Date类

Date类表示特定的瞬间，精确到毫秒

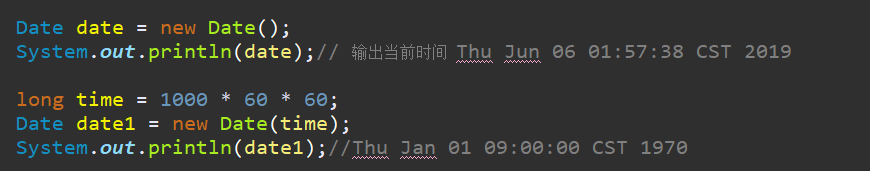
构造方法

Date()

根据当前的默认毫秒值创建日期对象

Date(long date)

根据给定的毫秒值创建日期对象



第一个，根据当前的默认毫秒值创建日期对象，这个毫秒值是1970年1月1日0点到现在的时间差

第二个，给的参数是一个小时的毫秒值，打印出的是1970年1月1日早上9点，因为我们所处东八区，时差8小时，所以8小时加1

成员方法

**public long getTime()**从date得到毫秒值

获取当前时间，以毫秒为单位

**public void setTime(long time)**从毫秒值到date

设置时间，以1970年1月1日8点为起点（东八区）

DateFormat类

一般直接输出日期格式，很难看懂，不是我们想要的

DateFormat可以进行日期和字符串的格式化和解析，但是由于是抽象类，所以使用具体子类SimpleDateFormat

SimpleDateFormat构造方法

SimpleDateFormat()

创建一个SimpleDateFormat对象（默认模式）

SimpleDateFormat(String pattern)

给定模式创建一个对象（具体模式看API）

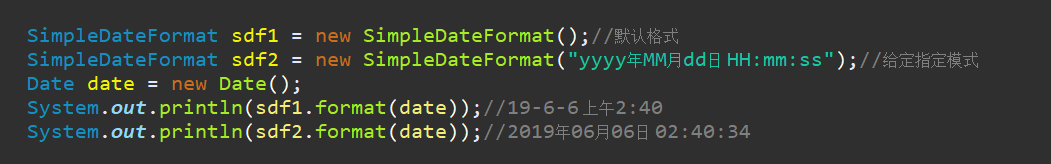
一般用格式”yyyy-MM-dd HH:mm:ss”（注意字母大小写）

DateFormat类成员方法

public final String format(Date date)

把日期类以字符串形式输出(需要用子类对象调用)

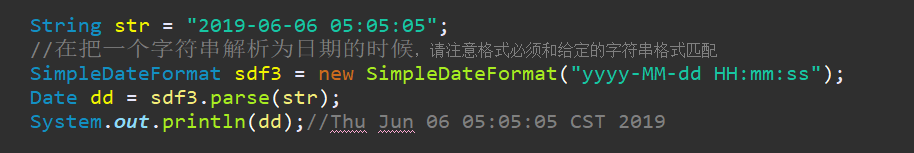
把日期格式化为字符串输出



public Date parse(String val)

把字符串解析为日期格式

把字符串解析为日期格式



Calendar类（抽象类）

Calendar 类是一个抽象类，它为特定瞬间与一组诸如 YEAR、MONTH、DAY\_OF\_MONTH、HOUR 等 日历字段之间的转换提供了一些方法，并为操作日历字段（例如获得下星期的日期）提供了一些方法。

成员方法

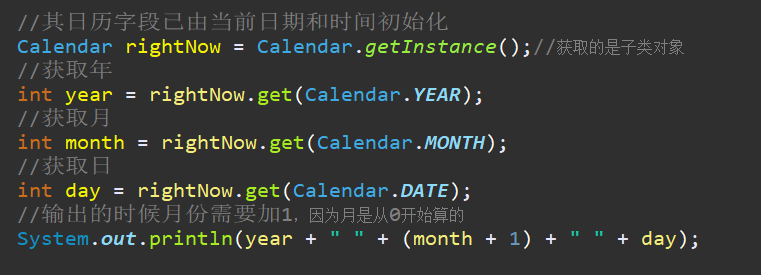
public int get(int field)

返回给定日历字段的值，日历类中的每个日历字段都是静态的成员变量，并且是int类型

因为Calendar是个抽象类，所以不能直接调用构造方法，但是其中提供了一个静态方法getInstance()方法，直接调用可以获得一个对象

Calendar rightNow = Calendar.getInstance();

底层原理其实getInstance返回的是一个子类对象



public static Calendar getInstance()

获取一个Calendar对象（其实是子类对象）

public int get(int field)

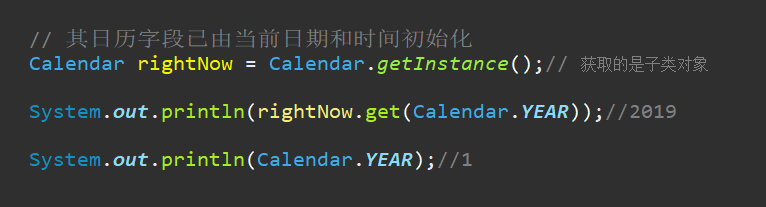
获取Calendar对象里的字段（年月日时分秒都有,都是int类型）

public void add(int field,int amount)

根据给定的日历字段和对应的时间来对当前的日历进行操作

public final void set(int year,int month,int date)

设置当前的日历的年月日



注意：YEAR,MONTH等都是静态字段，由类所有；所以需要使用对象的get()方法，才能获得需要的值，如果直接输出Calendar.YEAR则得到的是静态类属性