Java语言基础

【Day11】

java.lang.Object

常用的包

- Java的包结构：

  java.lang 包 是Java最核心的包，JVM一启动自动加载这个包的所有类和接口，无需import。常见的类：System/String/Object/Class/...

  java.util包 是Java的工具包，包括很多工具类和集合。

  java.io 包 是输入输出的包，包括读写各种设备。

  java.net包 是网络编程的包，包括各种网络编程。

  java.sql 包 是操作数据库的所有类和接口。

- Java程序员在编程时，可以使用大量的类库，因此，Java编程需要记的很多，对编程能力的本身要求不是特别的高。

Object类的基本概念

- 在Java类继承结构中，java.lang.Object类位于顶端。
- 如果定义一个Java类时没有使用extends关键字声明其父类，则其父类为java.lang.Object 类。
- Object定义了"对象"的基本行为, 被子类默认继承.

```
public class Foo {
    …… …… ……
}
```

等价于

```
public class Foo  extends Object {
    …… …… ……
}
```

一切皆对象!

方法equals的详解

- Object的equals方法用于对象的“相等”逻辑。
- equals方法的定义如下：

```
public boolean equals(Object obj) {
    return (this == obj);
}
```

- equals()方法要求： 自反性/对称性/传递性/一致性/非空性。
- Java类可以根据需要重写继承自Object的equals方法。

方法equals的使用

```
class Point {
    private int x;
    private int y;
    public boolean equals(Object obj) {
        if (obj instanceof Point) {
            Point p = (Point) obj;
            return x == p.x && y == p.y;
        } else {
            return false;
        }
    }
}
```

首先判断参数对象是否为本类类型，否则直接返回false；是则再判断其属性值是否相等

方法equals的使用

```
Point p1 = new Point(1, 2);
Point p2 = new Point(1, 2);
System.out.println(p1 == p2);
System.out.println(p1.equals(p2));
```

- p1==p2返回值为false，因为p1和p2指向的是不同的对象；
- p1.equals(p2) 返回true，因为Point重写了equals；
- 如果Point没有重写equals，则equals和==是等价的；

方法hashCode的详解

- int hashCode() - 返回对象的哈西码值，对应一个内存。
- hashCode的规范要求：

  1 一致性，同一个对象，如果没有改变属性值，它的hashcode应该是固定的。

  2 如果两个对象判定相等，它们的hashcode应该是同一个值。

  3 如果两个对象不相等，它们的hashcode可以相同，但最好不相同。
- hashCode()方法和equals()方法的判断条件必须保持一致，如果重写一个，另外一个也必须进行重写。

方法toString的详解

- Object类中定义有toString方法，用于返回对象的字符串表示。
- 所有Java类都继承了toString方法，该方法默认返回字符串的形式为：“包名.类名@ hashCode值的十六进制”
- Java类可以根据需要重写toString方法以返回更有意义的信息。
- JAVA在使用System.out.println()打印对象时或者用+连接字符串时，默认调用toString()方法。

方法toString的使用

```
class Point {
    private int x;
    private int y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public String toString() {
        return "x=" + x + ",y=" + y;
    }
}
```

```
Point p = new Point(1, 2);
System.out.println(p);
```

System.out.println方法会调用对象的toString方法，将返回的字符串输出

未重写toString方法时输出如：
test.Point@5d888759
重写toString后将输出：
x=1,y=2

包装类和数学处理类

包装类的概述

- Java语言8种基本类型分别对应了8种“包装类”，每一种包装类都封装了一个对应的基本类型成员变量，还提供了一些针对该数据类型的实用方法。

包装类	对应的基本类型
java.lang.Integer	int
java.lang.Long	long
java.lang.Double	double
java.lang.Character	char
java.lang.Boolean	boolean
java.lang.Byte	byte
java.lang.Float	float
java.lang.Short	short

装箱和拆箱

- JDK 5发布之前使用包装类对象进行运算时，需要较为繁琐的“拆箱”和“装箱”操作；即运算前先将包装类对象拆分为基本类型数据，运算后再将结果封装成包装类对象。

```
Integer i = Integer.valueOf(100);
Integer j = Integer.valueOf(200);
Integer k = Integer.valueOf(i.intValue() + j.intValue());
```

- JDK5 增加了自动“拆箱”和“装箱”的功能。

```
Integer i = 100;
Integer j = 200;
Integer k = i+j;
```

事实上JDK5的自动“拆箱”和“装箱”是依靠JDK5的编译器在编译器的“预处理”工作。

BigDecimal类的详解

- Java浮点数据类型（float和double）在运算时会有舍入误差；如果希望得到精确的计算结果，可以使用java.math. BigDecimal类。

```
BigDecimal d1 = new BigDecimal("3.0");
BigDecimal d2 = new BigDecimal("2.9");
BigDecimal d3 = d1.subtract(d2);
System.out.println(d3);  0.1

BigDecimal d4 = d1.divide(d2, 8, BigDecimal.ROUND_HALF_UP);
System.out.println(d4);  1.03448276
```

对于divide方法，通常需要制定精度和舍入模式，否则当遇到无限小数时，除法会一直进行下去直至抛出异常。

字符串String类型

String类型的特性

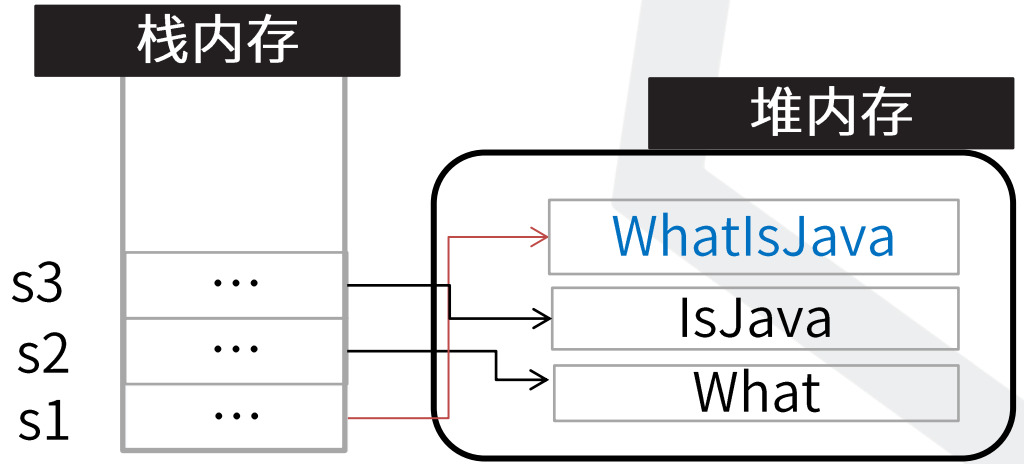
- java.lang.String用于封装字符串序列。
- Java字符串在内存中采用Unicode编码方式，任何一个字符对应两个字节的定长编码。
- String类属于不可变类，即String对象一经创建后，其封装的字符串序列是不能改变的。

String类的不可变性

```
String s1= "What";
String s2 = s1;
String s3 = "IsJava";
s1+=s3;
System.out.println(s1);
System.out.println(s2);
System.out.println(s3);
```

输出:
WhatIsJava
What
IsJava

s1引用了s1与s3连接产生新对象"WhatIsJava".
s1 引用的原对象"What" 是没有变化的.
s2引用的是原对象"What".
引用变量变了, String对象没有变!



String类的常量池

- Java语言中可以使用直接量“字符序列”创建字符串序列

  String str = “WhatisJava”

- 出于性能的考虑，JVM会将字符串字面量对象缓存在常量池中；对于重复出现的字符串直接量，JVM会首先在缓存池中查找，如果存在即返回该对象。

```
String str1 = "WhatisJava";
String str2 = "WhatisJava";
System.out.println(str1 == str2);
String str3 = new String("WhatisJava");
System.out.println(str1 == str3);
```

返回true，第二次写出“WhatisJava”不会重复创建String对象，使用str1所指向的对象

返回false，使用new关键字将会创建新的String对象。

总结与答疑

技术资料

IT兄弟连

ITXDL.CN

变态严管　让学习成为一种习惯