متغیر در دستور for-مثال

```
1. /*
      Declare multiple variables in for loop Example
     This Java Example shows how to declare multiple variables in Java For loop using
     declaration block.
     */
 5.
 6.
     public class DeclaringMultipleVariablesInForLoopExample {
 8.
      public static void main(String[] args) {
10.
11.
12.
        * Multiple variables can be declared in declaration block of for loop.
13.
14.
15.
        for(int i=0, j=1, k=2; i<5; i++)
16.
         System.out.println("I:" + i + ",j:" + j + ", k:" + k);
17.
18.
         * Please note that the variables which are declared, should be of same type
19.
20.
         * as in this example int.
21.
22.
23.
        //THIS WILL NOT COMPILE
24.
        //for(int i=0, float j; i < 5; i++);
25.
26. }
```



•اگر شرط تکرار حلقه در داخل for حذف شود، به معنای آن است که شرط همواره درست بوده و یک حلقه بی نهایت ایجاد خواهد شد.

```
for (;;) {
   // Do something
}
Equivalent
// Do something
}

(a)

(b)
```



•اضافه نمودن ; بعد از دستور for پیش از آغاز بدنه حلقه یک خطای منطقی متداول است!

•برنامه لزوما دچار خطای نحوی یا زمان اجرا نمی شود. اما نتیجه مورد انتظار تولید نخواهد شد، زیرا بدنه حلقه فقط یک بار اجرا خواهد شدهٔ خواهد شمی للمی للمی للمی کواهد شمی کواهد کواه

```
for (int i=0; i<10; i++);
{
    System.out.println("i is " + i);
}</pre>
```



```
•به دلیل مشابه حلقه زیر نیز نادر ست است!
int i=0;
while (i < 10); ——Logic Error
System.out.println("i is " + i);
i++;
       •در مورد حلقه های do-while به ; در انتهای حلقه نیاز داریم.
int i=0;
do {
System.out.println("i is " + i);
i++;
} while (i<10);——— Correct
```



از كدام نوع حلقه استفاده كنيم؟

•سه ساختار حلقه بیان شده، do-while،while و for عملاً معادل یکدیگر هستند. •برای مثال، یک حلقه while در (a) در شکل زیر را همیشه می توان با for به صورت (b) نوشت.

```
while (loop-continuation-condition) {
    // Loop body
}

    for ( ; loop-continuation-condition; )
    // Loop body
}

    (a)
```

یک حلقه for در (a) در شکل زیر می تواند جز در موارد معدودی به فرم (b) نوشته شود.



- •از دستور حلقه ای استفاده کنید که استفاده از آن آسانتر و بدیهی تر است.
- •عموماً یک حلقه for در مواقعی استفاده می شود که تعداد تکرارها از قبل معلوم باشد.
 - •برای مثال بخواهید، 100 پیغام را چاپ کنید.
- •یک حلقه while معمولاً مواقعی استفاده می شود که تعداد تکرارها مشخص نیست و به مقدار نگهبان نیاز می باشد.
- •یک حلقه do-while در مواقعی به جای while استفاده می شود که بخواهیم بدنه حلقه را پیش از چک کردن شرط تکرار حلقه اجرا کنیم.



دستور break

- •استفاده از دستور break در داخل یک حلقه سبب می شود، به کل از حلقه خارج شویم.
 - •در مثال زیر پس از اینکه مقدار var برابر 3 شد، از حلقه خارج می شویم.
 - •بنابراین برنامه خروجی زیر را چاپ خواهد کرد:

0,1,2,3



دستور break-یک مثال دیگر

برنامه حدس اعداد را اینبار به کمک دستور break بنویسید!

```
import java.util.Scanner;
public class GuessNumberUsingBreak {
 public static void main(String[] args) {
  // Generate a random number to be guessed
  int number = (int)(Math.random() * 101);
  Scanner input = new Scanner(System.in);
  System.out.println("Guess a magic number between 0 and 100");
  while (true) {
   // Prompt the user to guess the number
   System.out.print("\nEnter your guess: ");
   int guess = input.nextInt();
   if (guess == number) {
    System.out.println("Yes, the number is " + number);
    break:
   else if (guess > number)
    System.out.println("Your guess is too high");
   else
    System.out.println("Your guess is too low");
  } // End of loop
```



مثال Break

```
public class TestBreak {
 public static void main(String[] args) {
  int sum = 0;
  int number = 0;
  while (number < 20) {
   number++;
   sum += number;
   if (sum >= 100)
    break;
  System.out.println("The number is " + number);
  System.out.println("The sum is " + sum);
```



دستور continue

- •استفاده از دستور continue در داخل یک حلقه سبب می شود سایر دستورات حلقه در آن تکرار مشخص اجرا نشوند و تکرا بعدی آغاز شود.
- •در مثال زیر پس از اینکه مقدار var برابر 2 شد، دستور println اجرا نشده و تکرار بعدی 3 آغاز می شود.
 - •بنابراین برنامه خروجی زیر را چاپ خواهد کرد:

•0,1,3,4

```
for(int var1 = 0; var1 < 5; var1++)
 2.
 3.
          for(int var2=0 ; var2 < 5 ; var2++)
 6.
               if(var2 == 2)
                    continue;
 7.
 8.
 9.
                    System.out.println("var1:" + var1 + ", var2:"+ var2);
10.
          }
11.
12.
13. }
```



مثالcontinue

```
public class TestContinue {
 public static void main(String[] args) {
  int sum = 0;
 int number = 0;
  while (number < 20) {
   number++;
   if (number == 10 | number == 11) continue;
   sum += number;
 System.out.println("The sum is " + sum);
```





مسئله: برنامه ای بنویسید که از حلقه های تودرتو برای چاپ جدول ضرب استفاده کند.



```
public class MultiplicationTable {
 /** Main method */
 public static void main(String[] args) {
 // Display the table heading
 System.out.println("
                           Multiplication Table");
 // Display the number title
 System.out.print(" ");
 for (int j = 1; j \le 9; j++)
   System.out.print(" "+j);
 System.out.println("\n-----");
 // Print table body
  for (int i = 1; i \le 9; i++) {
   System.out.print(i + " | ");
   for (int j = 1; j \le 9; j++) {
    // Display the product and align properly
    System.out.printf("%4d", i * j);
   System.out.println();
```





مسئله: برنامه ای بنویسید که از کاربر دو مقدار صحیح مثبت گرفته و بزرگترین مقسوم الیه مشترک انها را محاسبه نماید.



تمرین: یافتن بزرگترین مقسوم الیه مشترک دو عدد

```
import java.util.Scanner;
public class GreatestCommonDivisor {
 /** Main method */
 public static void main(String[] args) {
  // Create a Scanner
  Scanner input = new Scanner(System.in);
  // Prompt the user to enter two integers
  System.out.print("Enter first integer: ");
  int n1 = input.nextInt();
  System.out.print("Enter second integer: ");
  int n2 = input.nextInt();
  int gcd = 1;
  int k = 2;
  while (k \le n1 \&\& k \le n2)
   if (n1 \% k == 0 \&\& n2 \% k == 0)
    gcd = k;
   k++;
  System.out.println("The greatest common divisor for " + n1 +
    and " + n2 + " is " + gcd);
```



مسئله: پیش بینی میزان افزایش در آمد

مسئله: فرض کنید در آمد کار مندی امسال 10,000 است و افز ایش حقوق هر سال 7%می باشد. حساب کنید در چند سال حقوق کار مند دوبر ابر اولین سال خواهد شد؟



مسئله: پیش بینی میزان افزایش در آمد

```
double tuition = 10000; int year = 1 // Year 1
tuition = tuition * 1.07; year++; // Year 2
tuition = tuition * 1.07; year++; // Year 3
tuition = tuition * 1.07; year++; // Year 4
...
```



```
public class FutureTuition {
 public static void main(String[] args) {
  double tuition = 10000; // Year 1
  int year = 1;
  while (tuition \leq 20000) {
    tuition = tuition * 1.07;
    year++;
  System.out.println("Tuition will be doubled in "+ year + " years");
```