

条件运算符 (?:)

条件运算符也被称为三元运算符。该运算符有3个操作数，并且需要判断布尔表达式的值。该运算符的主要是决定哪个值应该赋值给变量。

```
variable x = (expression) ? value if true : value if false
```

instanceOf 运算符

```
( Object reference variable ) instanceof (class/interface type)
```

如果运算符左侧变量所指的对象，是操作符右侧类或接口 (class/interface) 的一个对象，或比较的对象兼容于右侧类型，那么结果为真。

Java增强for循环

```
for (声明语句 : 表达式)
{
    //代码句子
}
```

声明语句：声明新的局部变量，该变量的类型必须和数组元素的类型匹配。其作用域限定在循环语句块，其值与此时数组元素的值相等。

表达式：表达式是要访问的数组名，或者是返回值为数组的方法。

实例

```
public class Test {

    public static void main(String args[]){
        int [] numbers = {10, 20, 30, 40, 50};

        for(int x : numbers ){
            System.out.print( x );
            System.out.print(",");
        }
    }
}
```

```

        System.out.print("\n");
        String [] names ={"James", "Larry", "Tom", "Lacy"};
        for( String name : names ) {
            System.out.print( name );
            System.out.print(",");
        }
    }
}

```

switch语句

```

switch(expression){
    case value :
        //语句
        break; //可选
    case value :
        //语句
        break; //可选
    //你可以有任意数量的case语句
    default : //可选。必须是switch语句的最后一个分支。default在没有case语句
的值和变量值相等的时候执行。default分支不需要break语句。
        //语句
}

```

+= 操作符实质

```

int i = 5;
long j = 8;

```

这时 `i = i + j` 不能编译，但 `i += j` 却可以编译。

`E1 op= E2` (`i += j`; `i -= j` 等等)，是等同于 `E1 = (T)((E1) op (E2))`

```

short x = 3;
x += 4.6;

```

等同于

```
short x = 3;  
x = (short) (x + 4.6);
```