

Java switch case 语句

switch case 语句判断一个变量与一系列值中某个值是否相等，每个值称为一个分支。

语法

switch case 语句语法格式如下：

```
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  
//语句  
}
```

switch case 语句有如下规则：

- switch 语句中的变量类型可以是：byte、short、int 或者 char。从 Java SE 7 开始，switch 支持字符串 String 类型了，同时 case 标签必须为字符串常量或字面量。
- switch 语句可以拥有多个 case 语句。每个 case 后面跟一个要比较的值和冒号。
- case 语句中的值的数据类型必须与变量的数据类型相同，而且只能是常量或者字面常量。
- 当变量的值与 case 语句的值相等时，那么 case 语句之后的语句开始执行，直到 break 语句出现才会跳出 switch 语句。
- 当遇到 break 语句时，switch 语句终止。程序跳转到 switch 语句后面的语句执行。case 语句不必须要包含 break 语句。如果没有 break 语句出现，程序会继续执行下一条 case 语句，直到出现 break 语句。
- switch 语句可以包含一个 default 分支，该分支一般是 switch 语句的最后一个分支（可以在任何位置，但建议在最后一个）。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。

switch case 执行时，一定会先进行匹配，匹配成功返回当前 case 的值，再根据是否有 break，判断是否继续输出，或是跳出判断。

实例

Test.java 文件代码：

```
public class Test {  
public static void main(String args[]){  
//char grade = args[0].charAt(0);  
}
```

```
char grade = 'C';
switch(grade)
{
case 'A' :
System.out.println("优秀");
break;
case 'B' :
case 'C' :
System.out.println("良好");
break;
case 'D' :
System.out.println("及格");
break;
case 'F' :
System.out.println("你需要再努力努力");
break;
default :
System.out.println("未知等级");
}
System.out.println("你的等级是 " + grade);
}
```

以上代码编译运行结果如下：

```
良好
你的等级是 c
```

如果 case 语句块中没有 break 语句时，JVM 并不会顺序输出每一个 case 对应的返回值，而是继续匹配，匹配不成功则返回默认 case。

Test.java 文件代码：

```
public class Test {
public static void main(String args[]){
int i = 5;
switch(i){
case 0:
System.out.println("0");
case 1:
System.out.println("1");
case 2:
System.out.println("2");
default:
System.out.println("default");
}
}
}
```

以上代码编译运行结果如下：

```
default
```

如果 case 语句块中没有 break 语句时，匹配成功后，从当前 case 开始，后续所有 case 的值都会输出。

Test.java 文件代码：

```
public class Test {  
    public static void main(String args[]){  
        int i = 1;  
        switch(i){  
            case 0:  
                System.out.println("0");  
            case 1:  
                System.out.println("1");  
            case 2:  
                System.out.println("2");  
            default:  
                System.out.println("default");  
        }  
    }  
}
```

以上代码编译运行结果如下：

```
1  
2  
default
```

如果当前匹配成功的 case 语句块没有 break 语句，则从当前 case 开始，后续所有 case 的值都会输出，如果后续的 case 语句块有 break 语句则会跳出判断。

Test.java 文件代码：

```
public class Test {  
    public static void main(String args[]){  
        int i = 1;  
        switch(i){  
            case 0:  
                System.out.println("0");  
            case 1:  
                System.out.println("1");  
            case 2:  
                System.out.println("2");  
            case 3:  
                System.out.println("3"); break;  
            default:  
                System.out.println("default");  
        }  
    }  
}
```

以上代码编译运行结果如下：

1

2

3

← Java 9 改进的 CompletableFuture API

 点我分享笔记