

日期类

- java.util.Date

Date 对象表示当前的系统时间

- java.util.Calendar

Calendar 用来完成日期数据的逻辑运算

运算思路：（op+com+t）

1、将日期数据传给 Calendar（Calendar 提供了很多静态常量，专门用来记录日期数据）

常量	描述
public static final int YEAR	年
public static final int MONTH	月
public static final int DAY_OF_MONTH	天，以月为单位
public static final int DAY_OF_YEAR	天，以年为单位
public static final int HOUR_OF_DAY	小时
public static final int MINUTE	分钟
public static final int SECOND	秒
public static final int MILLSECOND	毫秒

2、调用相关方法进行运算

方法	描述
public static Calendar getInstance()	获取Calendar实例化对象
public void set(int field,int value)	给静态常量赋值
public int get(int field)	获取静态常量的值
public final Date getTime()	将Calendar转为Date对象

```

package com.southwind.demo2;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class Test {
    public static void main(String[] args) {
        //计算今天所在的周是2020年的第几周
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.YEAR,2020);
        //1月为0, 4月为3
        calendar.set(Calendar.MONTH,3);
        calendar.set(Calendar.DAY_OF_MONTH,9);
        int week =
calendar.get(Calendar.WEEK_OF_YEAR);
        System.out.println(week);
        //今天之后的63天是几月几号
        int days =
calendar.get(Calendar.DAY_OF_YEAR);
        days += 63;
        calendar.set(Calendar.DAY_OF_YEAR,days);
    }
}

```

```

        Date date = calendar.getTime();
        SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("yyyy-MM-dd");

        System.out.println(simpleDateFormat.format(date));
        //今天之前的63天是几月几号

//        calendar.set(Calendar.YEAR,2020);
//        //1月为0, 4月为3
//        calendar.set(Calendar.MONTH,3);
//        calendar.set(Calendar.DAY_OF_MONTH,9);

        calendar.set(Calendar.DAY_OF_YEAR,100);

        calendar.set(Calendar.DAY_OF_YEAR,calendar.get(Cale
ndar.DAY_OF_YEAR)-63);
        date = calendar.getTime();

        System.out.println(simpleDateFormat.format(date));
    }
}

```

输入输出流 IO 流

File 类

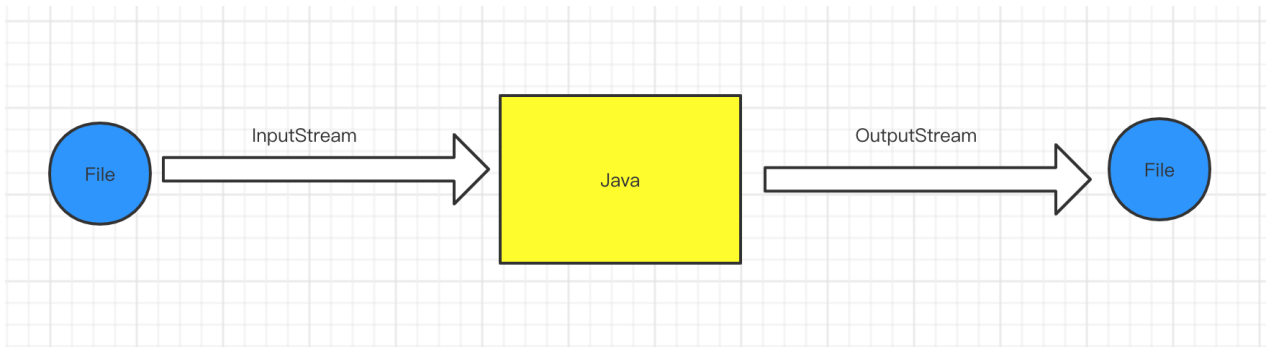
java.io.File，使用该类的构造函数就可以创建文件对象，将硬盘中的一个具体的文件以 Java 对象的形式来表示。

方法	描述
<code>public File(String pathname)</code>	根据路径创建对象
<code>public String getName()</code>	获取文件名
<code>public String getParent()</code>	获取文件所在的目录
<code>public File getParentFile()</code>	获取文件所在目录对应的File对象
<code>public String getPath()</code>	获取文件路径
<code>public boolean exists()</code>	判断文件是否存在
<code>public boolean isDirectory()</code>	判断对象是否为目录
<code>public boolean isFile()</code>	判断对象是否为文件
<code>public long length()</code>	获取文件的大小
<code>public boolean createNewFile()</code>	根据当前对象创建新文件
<code>public boolean delete()</code>	删除对象
<code>public boolean mkdir()</code>	根据当前对象创建目录
<code>public boolean renameTo(File file)</code>	为已存在的对象重命名

IO

Input 输入流（将外部文件读入到Java程序中）

Output 输出流（将Java程序中的数据输出到外部）



Java 中的流有很多种不同的分类。

- 按照方向分，输入流和输出流
- 按照单位分，可以分为字节流和字符流（字节流是指每次处理数据以字节为单位，字符流是指每次处理数据以字符为单位）
- 按照功能分，可以分为节点流和处理流。