缓冲字节流—对象流----字符流

两个抽象的类：

1. Reader：字符输入流的父类
2. Writer：字符输出流的父类

实现类

1. InputStreamReader: 字符输入流
   1. InputStreamReader(InputStream is)
   2. InputStreamReader(InputStream is, String charset)
2. OutputStreamReader: 字符输出流
   1. OutputStreamWriter(OutputStream os)
   2. OutputStreamWriter(OutputStream os, String charset)

区别字符流和字节流：Reader/Writer都是字符流

字符流：是以字符为单位读写数据（限制：只用于读写字符数据）

1. 用于读写文本，不能读写图片，视频，音频。。。（这些可以转换成字节，不能转换字符）

用字符流来封装字节流，用来加快效率

缓冲字符流（缓存区）

PrinterWriter：缓冲字符输出流

1. 不能指定字符集编码
   1. new PrinterWriter(File file)
   2. new PrinterWriter(String filePath)
   3. new PrinterWriter(OutputStream os)
2. 可以指定字符集编码
   1. new PrinterWriter(OutputStreamWriter osw)
   2. new PrinterWriter(OutputStreamWriter osw, Boolean autFlush) 具有自动刷新缓冲区的功能（最常用）
3. –print():
   1. –println():
   2. 换行占两个字符，创建具有行刷新的new PrinterWriter(osw, true)后，每当我们调用该方法println()都会自动flush将未满的缓冲区数据写出
   3. 注意：会增加读写次数，降低读写效率
   4. print()：不会进行行刷新，依然会将数据写入缓冲区中

BufferedReader：缓冲字符输入流

new BufferedReader(InputStreamReader isr)

readLine()：该方法读取一行字符数据，如果没有数据返回null

异常：

Exception：代码逻辑问题出现的一场，（需要我们去处理）

1. Runtime-Exception：运行异常，不一定捕获
2. NON-Runtime-Exception：非运行异常，这个是可以捕获的

ERROR：不允许捕获，代表编译和系统错误（必须处理）

异常处理机制：（程序员处理异常的方式）

1. 抛出throws
2. 捕获try-catch
3. 所有异常不要抛给main方法，如果抛给main程序出现异常后，main方法会将整个程序直接停止。

Try块中出现异常，被虚拟机捕获到了，而且程序员做了相应措施，如果是捕获则会在catch中会遍历异常类型，按照顺序从上到下开始遍历，直到遍历到相应的异常类型位置，这是遍历借宿跳出try-catch执行，

Try-catch后面的正常的代码

注意：在catch块中，异常排序为：小异常在上，最大的异常最后

抛出：throws，在方法上声明，提示调用者该方法可能会出现异常，让调用者进行处理，调用者处理的方式

1. 捕获try-catch
2. 抛出throws

throw（自定义异常）

Runtime-Exception：属于常见的异常，不需要进行提示，程序员能够解决的问题  
Exception：使用throws抛出异常，

1. 捕获
2. throws

finally：

和try-catch一起使用，无论try块中的代码是否出现异常finally都会被执行，通常在finally中做清理操作，比如关闭流，关闭数据库连接