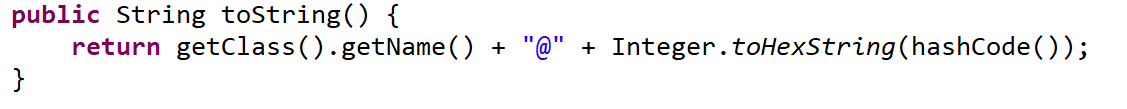
1. Object
2. Object

>.在Java类继承结构中，Java.lang.Object位于顶端。

>.如果没有显式的写出一个类的父类，则这个类默认继承Object。

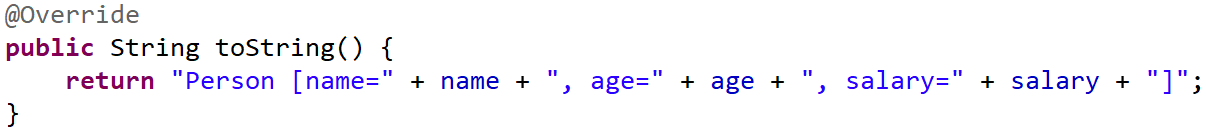
>.Object类型引用可以指向任意类型。

1. toString方法



>. Object的toString方法打印类全名+@+散列码。

>. 重写toString格式遵循“类名[域值]”。



* 如何重写toString方法

>. Eclipse可以直接生成，右键->source->generate toString

>.Java语言中很多地方会默认调用对象的toString方法:

- 字符串 + 对象时，会自动调用对象的toString方法。

- System.out.print(任意对象),直接调用toString方法

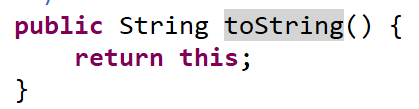
>. 如果不重写则会调用Object类中的toString方法

类名 @ 散列码

>. toString是常用的高效的调试工具，方便用户获取对象的信息，!强烈建议对每一个类都增加toString方法。

* String重写toString

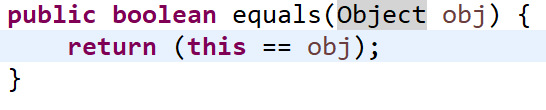
>. String类中toString方法返回值为自身。



1. equals方法

* 如何重写equals方法

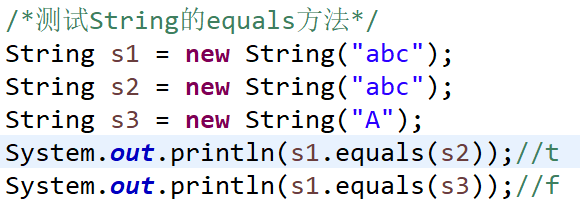
>. Object中的方法，作用在于检测一个对象是否等于另一个对象；



>. Object中判读这两个对象是否具有相同的引用。

>. 注意:一般情况下我们需要在实体类中重写该方法，通过比较对象的成员变量，使该方法更具有意义。

* String重写equals



String中也重写了equals方法，则比较的是String的值。

* equals和==区别(面试)

>.在没有重写equals的情况下，equals和==没有任何区别，都是比较对象的引用(地址值)

>.重写后，equals比较的是对象的内容(属性值)。

二、 包装类

1. 包装类的概述

>. 在进行类型转换的范畴内，有一种特殊的转换，将基本类型转换为对象。

>. 所有基本类型都有一个与之对应的对象，包装类(Wapper)。

>. 包装类是不可变类，在构造包装类之后，不允许更改包装类中的值。   
 >.8中基本类型对应的包装类

|  |  |  |
| --- | --- | --- |
| 基本类型 | 基本类型对应的包装类 | 包装类的父类 |
| int | java.lang.Integer | java.lang.Number |
| long | java.lang.Long | java.lang.Number |
| short | java.lang.Short | java.lang.Number |
| double | java.lang.Double | java.lang.Number |
| float | java.lang.Float | java.lang.Number |
| byte | java.lang.Byte | java.lang.Number |
| char | java.lang.Character | java.lang.Object |
| Boolean | java.lang.Boolean | java.lang.Object |

2. 8种基本类型包装类

- Number及其主要方法



>. 抽象类Number是Byte、Double、Float、Integer、Long和Short类的父类

>. 它提供了将表示的数值转化为基本类型的方法。

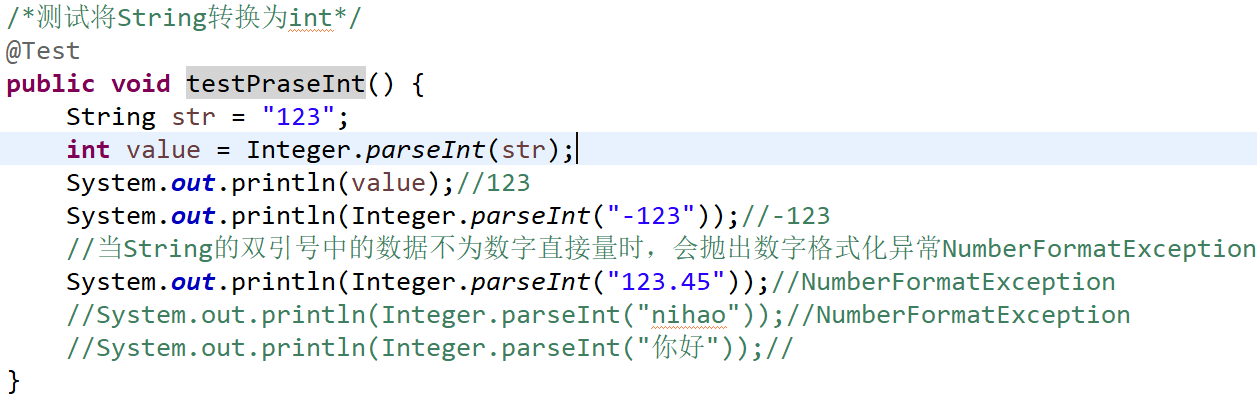
- Integer常用功能

>. 它能在int类型和String类型之间互相转换。

static int MAX\_VALUE = 2^31-1 int类型最大值

static int MIN\_VALUE = -2^31 int类型最小值

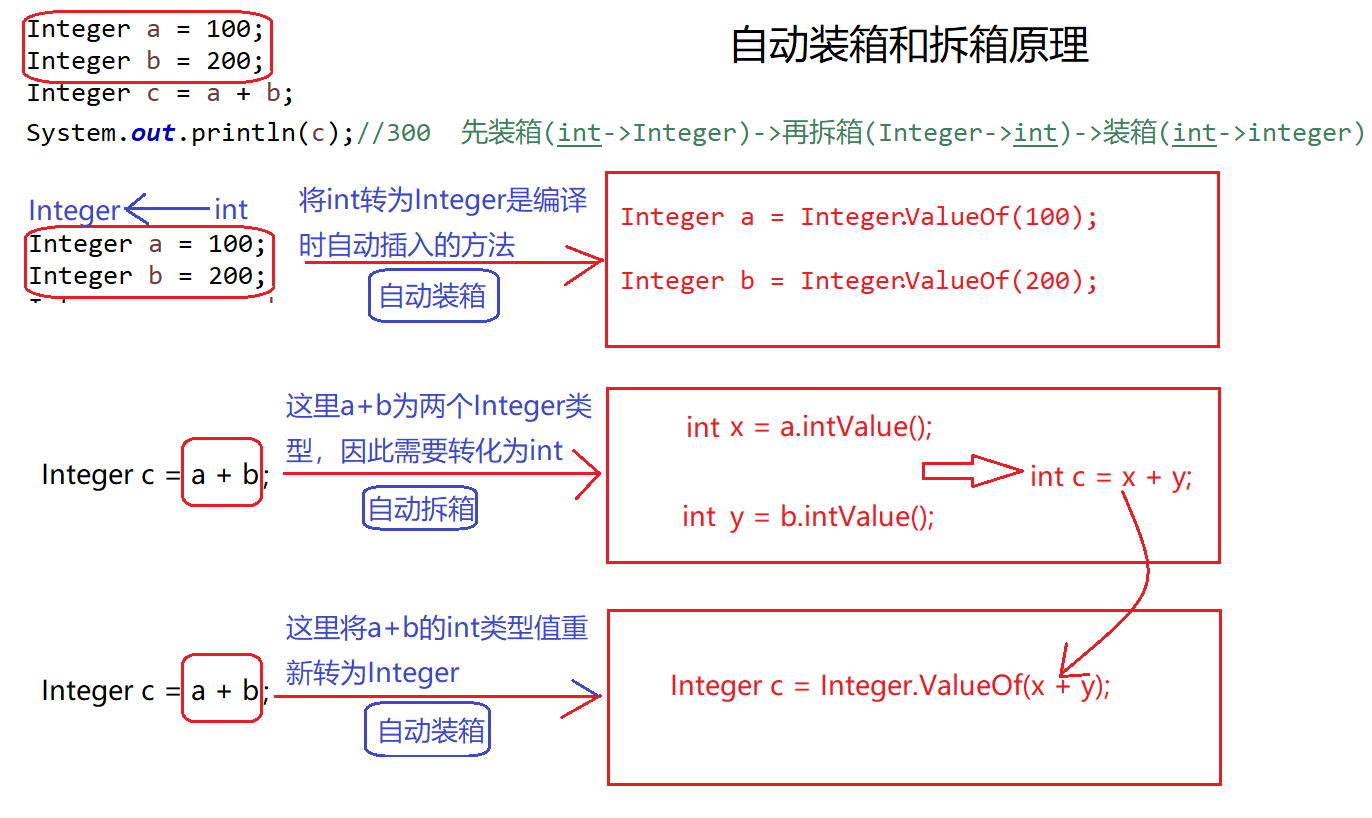
>. parseInt方法用于将字符串转为int类型



- Double的常用功能

>. parseDouble用于将String类型转为小数类型。

3. 自动装箱和拆箱(理解)



三、日期操作

1.Date及其常用API

- Java中的时间

>. Java中的时间使用Date类表示，用距离一个固定时间点的毫秒数(可负可正，long类型)表示一个特定的时间。

>. 固定的时间点(epoch)纪元，是UTC时间1970年1月1日 00:00:00;

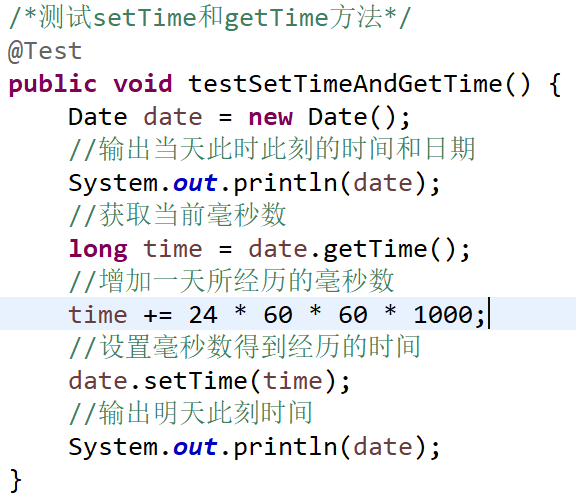
>. UTC(Universal Time Goordinated世界调整时间)与GMT(Greenwich Mean Time格林威治时间)一样，是一种具有实际目的的科学标准时间。

- Date类

>.java.util.Date类封装时间及日期信息。

>.Date类的大多数用用于进行时间分量计算的方法已经被Calendar所取代。

- setTime和getTime方法



- Date重写toString方法

>. Date的toString方法底层采用StringBuilder进行字符串拼接，格式为:

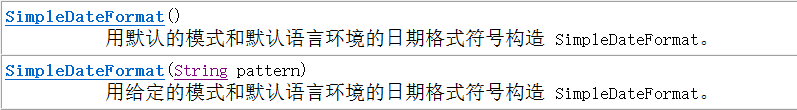
Thu Jan 28 09:38:32 CST 2021

2.SimpleDateFormat

- 简介

>.java.text.SimpleDateFormat是一个以语言环境有关的方式来格式化和解析日期的具体类。 格式化日期--->文本 解析文本--->日期和规范化。

>.构造方法



>.常用方法

(DataFormat)

Date ------> String

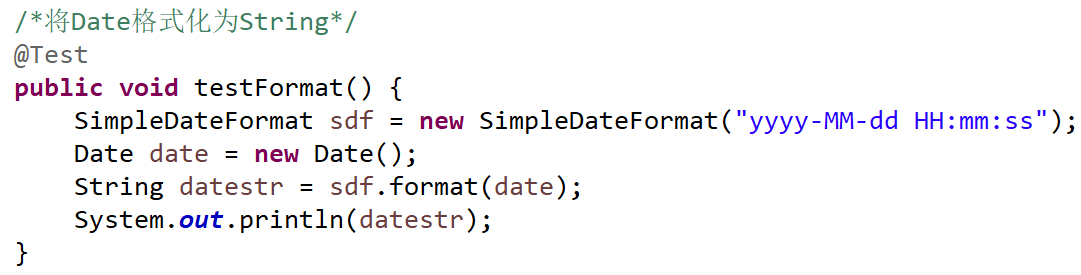


String ------>Date

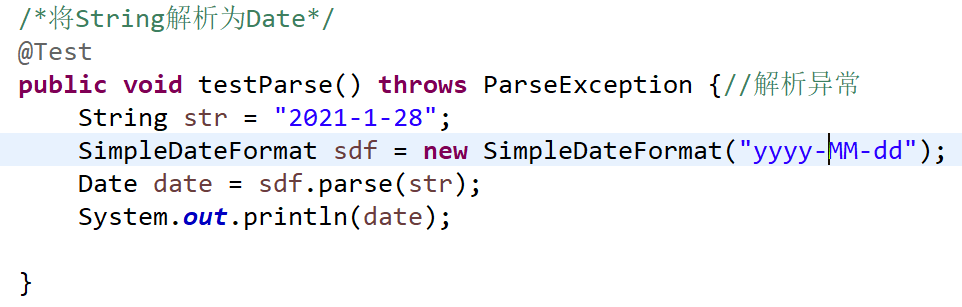
- 日期模式匹配字符

|  |  |  |
| --- | --- | --- |
| 字符 | 含义 | 示例 |
| y | 年 | yyyy年-2013年；yy—13年 |
| M | 月 | MM月-01月;M-1月 |
| d | 日 | dd日-06日;d日-6日 |
| E | 星期 | E-星期日 |
| a | AM或PM的标识 | a-下午(PM) |
| H | 小时(24小时制) | a h时-下午12时  HH:mm:sss – 12:45:37  hh(a):mm:ss – 12(下午):47分:48 |
| h | 小时(12小时制) |
| m | 分钟 |
| s | 秒 |

- 将Date格式化String



- 将String格式化为Date



3.Calendar

- 简介

>. java.util.Calendar类用于封装日历信息，其主要作用在于其方法可以对时间分量进行运算；

>.Calendar是抽象类，其具体子类针对不同国家的日历系统，有着不同的实现，最常见的为GregorianCalendar(格利高里历，阳历)。对应世界上绝大多数标准日历系统。

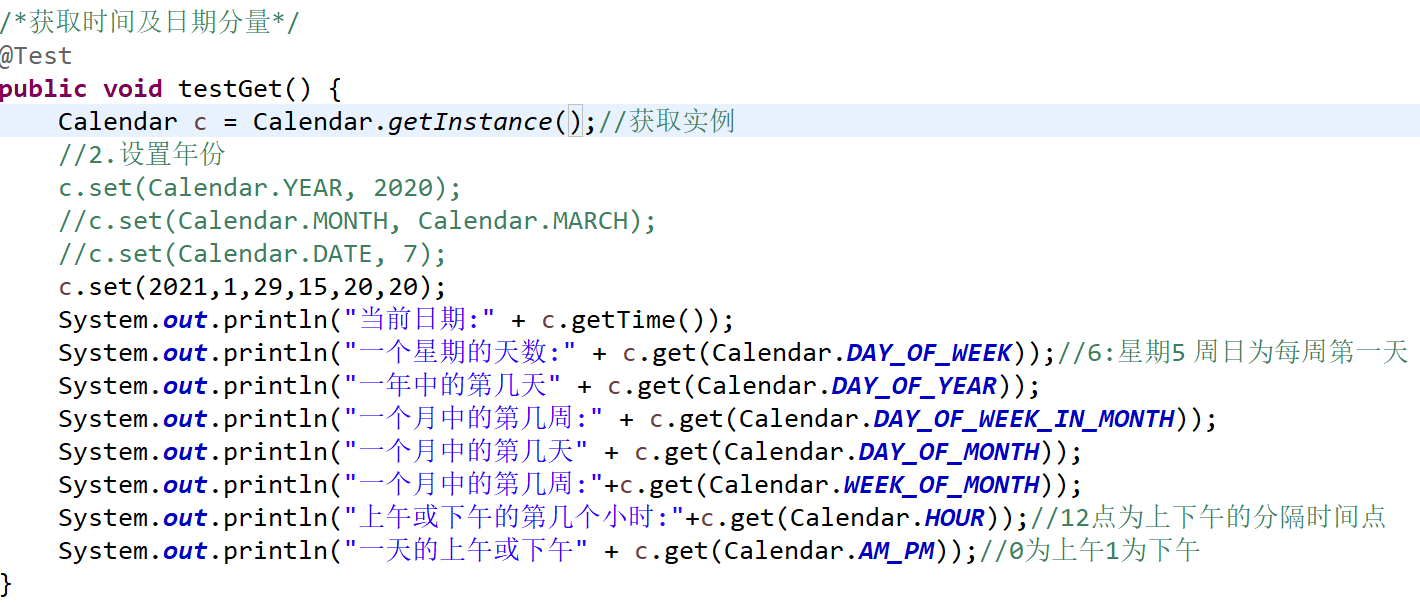
- getInstance方法

>.Calendar提供了一个类方法getInstance，以获得此类型的一个通用对象。

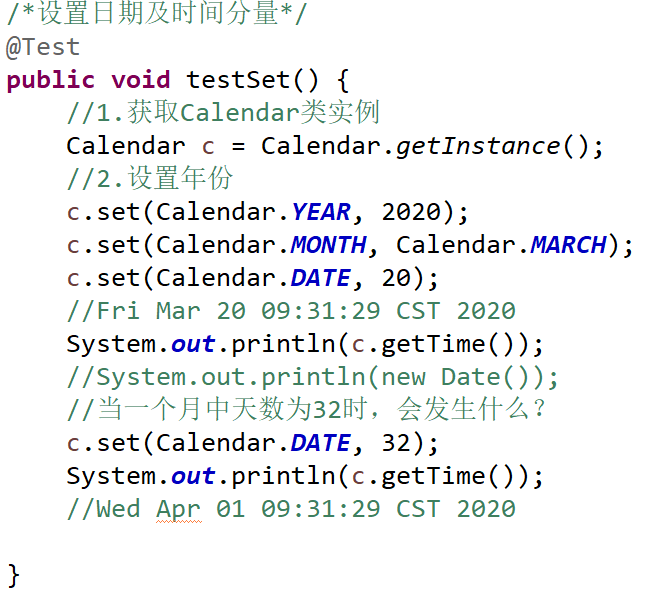
>.Calendar的getInstance返回的对象，其日历字段由当前时间初始化。

Calendar c = Calendar.getInstance();

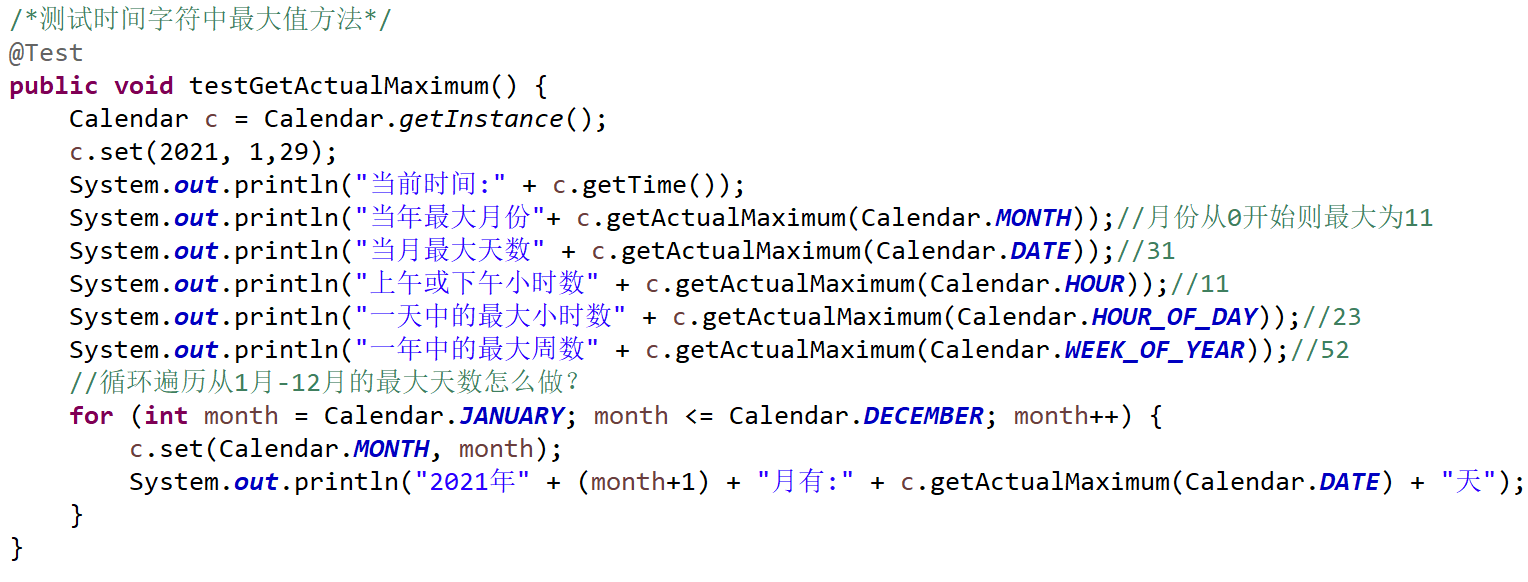
- 获取日期及时间分量



- 设置日期及时间分量

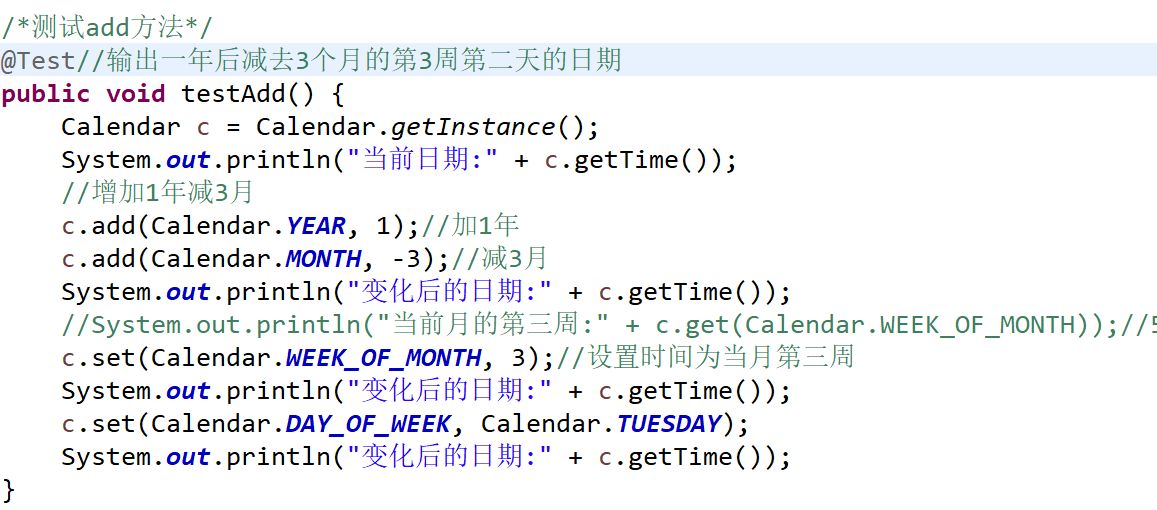


- getActualMaximum方法



- add方法

>.如果给定的时时间量为负数，则减去



- setTime和getTime方法

