

一，日期类

1, *java.lang.System*类

system类提供的public static long currentTimeMillis () 用来返回当前时间与1970年1月1日0:0:0之间的毫秒为单位的时间差

此方法计算的是时间差

- 计算世界时间的主要标准有：
 - UTC(Universal Time Coordinated)
 - GMT(Greenwich Mean Time)
 - CST(Central Standard Time)

2, *java.util.Date* (*sql下的Date类是util包下Date类的子类*)

1, *System*类下的*currentTimeMillis ()* ;

2, *Date*类: *java.util.Date*以及子类*java.sql.Date*

(不利于国际化)

创建实例:

```
Date d1 = new Date();  
long l1 = d1.getTime();  
Date d2 = new Date(l1);
```

常用方法:

getTime () ---->得到一个长整型的整数

toString () ---->将时间打印出来

3, *SimpleDateFormat*类: (*利于国际化*)

java.text.SimpleDateFormat类是一个不

与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

格式化: SimpleDateFormat():默认的模式和语言环境创建对象

SimpleDateFormat(String pattern):该构造方法可以用参

数

pattern指定的格式创建一个对象, 该对象调用

format(Date date):格式化时间对象date

解析: Date parse(String source):从给定字符串的开始解析文

本, 以生成一个日期

```

        //格式化1(18-7-21 下午9:32)
        SimpleDateFormat spdDateFormat = new SimpleDateFormat();
        String date = spdDateFormat.format(new Date());

        //格式化2(2018-34-21)
        SimpleDateFormat spDateFormat = new SimpleDateFormat("yyyy-mm-
dd");

        String date1 = spDateFormat.format(new Date());

        //解析
        SimpleDateFormat spDateFormat1 = new SimpleDateFormat("yyyy-mm-dd");
        String string = "1997-04-22";
        Date date3=null;
        try {
            date3 = spDateFormat1.parse(string);
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        System.out.println(date3);

```

//三天打渔两天晒网

```

@Test
public void test2() throws ParseException{
    String s1 = "1990-01-01";
    String s2 = "2014-05-12";
    int date = getdays(s1, s2);
    System.out.println(date);
    if (date % 5==0 || date % 5 ==4) {
        System.out.println("晒网！");
    }
    else {
        System.out.println("打渔！");
    }
}

```

```

@Test
public int getdays(String s1,String s2) throws ParseException{
    SimpleDateFormat sp = new SimpleDateFormat("uuu-aa-ff");
    Date d1 = sp.parse(s1);
    Date d2 = sp.parse(s2);
    long mis = Math.abs(d1.getTime() - d2.getTime());
    int s = (int)mis/1000/3600+1;
    return s;
}

```

4, Calendar类

calender类是一个抽象基类，主要用于完成日期字段之间相互操作的功能

①获取calender实例的方法

- a, 使用calender.getInstance () 方法
- b, 调用它的子类GreorianCalender () 的构造器

②一个calender的实例是系统时间的抽象表示，通过get (int field) 方法来获得想要的时间信息

```

Calendar calendar = Calendar.getInstance();
int day =calendar.get(Calendar.DAY_OF_MONTH);
System.out.println(day);
//添加
calendar.add(Calendar.DAY_OF_MONTH, 4);
//获取属性
day = calendar.get(Calendar.DAY_OF_MONTH);
System.out.println(day);
//设置
calendar.set(calendar.DAY_OF_MONTH, 23);
System.out.println(calendar.getTime());

```