

1、Date 两个包中都有 java.util包 和 java.spl

1>日期对象，内部封装了一个当前系统的毫秒数（时间戳）

从1970-1-1 0:0:0 开始没往后一秒，数值加1000

2>对象创建

1) Date date = new Date();

其内部封装了当前系统的时间戳

2) Date date1 = new Date(long timeMillis);

封装指定的时间戳

3>主要方法

1)setTime() 修改内部的时间戳

2)getTime() 获取内部的时间戳

Java.text.SimpleDateFormat

作用：1、日期格式转换工具可以按照指定的格式将日期对象转成一个字符串也可以将一个字符串转为日期。

2、对象创建

SimpleDateFormat sdf = new SimpleDateFormat(String 指定格式);

格式:

| | |
|------|-------|
| yyyy | 四位数年份 |
| MM | 月份 |
| dd | 日期 |
| HH | 小时 |
| mm | 分钟 |
| ss | 秒 |
| | |

3、主要方法

1>String format(Date date)

将Date日期对象按照指定格式转为String字符串

2>Date pars(String str)

将String字符串按照指定格式转为Date日期对象

Java.util.calender

1、日历类，其内部主要封装了对日期时间进行操作的相关方法

2、对象创建

1>Calendar c = new GregorianCalendar();

创建格里高利历，封装当前系统日期时间相关信息

2>Calendar c = new Calendar.getInstance()

根据当前系统所在时区创建对象，封装系统日期相关信息

3、主要方法

1>get()

根据传入的常量字段值获取相关信息

2>getActualMaximum()

获取指定字段上的最大值

3>add()

获取在指定字段上加上一个值之后的日期

4>set()

修改Calendar对象内部的日期时间

4、主要的常量字段

1>YEAR 年

2>MONTH 月

3>DAY_OF_MONTH 日

4>DAY_OF_YEAR 一年中第几天

5>HOUR_OF_DAY 一天中第几小时

6>MINUTE 秒

7>MILLISECOND 毫秒

9>WEEK_OF_MONTH 一月中第几周

10>WEEK_OF_YEAR 一年中第几周

11>DAY_OF_WEEK 星期几

注意

1>/**获取星期**

一周中周日是第一天，周六是最后一天

*** 在系统中使用1-7来存储周日到周六**

*** 所以我们需要从系统中获取到的星期 减一**

*** 减完一之后为0 说明当天是周日**

*** 存入到系统中的时间要加1，周日就存储一***

*** 实际上： 周日 周一 周二 周三 周四 周五 周六**

*** 系统中： 1 2 3 4 5 6 7*/**

LocalDate

1、主要用于封装年月日的日期相关数据。

1>封装当前日期

`LocalDate.now();`

2>自己指定年、月、日

`LocalDate.of(2008,8,8);`

3>从字符串中解析

3.1解析固定格式

`LocalDate.parse("2013-10-5")`

从字符串中解析`LocalDate`对象

//默认只能解析yyyy-MM-dd格式的日期。

//要求年份最短是4为，最长是10位

//月份和日期如果小于10，前面必须带上0

3.2>解析自己指定格式的字符串

`LocalDate.parse(s, DateTimeFormatter.ofPattern("yyyy/M/dd"));`

3、主要方法

1>getYear() 获取年

2>getMonthValue() 获取月

3>getDayOfMonth() 获取日

4>getDayOfWeek() 获取星期

5>getDayOfYear() 获取年中的天数

6>isLeapOfYear() 判断是否位闰年

7>lengthOfMonth() 获取月中最大天数

8>plus xx() 在指定日期上增加一个值

9>minus xx() 在指定日期上减去一个值

xx:Days/Months/Years/Weeks

LocalTime

1、主要用于封装时分秒的时间相关数据

1、主要用于封装时分秒。

1>封装当前时间

`LocalTime.now();`

2>自己指定时分秒

`LocalTime.of(13,14,15);`

3>从字符串中解析

3.1解析固定格式

`LocalTime.parse("13-14-15")`

从字符串中解析`LocalTime`对象

//默认只能解析 HH:mm:ss 格式的日期。

//要求时分秒如果小于10，前面必须带上0

3.2>解析自己指定格式的字符串

```
LocalTime.parse(s, LocalTimeFormatter.ofPattern("HH/mm/ss"));
```

3、主要方法

1>getHour() 获取时

2>getMinute() 获取分

3>getSecond() 获取秒

4>plus xx() 在指定时间上增加一个值

5>minus xx() 在指定使时间上减去一个值

xx:Hours/Minutes/Seconds/Nanos

LocalDateTime

1、主要应用于封装年月日时分秒的日期时间相关数据

LocalDateTime = LocalDate + LocalTime

2、对象创建

3、主要方法

枚举

1、什么是枚举类

当一个类中只有固定的几个对象，且不能在外界随意构建这个类的对象是，我们就可以将这个类定义位枚举类，枚举类使用enum来代替

枚举的特点

1>所有的枚举类默认会继承Enum类，所有枚举类不能再通过extends显示的继承其他类。

2>枚举类不能在外界创建，所有枚举对象都是直接在枚举类中直接定义的，枚举对象名称建议使用全大写，枚举对象必须定义在其他属性或方法上边，在外界可以通过枚举类名.对象名来进行获取。

3>枚举类中默认有一个无参的私有的构造方法，自己定义的构造方法也只能是私有的。

4>枚举类型可以作用在Switch-case上。

5>枚举类中默认有一个values()静态方法来获取所有枚举对象。