



دانشگاه مهندسی و علوم کامپیوتر

برنامه نویسی پیشرفته وحیدی اصل

دستورات کنترلی ۱

- در مثال محاسبه مساحت دایره اگر کاربر یک مقدار منفی را برای شعاع وارد کند، برنامه یک نتیجه نادرست تولید خواهد کرد.
- بنابراین اگر مقدار شعاع منفی باشد، به سازوکاری نیاز داریم که اجازه ندهد مساحت محاسبه شود.
- این سازو کار چه باید باشد؟

- نوع بولین
- عملگرهای بولین
- دستور ساده if
- دستور if-else
- دستور switch-case
- تقدم و شرکت پذیری عملگرها
- نوشتن چند برنامه

• در بیشتر مواقع در یک برنامه می خواهیم دو مقدار را با هم مقایسه کنیم: مثلاً چک کنیم که آیا a بزرگتر از b است یا خیر؟

• جاوا به این منظور شش عملگر مقایسه ای را در اختیار ما می گذارد که به آنها عملگرهای رابطه ای نیز گفته می شود که با استفاده از آنها می توانیم دو مقدار را با هم مقایسه کنیم.

• حاصل این مقایسه یک مقدار بولین خواهد بود: `true` (درست) یا `false` (نادرست)

```
boolean b = (1 > 2);
```

عملگر	نام
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

• این برنامه برای آزمون ریاضی یک دانش آموز کلاس اولی نوشته می شود.

• این برنامه دو عدد تک رقمی `number1` و `number2` (مثلاً ۷ و ۹) را به طور تصادفی تولید کرده و پرسش زیر را مطرح می کند:

• “What is $7 + 9$?”

• پس از اینکه دانش آموز پاسخ خود را وارد می کند، پاسخ وارد شده با پاسخ واقعی مقایسه شده و پیغامی مبنی بر `true` یا `false` بودن پاسخ دانش آموز چاپ می کند.

• برای تولید اعداد تصادفی از تابع زیر استفاده کنید:

```
int num1= (int) (Math.random()*10); //returns a number from 0 to 0.9
```

```
import java.util.Scanner;

public class AdditionQuiz {
    public static void main(String[] args) {
        int number1 = (int) (Math.random()*10);
        int number2 = (int) (Math.random()*10);

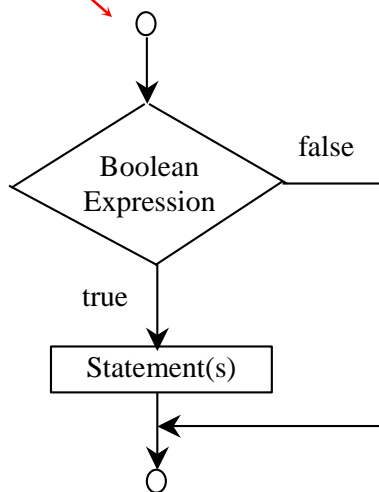
        // Create a Scanner
        Scanner input = new Scanner(System.in);

        System.out.print( "What is " + number1 + " + " + number2 + "? ");

        int answer = input.nextInt();

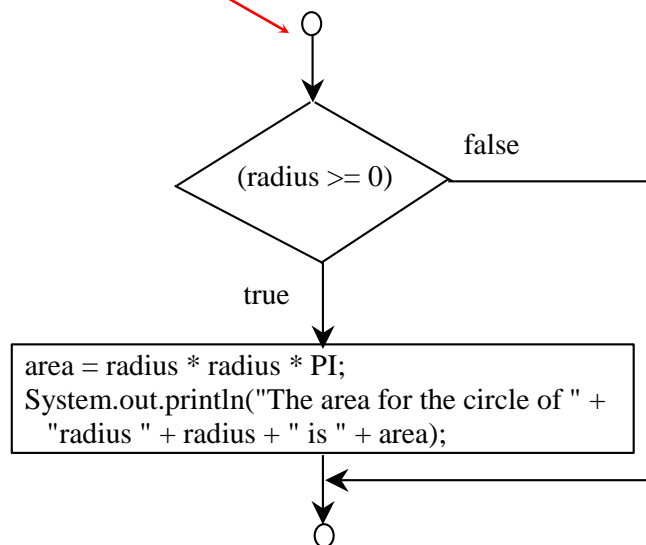
        System.out.println( number1 + " + " + number2 + " = " + answer + " is " + (number1 + number2 == answer));
    }
}
```

```
if (boolean-expression) {
    statement(s);
}
```



(A)

```
if (radius >= 0) {
    area = radius * radius * PI;
    System.out.println("The area"
        + " for the circle of radius "
        + radius + " is " + area);
}
```



(B)

برنامه ای بنویسید که یک عدد صحیح را از کاربر بگیرد. اگر عدد مضرب ۵ بود پیام HiFive و اگر مضرب ۲ بود پیام HiEven چاپ کند.

```
import java.util.Scanner;

public class SimpleIfDemo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter an integer: ");
        int number = input.nextInt();

        if (number % 5 == 0)
            System.out.println("HiFive");

        if (number % 2 == 0)
            System.out.println("HiEven");
    }
}
```

```
if i > 0 {
    System.out.println("i is positive");
}
```

(a) Wrong

```
if (i > 0) {
    System.out.println("i is positive");
}
```

(b) Correct

```
if (i > 0) {
    System.out.println("i is positive");
}
```

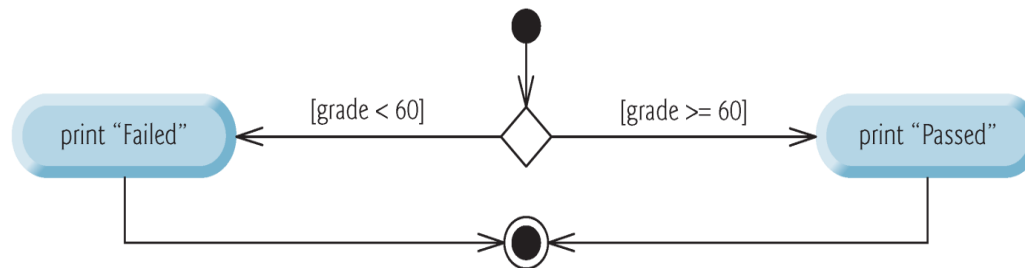
(a)

Equivalent

```
if (i > 0)
    System.out.println("i is positive");
```

(b)

```
if (boolean-expression) {
    statement(s) -for-the-true-case;
}
else {
    statement(s) -for-the-false-case;
}
```



مثال if...else

```

if (radius >= 0) {
    area = radius * radius * 3.14159;

    System.out.println("The area for the "
        + "circle of radius " + radius +
        " is " + area);
}
else {
    System.out.println("Negative input");
}
    
```

```
if (score >= 90.0)
    grade = 'A';
else
    if (score >= 80.0)
        grade = 'B';
    else
        if (score >= 70.0)
            grade = 'C';
        else
            if (score >= 60.0)
                grade = 'D';
            else
                grade = 'F';
```

Equivalent

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

Suppose score is 70.0

The condition is false

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

Suppose score is 70.0

The condition is false

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

Suppose score is 70.0

The condition is true

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```


Suppose score is 70.0

grade is C

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

Suppose score is 70.0

Exit the if statement

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

بخش **else** با نزدیکترین دستور **if** در یک بلاک تطبیق داده می شود.

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

(a)

Equivalent

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

(b)

• در برنامه زیر عبارتی چاپ نمی شود، مگر اینکه از {} استفاده کنیم.

```
int i = 1;
int j = 2;
int k = 3;
if (i > j)
    if (i > k)
        System.out.println("A");

else
    System.out.println("B");
```

• قراردادن یک ; در انتهای دستور if خطا به حساب می آید.

```
if (radius >= 0); ← Wrong
{
    area = radius*radius*PI;
    System.out.println(
        "The area for the circle of radius " + radius + " is " + area);
}
```

یافتن این خطا دشوار است، چون یک اشتباه نحوی یا خطای زمان اجرا به حساب نمی آید.

```
if (number % 2 == 0)
    even = true;
else
    even = false;
```

(a)

Equivalent

```
boolean even
    = number % 2 == 0;
```

(b)

```
if (even == true)
    System.out.println(
        "It is even.");
```

(a)

Equivalent

```
if (even)
    System.out.println(
        "It is even.");
```

(b)

ابزاری بنویسید که به دانش آموز کلاس اول دبستان تفریق ساده را بیاموزد.
برنامه دو رقم number1 و number2 را به طور تصادفی تولید می کند.
و از دانش آموز با فرض اینکه number1 > number2 سوال زیر را می پرسد:

“What is 9 – 2?”

سپس دانش آموز حاصل را تایپ کرده و در صورت صحیح بودن پاسخ، پیغام
می دهد: “You are correct”

در غیر اینصورت می گوید Your answer is wrong

و در ادامه می گوید پاسخ درست برای تفریق دو عدد number1 و number2 چیست؟


```
import java.util.Scanner;

public class SubtractionQuiz {
    public static void main(String[] args) {
        // 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) {
            int temp = number1;
            number1 = number2;
            number2 = temp;
        }

        // 3. Prompt the student to answer "what is number1 - number2?"
        System.out.print
            ("What is " + number1 + " - " + number2 + "? ");
        Scanner input = new Scanner(System.in);
        int answer = input.nextInt();

        // 4. Grade the answer and display the result
        if (number1 - number2 == answer)
            System.out.println("You are correct!");
        else
            System.out.println("Your answer is wrong.\n" + number1 + " - "
                + number2 + " should be " + (number1 - number2));
    }
}
```

نام	عملگر
not	!
and	& &
or	
exclusive or	^

p	!p
---	----

true	false
------	-------

false	true
-------	------

Example (assume age = 24, gender = 'M')

!(age > 18) is false, because (age > 18) is true.

!(gender != 'F') is false, because (gender != 'F') is true.

p	!p
---	----

true	false
------	-------

false	true
-------	------

Example

!(1 > 2) is true, because (1 > 2) is false.

!(1 > 0) is false, because (1 > 0) is true.

p1	p2	p1 && p2	Example (assume age = 24, gender = 'F')
false	false	false	<u>(age > 18) && (gender == 'F')</u> is true, because <u>(age > 18)</u> and <u>(gender == 'F')</u> are both true.
false	true	false	
true	false	false	<u>(age > 18) && (gender != 'F')</u> is false, because <u>(gender != 'F')</u> is false.
true	true	true	

p1	p2	p1 && p2	Example
false	false	false	(3 > 2) && (5 >= 5) is true, because (3 > 2) and (5 >= 5) are both true.
false	true	false	
true	false	false	(3 > 2) && (5 > 5) is false, because (5 > 5) is false.
true	true	true	

p1	p2	p1 p2
----	----	----------

false	false	false
false	true	true
true	false	true
true	true	true

Example (assume age = 24, gender = 'F')

(age > 34) || (gender == 'F') is true, because (gender == 'F') is true.

(age > 34) || (gender == 'M') is false, because (age > 34) and (gender == 'M') are both false.

p1	p2	p1 p2
----	----	----------

false	false	false
false	true	true
true	false	true
true	true	true

Example

(2 > 3) || (5 > 5) is false, because (2 > 3) and (5 > 5) are both false.

(3 > 2) || (5 > 5) is true, because (3 > 2) is true.

جدول درستی برای عملگر ^

p1	p2	p1 ^ p2	Example (assume age = 24, gender = 'F')
false	false	false	<u>(age > 34) ^ (gender == 'F')</u> is true, because <u>(age > 34)</u> is false but <u>(gender == 'F')</u> is true.
false	true	true	
true	false	true	<u>(age > 34) (gender == 'M')</u> is false, because <u>(age > 34)</u> and <u>(gender == 'M')</u> are both false.
true	true	false	

برنامه ای بنویسید که بررسی کند آیا یک عدد بر ۲ و ۳ بخشپذیر است؟

آیا بر ۲ یا ۳ بخشپذیر است؟

آیا بر ۲ یا ۳ اما نه بر هر دو بخشپذیر است؟

```
System.out.println("Is " + number + " divisible by 2 and 3? " +  
((number % 2 == 0) && (number % 3 == 0)));
```

```
System.out.println("Is " + number + " divisible by 2 or 3? " +  
((number % 2 == 0) || (number % 3 == 0)));
```

```
System.out.println("Is " + number +  
" divisible by 2 or 3, but not both? " +  
((number % 2 == 0) ^ (number % 3 == 0)));
```


این برنامه یک عدد را به عنوان سال (نوع صحیح) دریافت می کند و چک می کند که آیا آن سال کبیسه است؟

A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String args[]) {
        // Create a Scanner
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a year: ");
        int year = input.nextInt();

        // Check if the year is a leap year
        boolean isLeapYear =
            (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

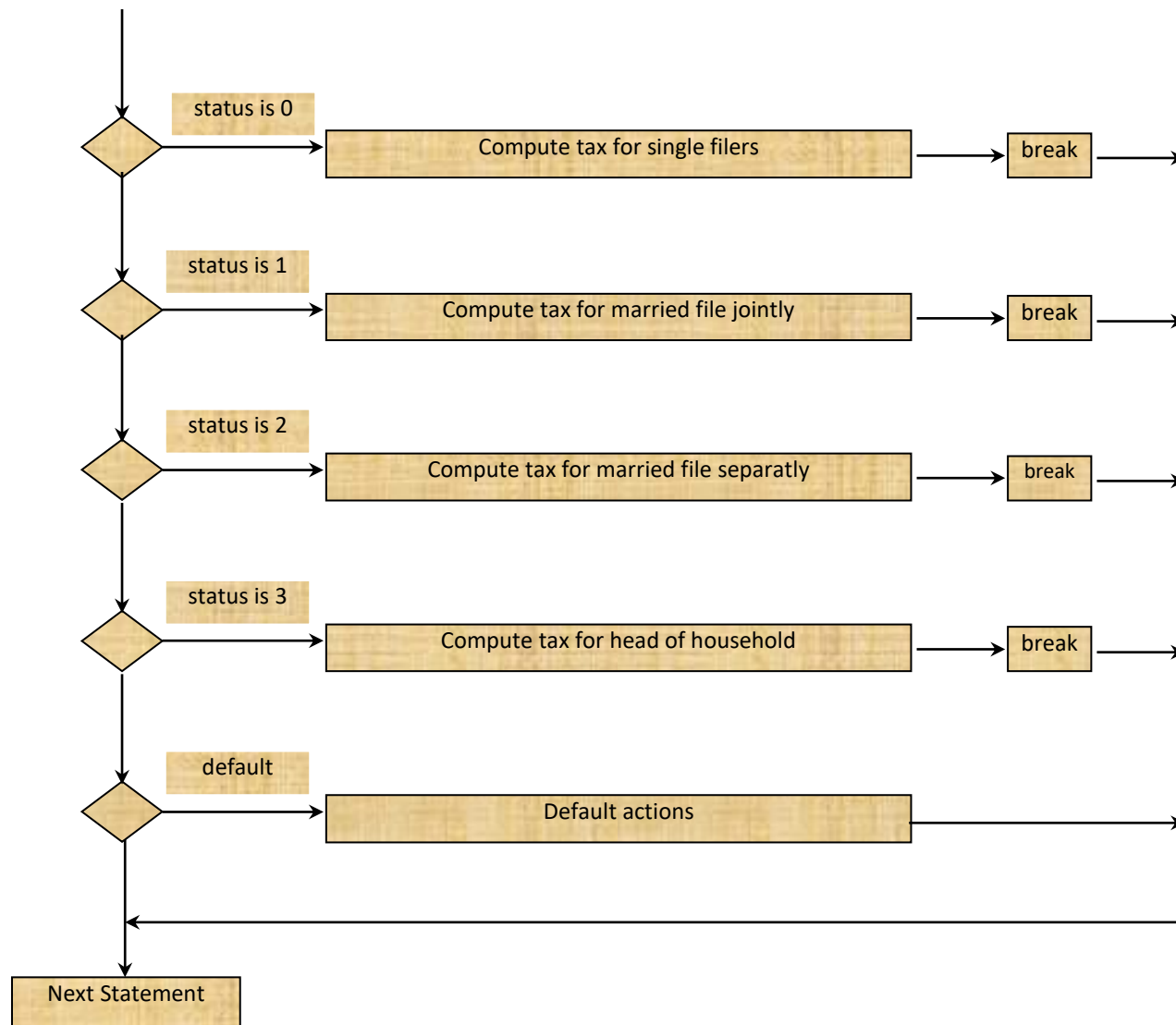
        // Display the result in a message dialog box
        System.out.println(year + " is a leap year? " + isLeapYear);
    }
}
```

Boolean Logical AND (&) and Boolean Logical Inclusive OR (|) Operators

```
( gender == 1 ) & ( age >= 65 )
```

```
( birthday == true ) | ( ++age >= 65 )
```

فلوچارت دستور switch



قواعد دستور switch

عبارت switch می تواند حاوی یک مقدار از نوع char, byte, short, int یا string باشد که همواره داخل پرانتز قرار دارند.

مقادیر value1, ..., و valueN باید هم نوع با عبارت داخل switch باشند. توجه کنید که مقادیر value1, ..., و valueN عبارات ثابت هستند. به این معنا که از متغیر در case ها استفاده نشود (مانند x+1)

```
switch (switch-expression) {
    case value1: statement(s)1;
                break;
    case value2: statement(s)2;
                break;
    ...
    case valueN: statement(s)N;
                break;
    default: statement(s)-for-default;
}
```

کلمه کلیدی break اختیاری است، اما در پایان هر case باید استفاده شود تا باقی دستورات case اجرا نشوند. اگر این دستور استفاده نشود، دستور case بعدی نیز اجرا خواهد شد.

حالت پیش فرض (default) که اختیاری است، در مواقعی استفاده می شوند که عبارت switch در هیچ case ای صدق نکند.

```
switch (switch-expression) {
    case value1: statement(s)1;
        break;
    case value2: statement(s)2;
        break;
    ...
    case valueN: statement(s)N;
        break;
    default: statement(s)-for-default;
}
```

Execute this line

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

Execute this line

```
switch (ch) {
    case 'a': System.out.println(ch);
    case 'b': System.out.println(ch);
    case 'c': System.out.println(ch);
}
```


Execute this line

```
switch (ch) {
    case 'a': System.out.println(ch);
    case 'b': System.out.println(ch);
    case 'c': System.out.println(ch);
}
```

Execute next statement

```
switch (ch)
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case ' ': System.out.println(ch);
}
```

Next statement;

Suppose ch is 'a':

```
switch (ch) {
    case 'a': System.out.println(ch);
               break;
    case 'b': System.out.println(ch);
               break;
    case 'c': System.out.println(ch);
}
}
```

ch is 'a':

```
switch (ch) {
    case 'a': System.out.println(ch);
               break;
    case 'b': System.out.println(ch);
               break;
    case 'c': System.out.println(ch);
}
}
```

Execute this line

```
switch (ch) {
  case 'a': System.out.println(ch);
             break;
  case 'b': System.out.println(ch);
             break;
  case 'c': System.out.println(ch);
}
}
```

Execute this line

```
switch (ch) {
    case 'a': System.out.println(ch);
               break;
    case 'b': System.out.println(ch);
               break;
    case 'c': System.out.println(ch);
}
}
```

Execute next statement

```
switch (ch)
  case 'a': System.out.println(ch);
             break;
  case 'b': System.out.println(ch);
             break;
  case 'c': System.out.println(ch);
}

```

Next statement;

```
if (x > 0)
```

```
    y = 1
```

```
else
```

```
    y = -1;
```

is equivalent to

```
y = (x > 0) ? 1 : -1;
```

```
(boolean-expression) ? expression1 : expression2
```



```

if (num % 2 == 0)
    System.out.println(num + "is even");
else
    System.out.println(num + "is odd");
    
```

```

System.out.println(
    (num % 2 == 0) ? num + "is even" :
    num + "is odd");
    
```

`(boolean-expression) ? exp1 : exp2`

از دستور printf برای ساختار بندی خروجی استفاده می کنیم:

`System.out.printf(format, items);`

ساختار (فرمت) رشته ای است که از زیررشته ها و تعیینگرهای ساختار استفاده می شود.

یک تعیینگر فرمت (foramt specifier) نحوه نمایش یک آیتم را مشخص می کند. یک آیتم می تواند یک مقدار عددی، کاراکتر، بولین یا رشته باشد.

هرتعیینگر با نماد % شروع می شود.

Specifier Output

<u>%b</u>	a boolean value
<u>%c</u>	a character
<u>%d</u>	a decimal integer
<u>%f</u>	a floating-point number
<u>%e</u>	a number in standard scientific notation
<u>%s</u>	a string

Example

true or false

'a'

200

45.460000

4.556000e+01

"Java is cool"

```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);
```

items

display

count is 5 and amount is 45.560000

عملگرهای بیتی

- جاوا چند عملگر بیتی تعریف کرده که می توانند بر روی انواع صحیح نظیر long، int، short، char و byte اعمال کنند.
- عملگر بیتی بر روی بیتها کار می کند و عملیات را بیت به بیت انجام می دهد.
- فرض کنید $a=60$ و $b=13$ باشند.
- نمایش دودویی این اعداد به صورت زیر می باشد:
- $a = 0011\ 1100$
- $b = 0000\ 1101$
- عملیات and، or، xor و not به ترتیب بر روی بیتهای دو عملوند a و b اعمال می شوند:

$$a \& b = 0000\ 1100$$

$$a | b = 0011\ 1101$$

$$a \wedge b = 0011\ 0001$$

$$\sim a = 1100\ 0011$$

جدولی از همه عملگرهای بیتی

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) will give 12 which is 0000 1100
	Binary OR Operator copies a bit if it exists in either operand.	(A B) will give 61 which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) will give 49 which is 0011 0001
~	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.	(~A) will give -61 which is 1100 0011 in 2's complement form due to a signed binary number.
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 will give 240 which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 will give 15 which is 1111
>>>	Shift right zero fill operator. The left operands value is moved right by the number of bits specified by the right operand and shifted values are filled up with zeros.	A >>> 2 will give 15 which is 0000 1111

- `var++`, `var--` (unary postfix increment and decrement)
- (unary prefix increment and decrement)
- `!` (Not)
- `~` unary bitwise complement
- `(type)` Casting
- `*`, `/`, `%` (Multiplication, division, and remainder)
- `+`, `-` (Binary addition and subtraction)
- `<<`, `>>`, `>>>` shift bits
- `<`, `<=`, `>`, `>=` (Comparison)
- `==`, `!=`; (Equality)
- `^` (Exclusive OR)
- `&` bitwise and, boolean logical and
- `^` bitwise xor , boolean logical xor
- `|` bitwise or , boolean logical or
- `&&` (Conditional AND) Short-circuit AND
- `||` (Conditional OR) Short-circuit OR
- `?:` conditional
- `=`, `+=`, `-=`, `*=`, `/=`, `%=`, `&=`, `^=`, `|=`, `<<=`, `>>=`, `>>>=` (Assignment operator)

تقدم عملگرها و شرکت پذیری

- شرکت پذیری همه عملگرهای دو عملوندی به جز انتساب از چپ به راست می باشد.
- شرکت پذیری عملگرهای تک عملوندی افزایشی (پیش و پس) از راست به چپ می باشد.
- شرکت پذیری عملگر شرطی: ؟ از راست به چپ می باشد.

3 + 4 * 4 > 5 * (4 + 3) - 1

3 + 4 * 4 > 5 * 7 - 1

3 + 16 > 5 * 7 - 1

3 + 16 > 35 - 1

19 > 35 - 1

19 > 34

false

(1) inside parentheses first

(2) multiplication

(3) multiplication

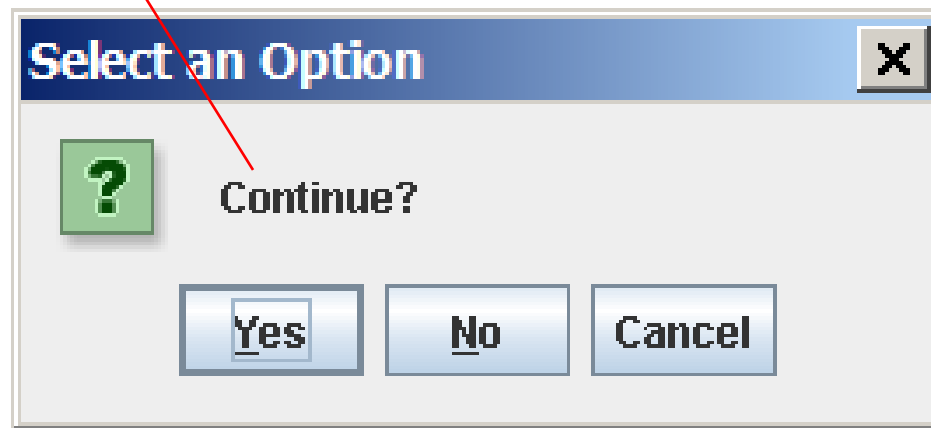
(4) addition

(5) subtraction

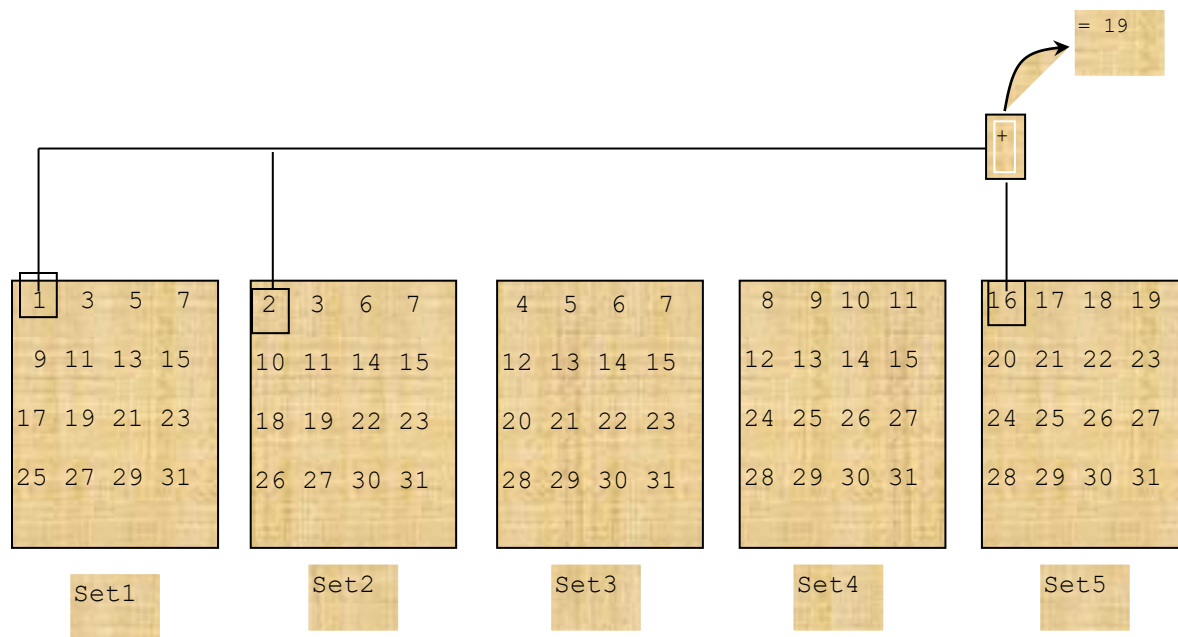
(6) greater than

(GUI) Confirmation Dialogs

```
int option = JOptionPane.showConfirmDialog (null, "Continue");
```



- این برنامه روز تولد شما (در ماه مربوطه) را حدس می زند.



```
import javax.swing.JOptionPane;
public class GuessBirthdayUsingConfirmationDialog
{
    public static void main(String[] args) {
        String set1 =
            " 1  3  5  7\n" +
            " 9 11 13 15\n" +
            "17 19 21 23\n" +
            "25 27 29 31";

        String set2 =
            " 2  3  6  7\n" +
            "10 11 14 15\n" +
            "18 19 22 23\n" +
            "26 27 30 31";

        String set3 =
            " 4  5  6  7\n" +
            "12 13 14 15\n" +
            "20 21 22 23\n" +
            "28 29 30 31";

        String set4 =
            " 8  9 10 11\n" +
            "12 13 14 15\n" +
            "24 25 26 27\n" +
            "28 29 30 31";

        String set5 =
            "16 17 18 19\n" +
            "20 21 22 23\n" +
            "24 25 26 27\n" +
            "28 29 30 31";

        int day = 0;
```

```
// Prompt the user to answer questions
int answer = JOptionPane.showConfirmDialog(null,
    "Is your birthday in these numbers?\n" + set1);

if (answer == JOptionPane.YES_OPTION)
    day += 1;

answer = JOptionPane.showConfirmDialog(null,
    "Is your birthday in these numbers?\n" + set2);

if (answer == JOptionPane.YES_OPTION)
    day += 2;

answer = JOptionPane.showConfirmDialog(null,
    "Is your birthday in these numbers?\n" + set3);

if (answer == JOptionPane.YES_OPTION)
    day += 4;

answer = JOptionPane.showConfirmDialog(null,
    "Is your birthday in these numbers?\n" + set4);

if (answer == JOptionPane.YES_OPTION)
    day += 8;

answer = JOptionPane.showConfirmDialog(null,
    "Is your birthday in these numbers?\n" + set5);

if (answer == JOptionPane.YES_OPTION)
    day += 16;
JOptionPane.showMessageDialog(null, "Your birthday is " +
    day + "!");
    }
}
```

- Syntax Errors
 - Detected by the compiler
- Runtime Errors
 - Causes the program to abort
- Logic Errors
 - Produces incorrect result

```
public class ShowSyntaxErrors {
    public static void main(String[] args) {
        i = 30;
        System.out.println(i + 4);
    }
}
```

```
public class ShowRuntimeErrors {
    public static void main(String[] args) {
        int i = 1 / 0;
    }
}
```

```
int average(int a, int b)
{
    return a + b / 2;    /* should be (a + b) / 2 */
}
```