Java SE笔记----

# java.lang.Object类

## hashCode()

理解:返回当前对象的一个16进制的数字,用于区别不同的对象.每个对象的hashCode()值都不一样.

用法: int i = 对象名.hashCode();

## toString()

功能: 以字符串形式返回对象信息.

## equals()

功能:判断对象是否相等

equals(): 比较堆(两个相同的字符串, 公用一个堆)

== : 比较栈(基本类型的值 和 对象的引用的值)

## finalize()

功能: 垃圾收集器在销毁一个对象之前,会自动调用该对象的finalize()方法,然后才释放对象的内存空间.

## close()

功能: 对象拷贝

方法格式: protected native Object clone() throws CloneNotSupportedException;

注意: 要进行”克隆” 的对象所属的类必须实现java.lang.Cloneable接口.

## wait()

## notify()/notifyAll()

String

代表字符串.字符串是常量；它们的值在创建之后不能更改。字符串缓冲区支持可变的字符串。因为 String 对象是不可变的，所以可以共享。

StringBuffer

线程安全的可变字符序列。一个类似于 String 的字符串缓冲区，但不能修改。虽然在任意时间点上它都包含某种特定的字符序列，但通过某些方法调用可以改变该序列的长度和内容。

append

方法始终将这些字符添加到缓冲区的末端；而 insert 方法则在指定的点添加字符。

StringBuilder

线程不安全安全的可变字符序列。

java.util.StringTokenizer

允许应用程序将字符串分解为标记。tokenization 方法比 StreamTokenizer 类所使用的方法更简单。StringTokenizer 方法不区分标识符、数和带引号的字符串，它们也不识别并跳过注释。

可以在创建时指定，也可以根据每个标记来指定分隔符（分隔标记的字符）集。

StringTokenizer 的实例有两种行为方式，这取决于它在创建时使用的 returnDelims 标志的值是 true 还是 false：

如果标志为 false，则分隔符字符用来分隔标记。标记是连续字符（不是分隔符）的最大序列。

如果标志为 true，则认为那些分隔符字符本身即为标记。因此标记要么是一个分隔符字符，要么是那些连续字符（不是分隔符）的最大序列。

# 八种基本数据类型----引用类型

|  |  |
| --- | --- |
| 基本数据类型 | 封装类 |
| boolean | Boolean |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| char | Character |
| float | Float |
| double | Double |

# 日期相关类型:

## Date类

用法: 用于表示特定的时间点,能精确到毫秒, 但不支持日期的国际化,和分时区显示.

## Calendar类

抽象类,主要提供日期修改功能,和常用国际化支持.

## Locale类

主要用于在国际化/本地化程序中以地区/语言相关的方式显示日期/数字或文本信息等.

## TimeZone类

## GregorianCalendar类

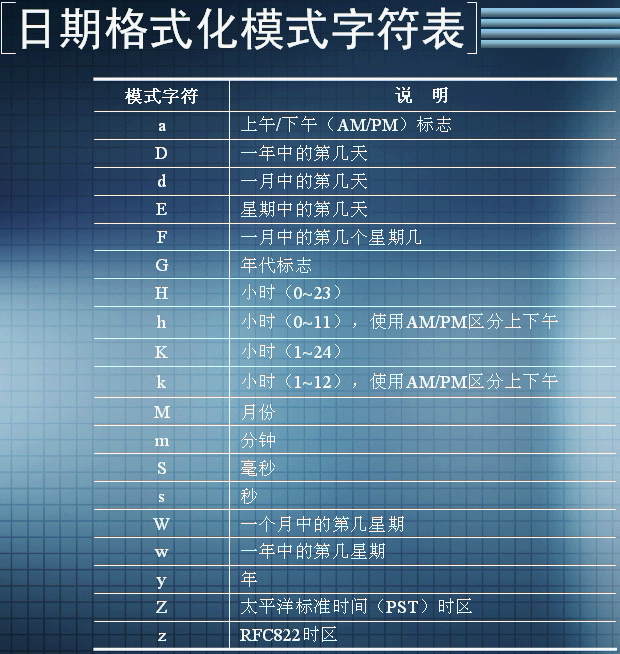
Calendar类的子类,提供了世界上大多数国家/地区使用的标准日历系统,并添加判断闰年功能.

## java.text.DateFormat类

是日期/时间格式化子类的抽象类，它以与语言无关的方式格式化并解析日期或时间。日期/时间格式化子类（如 SimpleDateFormat）允许进行格式化（也就是日期 -> 文本）、解析（文本-> 日期）和标准化。将日期表示为 Date 对象，或者表示为从 GMT（格林尼治标准时间）1970 年 1 月 1 日 00:00:00 这一刻开始的毫秒数。

## java.text.SimpleDateFormat类

是一个以与语言环境有关的方式来格式化和解析日期的具体类。它允许进行格式化（日期 -> 文本）、解析（文本 -> 日期）和规范化。



# IO

File类

RandomAccessFile类

各种节点流类

字符编码

各种过滤流与包装类

IO类的相关运用