

菜鸟教程阅读笔记

笔记本: Java

创建时间: 2022/1/4 11:08

更新时间: 2022/1/4 16:54

作者: mdq1357@163.com

URL: <https://www.runoob.com/java/java-switch-case.html>

- **Java switch case 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。

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。

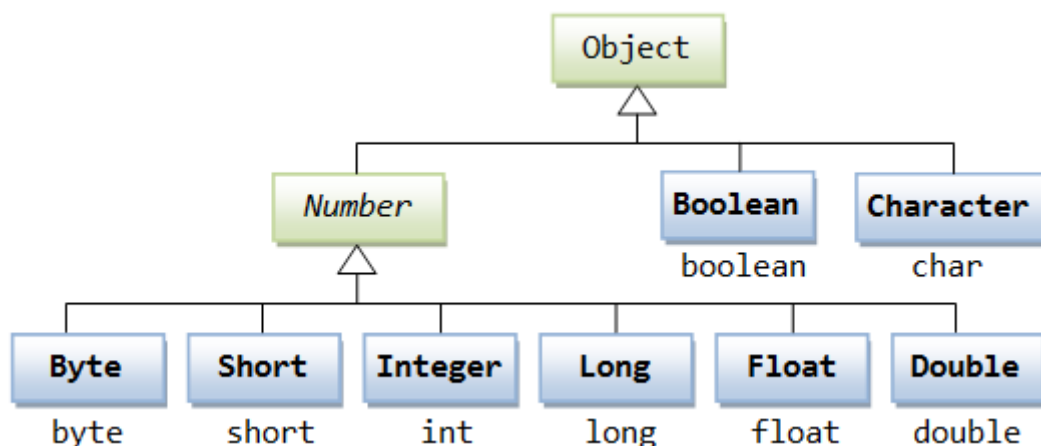
如果 case 语句块中没有 break 语句时, 匹配成功后, 从当前 case 开始, 后续所有 case 的值都会输出。如果后续的 case 语句块有 break 语句则会跳出判断。

- **Java Number & Math 类**

- <https://www.runoob.com/java/java-number.html>

内置数据类型, 如: byte、int、long、double 等。在实际开发过程中, 我们经常会遇到需要使用对象, 而不是内置数据类型的情形。为了解决这个问题, Java 语言为每一个内置数据类型提供了对应的包装类。

所有的包装类 (Integer、Long、Byte、Double、Float、Short) 都是抽象类 Number 的子类。



当做对象使用时: 装箱

进行运算时: 拆箱

Java 的 Math 包含了用于执行基本数学运算的属性和方法, 如初等指数、对数、平方根和三角函数。例如: `Math.sin(Math.PI/2)` 和 `Math.toDegrees(Math.PI/2)`。

- **Java Character 类**

- <https://www.runoob.com/java/java-character.html>

前面有反斜杠 (\) 的字符代表转义字符, 它对编译器来说是有特殊含义的。

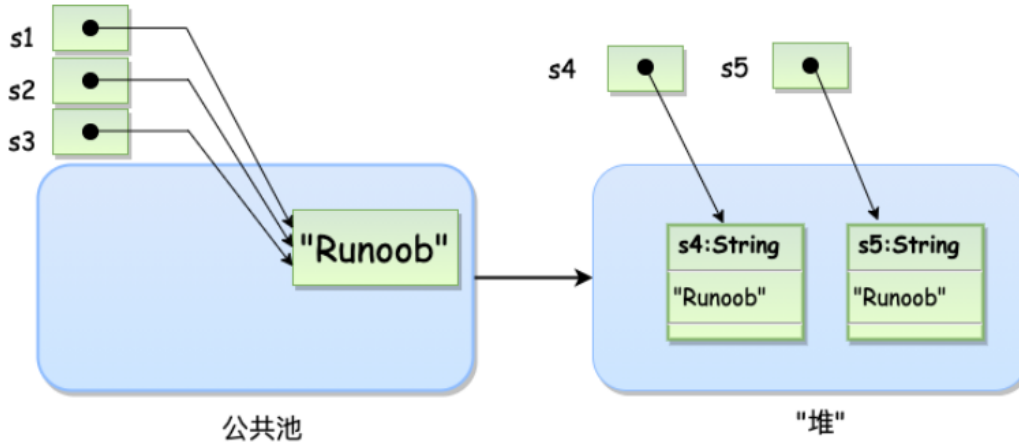
`toString()`

返回字符的字符串形式, 字符串的长度仅为1

- Java String 类
- <https://www.runoob.com/java/java-string.html>

String 创建的字符串存储在公共池中，而 new 创建的字符串对象在堆上：

```
String s1 = "Runoob";           // String 直接创建
String s2 = "Runoob";           // String 直接创建
String s3 = s1;                 // 相同引用
String s4 = new String("Runoob"); // String 对象创建
String s5 = new String("Runoob"); // String 对象创建
```



String 类是不可改变的，所以你一旦创建了 String 对象，那它的值就无法改变了。如果需要对字符串做很多修改，那么应该选择使用 StringBuffer & StringBuilder 类。

- 连接字符串的两种方法：

```
string1.concat(string2);
"Hello," + " runoob" + "!"
```

- 创建格式化字符串

我们知道输出格式化数字可以使用 printf() 和 format() 方法。

String 类使用静态方法 format() 返回一个 String 对象而不是 PrintStream 对象。

String 类的静态方法 format() 能用来创建可复用的格式化字符串，而不仅仅是用于一次打印输出。

如下所示：

```
System.out.printf("浮点型变量的值为 " + "%f", 整型变量的值为 " + "%d", 字符串变量的值为 " + "%s", floatVar, intVar, stringVar);
```

还可以这样写：

```
String fs;
fs = String.format("浮点型变量的值为 " + "%f", 整型变量的值为 " + "%d", 字符串变量的值为 " + "%s", floatVar, intVar, stringVar);
```

- Java compareTo() 方法

```
int result = str1.compareTo(str2);
```

1、不同的字符在较短字符串长度之内时，返回值=原字符串与参数字符串中第一个不同字符差的ASCII码值，为原减参。例子如下：

```
String str1="javasdrip";
String str2="javidscript";
str1.compareTo(str2);
```

此时返回值为-3，是a的ASCII码（97）减去了d的ASCII码值（100）得到。

注意：只比较第一个不同的字符，后面的d和c也不一样但不会进行比较了。

2、不同的字符在较短字符串长度之外时，返回值=原字符串与参数字符串相差的字符个数，原字符串长度大时为正，反之为负。例子如下：

```
String str1="java";
```

```
String str2="javascript";
```

```
str1.compareTo(str2);
```

此时返回值为-6，是str1相比str2少去的字符个数。

