Java常用类

优逸客 让天下没有难学的技术

官网：http://www.sxuek.com/

# 第一章 基本数据包装类

数据类型：基本数据类型、引用数据类型（数组、class、接口）；

区别：基本数据类型没有属性和方法；

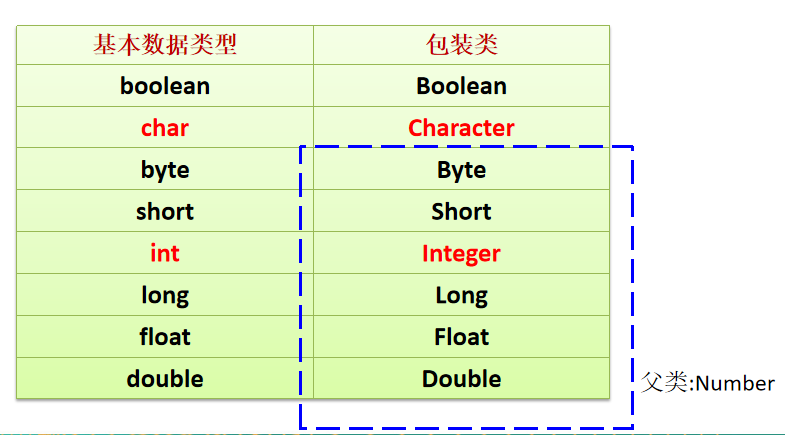
需求：

（1）获取int类型的最大值；

（2）将某个整数转成十六进制形式；

包装类概念：为八大基本数据类型对应的提供的引用数据类型，对值进行了包装；

|  |
| --- |
| *Int i = 10;*  *Integer i1 = new Integer(i);*  *i1.属性； i1.方法();* |



1. 基本数据类型包装成包装类的实例：装箱

装箱：包装类使得一个基本数据类型的数据变成了类。有了类的特点，可以调用类中的方法。

（1）通过包装类的构造器实现：

|  |
| --- |
| *int i = 500;*  *Integer t = new Integer(i);* |

1. 还可以通过字符串参数构造包装类对象：

|  |
| --- |
| *Float f = new Float(“4.56”);*  *Long l = new Long(“asdf”); //NumberFormatException* |

2）获得包装类对象中包装的基本类型变量：拆箱

拆箱：将数字包装类中内容变为基本数据类型。

调用包装类的.xxxValue()方法：

*boolean b = bObj.booleanValue();*

JDK1.5之后，支持自动装箱，自动拆箱。但类型必须匹配。

3）字符串转换成基本数据类型：

通过包装类的构造器实现：

*int i = new Integer(“12”);*

通过包装类的parseXxx(String s)静态方法：

*Float f = Float.parseFloat(“12.1”);*

4）基本数据类型转换成字符串：

调用字符串重载的valueOf()方法：

*String fstr = String.valueOf(2.34f);*

更直接的方式：

*String intStr = 5 + “”*

5）Character：

这类包装 Character值对象中的原始类型 char。 Character类型的对象包含一个字段的类型是 char。 此外，该类提供了用于确定字符的分类的几种方法（小写字母，数字，等）和由大写转换为小写字符，反之亦然。

# 第二章 字符串相关类

## 2.1 String类：

String类：用于保存一组字符串序列，相当于字符串常量对象；

常量对象：字符串常量对象是用双引号括起的字符序列。例如："你好"、"12.97"、"boy"等。

字符串的字符使用Unicode字符编码，一个字符占两个字节；

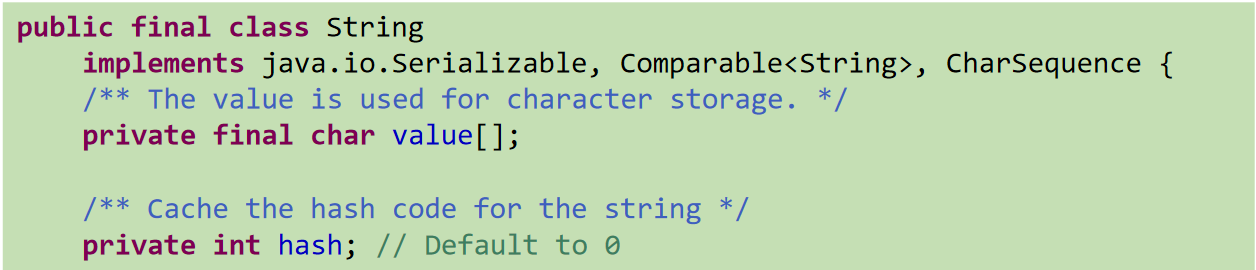
String的特性

String类：代表字符串。Java 程序中的所有字符串字面值（如 "abc" ）都作为此类的实例实现。

String是一个final类，代表不可变的字符序列。

字符串是常量，用双引号引起来表示。它们的值在创建之后不能更改。

String对象的字符内容是存储在一个字符数组value[]中的。



### 2.1.1构建String类对象：

1）通过调用String类的构造器；*String str = new String("abc");*

2）直接赋值；*String str ="abc"；*

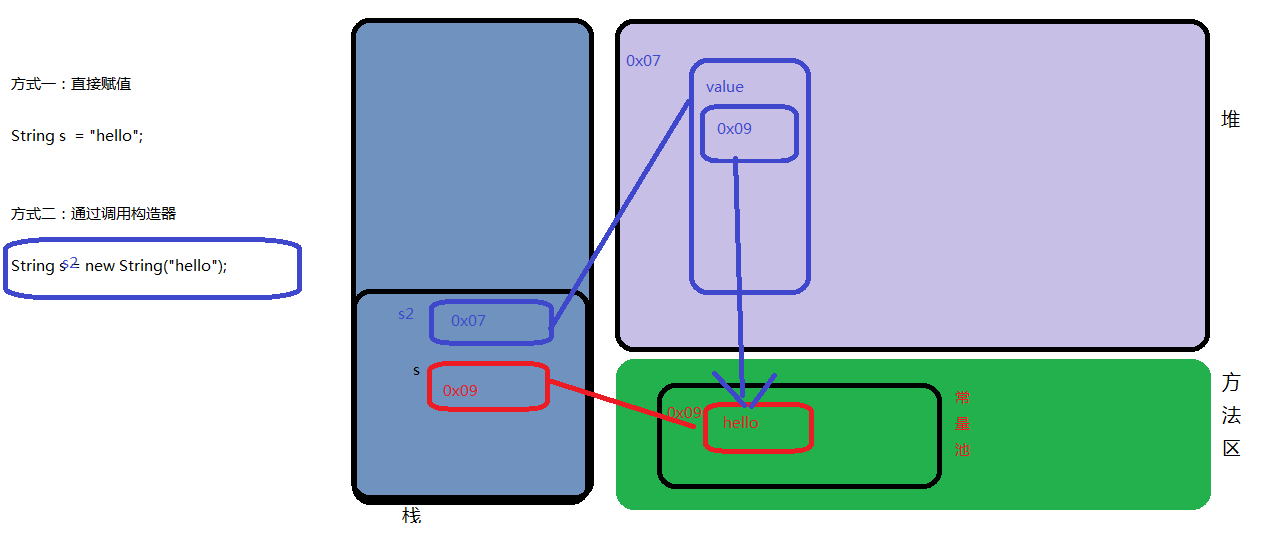
注意：常量池属于类信息的一部分，而类信息反映到JVM内存模型中是对应存在于JVM内存模型的方法区，也就是说这个类信息中的常量池概念是存在于在方法区中。一般这种情况下，"abc"在编译时就被写入字节码中，所以class被加载时，JVM就为"abc"在常量池中分配内存，所以和静态区差不多。

字符串常量池：

JAVA虚拟机(JVM)中存在着一个字符串常量池，其中保存着很多String对象;并且可以被共享使用，因此它提高了效率。由于String类是final的，它的值一经创建就不可改变。字符串池由String类维护，我们可以调用intern()方法来访问字符串池。

String类较常用构造方法:

|  |
| --- |
| *String s1 = new String();*  *String s2 = new String(String original);*  *String s3 = new String(char[] a);*  *String s4 = new String(char[] a,int startIndex,int count)* |



### 2.1.2 两种构建对象方式的区别：

String s = “hello”;

方式1：先去常量池查看是否有“hello”字符序列，如果没有，则创建，如果有直接引用该地址（s指向常量池的对象）；

String s = new String(“hello”);

方式2：需要在堆中创建对象，该对象维护了一个value属性，value指向常量池的“hello”,如果常量池中没有“hello”, 则创建，再指向；如果已经有了，则直接用value指向（s指向的是堆中的对象）；

案例：

### 2.1.3 字符串的特性：

String是一个final类，代表不可变的字符序列；

字符串是不可变的。一个字符串对象一旦被配置，其内容是不可变的。

案例：

### 2.1.4 字符串对象操作：

## 2.2 StringBuffer类：

java.lang.StringBuffer代表可变的字符序列，可以对字符串内容进行增删。很多方法与String相同，但StringBuffer是可变长度的。StringBuffer是一个容器。

StringBuffer和String的区别：

相同点：都用于保存字符串，都可以对字符串进行一些增删或者其他处理的操作；

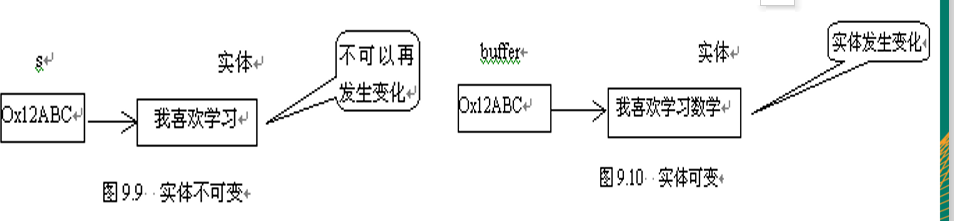
不同点：String类用于保存字符串常量；

StringBuffer用于保存字符串变量；

StringBuffer类有三个构造器：

|  |
| --- |
| *1．StringBuffer() 初始容量为16的字符串缓冲区*  *2．StringBuffer(int size) 构造指定容量的字符串缓冲区*  *3．StringBuffer(String str) 将内容初始化为指定字符串内容*  *public class TestStringBuffer {*  *public static void main(String[] args) {*  *StringBuffer s1 = new StringBuffer(); //初始化了char[],容量为16*  *StringBuffer s2 = new StringBuffer(20); //初始化了char[],容量为指定的容量20*  *StringBuffer s3 = new StringBuffer("hello");//初始化了char[],容量为str.length+16,初始值：str*  *}*  *}* |

|  |
| --- |
| *String s = new String("我喜欢学习");*  *StringBuffer buffer = new StringBuffer(“我喜欢学习”);*  *buffer.append("数学");* |



案例：

## 2.3 StringBuilder类：

StringBuilder 和 StringBuffer 非常类似，均代表可变的字符序列，而且方法也一样。

String：不可变字符序列；

StringBuffer：可变字符序列、效率低、线程安全；

StringBuilder(JDK1.5)：可变字符序列、效率高、线程不安全；

# 第三章 Math类

java.lang.Math提供了一系列静态方法用于科学计算；其方法的参数和返回值类型一般为double型。

# 第四章 System类

System类代表系统，系统级的很多属性和控制方法都放置在该类的内部。该类位于java.lang包。由于该类的构造器是private的，所以无法创建该类的对象，也就是无法实例化该类。其内部的成员变量和成员方法都是static的，所以也可以很方便的进行调用。

成员变量：

System类内部包含in、out和err三个成员变量，分别代表标准输入流(键盘输入)，标准输出流(显示器)和标准错误输出流(显示器)。

成员方法：

*native long currentTimeMillis()：*该方法的作用是返回当前的计算机时间，时间的表达格式为当前计算机时间和GMT时间(格林威治时间)1970年1月1号0时0分0秒所差的毫秒数。

*void exit(int status)：*该方法的作用是退出程序。其中status的值为0代表正常退出，非零代表异常退出。使用该方法可以在图形界面编程中实现程序的退出功能等。

*void gc()：*该方法的作用是请求系统进行垃圾回收。至于系统是否立刻回收，则取决于系统中垃圾回收算法的实现以及系统执行时的情况。

*String getProperty(String key)：*该方法的作用是获得系统中属性名为key的属性对应的值。系统中常见的属性名以及属性的作用如下表所示：



|  |
| --- |
| *public static void main(String[] args) {*  *String javaVersion = System.getProperty("java.version");*  *System.out.println("java的version:" + javaVersion);*  *String javaHome = System.getProperty("java.home");*  *System.out.println("java的home:" + javaHome);*  *String osName = System.getProperty("os.name");*  *System.out.println("os的name:" + osName);*  *String osVersion = System.getProperty("os.version");*  *System.out.println("os的version:" + osVersion);*  *String userName = System.getProperty("user.name");*  *System.out.println("user的name:" + userName);*  *String userHome = System.getProperty("user.home");*  *System.out.println("user的home:" + userHome);*  *String userDir = System.getProperty("user.dir");*  *System.out.println("user的dir:" + userDir);*  *}* |

# 第五章 BigInteger和BigDecimal：

Integer类作为int的包装类，能存储的最大整型值为，Long类也是有限的， 最大为。如果要表示再大的整数，不管是基本数据类型还是他们的包装类都无能为力，更不用说进行运算了。

java.math包的BigInteger可以表示不可变的任意精度的整数。BigInteger 提供

所有 Java 的基本整数操作符的对应物，并提供 java.lang.Math 的所有相关方法。 另外，BigInteger 还提供以下运算：模算术、GCD 计算、质数测试、素数生成、位操作以及一些其他操作。

构造器：

|  |
| --- |
| *BigInteger(String val)：*根据字符串构建BigInteger对象； |

一般的Float类和Double类可以用来做科学计算或工程计算，但在商业计算中， 要求数字精度比较高，故用到java.math.BigDecimal类。

BigDecimal类支持不可变的、任意精度的有符号十进制定点数。

构造器：

|  |
| --- |
| *public BigDecimal(double val)*  *public BigDecimal(String val)* |

常用方法：

|  |
| --- |
| *public BigDecimal add(BigDecimal augend)*  *public BigDecimal subtract(BigDecimal subtrahend)*  *public BigDecimal multiply(BigDecimal multiplicand)*  *public BigDecimal divide(BigDecimal divisor, int scale, int roundingMode)* |

案例：