**7.IO（输入输出）**

* 字节流
  + 字节流的概念
    - 在计算机中，无论是文本、图片、音频还是视频，所有文件都是以二进制（字节）形式存在的，IO流中针对字节的输入输出提供了一系列的流，统称为字节流。字节流是程序中常用的流，根据数据的传输方向可将其分为字节输入流和字节输出流。在IDK中，提供了两个抽象类InputStream和OutputStream，他们是字节流的顶级父类。
      * InputStream常用方法（描红部分为接下来重点所学）：（AudioInputStream, ByteArrayInputStream, FileInputStream, FilterInputStream（public class BufferedInputStream extends FilterInputStream）, InputStream, ObjectInputStream, PipedInputStream, SequenceInputStream, StringBufferInputStream）
      * abstract int read() ：Reads the next byte of data from the input stream.
      * int read(byte[] b) ：Reads some number of bytes from the input stream and stores them into the buffer array b.
      * int read(byte[] b, int off, int len) ： Reads up to len bytes of data from the input stream into an array of bytes.
      * void close() ： Closes this input stream and releases any system resources associated with the stream.
      * OutputStream常用方法（描红部分为接下来重点所学）：（Direct Known Subclasses:ByteArrayOutputStream, FileOutputStream, FilterOutputStream（public class BufferedOutputStream extends FilterOutputStream）, ObjectOutputStream, OutputStream, PipedOutputStream）
      * void write(byte[] b) ： Writes b.length bytes from the specified byte array to this output stream.
      * void write(byte[] b, int off, int len) ： Writes len bytes from the specified byte array starting at offset off to this output stream.
      * abstract void write(int b) ： Writes the specified byte to this output stream.
      * void flush() ： Flushes this output stream and forces any buffered output bytes to be written out.
      * void close() ： Closes this output stream and releases any system resources associated with this stream.
  + 字节流读写文件
    - 针对文件的读写，JdK专门提供了两个类，分别是FileInputStream和FileOutputStream。需要注意的是，如果要对写文件进行追加的话，FileOutputStream有如下构造方法：FileOutputStream(File file, boolean append) ： Creates a file output stream to write to the file represented by the specified File object.
  + 文件的拷贝
  + 字节流的缓冲区
    - 当通过流的方式拷贝文件时，为了提高效率也可以定义一个字节数组进行缓冲。在拷贝文件时，可以一次性读取多个字节的数据并保存在字节数组中，如后将字节数组中的数据一次性写入文件（void write(byte[] b) ： Writes b.length bytes from the specified byte array to this file output stream.）。
  + 字节缓冲流
    - 在IO包中提供两个带缓冲的字节流，分别是BufferedInputStream和BufferedOutputStream，它们的构造方法中分别接收InputStream和OutputStream类型的参数作为对象，在读写数据时提供缓冲功能。应用程序是通过缓冲来完成数据读写的，而缓冲流又是通过底层的字节流与设备进行关联的。
* 字符流
  + 字符流定义及基本用法
    - InputStream类和OutputStream类在读写文件时操作的都是字节，如果希望在程序中操作字符，使用这两个类就不太方便，为此JDK提供了字符流。同字节流一样，字符流也有两个抽象的顶级父类，分别是Reader和Writer。其中，Reader是字符输入流，用于从某个源设备读取字符。Writer是字符输出流，用于向某个目标设备写入字符。其中，FileReader和FileWriter用于读写文件，BufferedReader和BufferedWriter是具有缓冲功能的流。
    - Class Reader Direct Known Subclasses : BufferedReader, CharArrayReader, FilterReader, InputStreamReader(File Reader), PipedReader, StringReader
    - Class Writer Direct Known Subclasses : BufferedWriter, CharArrayWriter, FilterWriter, OutputStreamWriter(File Writer), PipedWriter, PrintWriter, StringWriter
  + 字符流操作文件
    - 如果想从文件中直接读取字符，便可以使用字符输入流FileReader。如果想向文件中写入字符就需要使用FileWriter类。该类是Writer的一个子类。
    - 字符流同样提供了带缓冲区的包装流，分别是BufferedReader和BufferedWriter。其中，BufferedReader用于对字符输入流进行包装，相应的，BufferedWriter对字符输出流Writer进行包装。在BufferedReader中有一个重要的方法read Line（），该方法用于一次读取一行文本。
    - 需要注意的是，由于字符缓冲流内部使用了缓冲区，因此在循环结束时一定要调用close（）方法，否则极有可能会导致部分存在缓冲区中的数据没有被写入目标文件。
  + 转换流
    - 前面提到IO流可分为字节流和字符流，有时字节流和字符流之间也需要进行转换。在JDK中提供了两个类可以将字节流转换为字符流，它们分别是InputStreamReader和OutputStreamWriter。
    - InputStreamReader(InputStream in) : Creates an InputStreamReader that uses the default charset.(可见这里的InputStreamReader的构造方法构造出来的是一个InputStreamReader对象，并不是FileReader。)，OutputStreamWriter同理。
* File类
  + File类的常用方法
    - 本章前面讲解的IO流可以对文件的内容进行写操作，在应用程序中还会经常对文件本身进行一些常规操作，例如创建一个文件、删除或重命名某个文件、判断硬盘上某个文件是否存在、查询文件最后修改时间等。针对文件的这类操作，JDK中提供了一个File类，该类封装了一个路径并提供了一系列的方法用于操作该路径所指向的文件。
      * File类常用构造方法：
      * File(File parent, String child)Creates a new File instance from a parent abstract pathname and a child pathname string.File(String pathname)Creates a new File instance by converting the given pathname string into an abstract pathname.File(String parent, String child)Creates a new File instance from a parent pathname string and a child pathname string.File(URI uri)Creates a new File instance by converting the given file: URI into an abstract pathname.
      * File类常用方法：
      * boolean delete() ： Deletes the file or directory denoted by this abstract pathname. boolean createNewFile() ： Atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist.boolean exists() ： Tests whether the file or directory denoted by this abstract pathname exists.String getName() ： Returns the name of the file or directory denoted by this abstract pathname.String getPath() ： Converts this abstract pathname into a pathname string.String getAbsolutePath() ： Returns the absolute pathname string of this abstract pathname.
  + 遍历目录下的文件
    - String[] list() ： Returns an array of strings naming the files and directories in the directory denoted by this abstract pathname.
    - public String[] list(FilenameFilter filter) ： Returns an array of strings naming the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter. The behavior of this method is the same as that of the list() method, except that the strings in the returned array must satisfy the filter. If the given filter is null then all names are accepted. Otherwise, a name satisfies the filter if and only if the value true results when the FilenameFilter.accept(File, String) method of the filter is invoked on this abstract pathname and the name of a file or directory in the directory that it denotes.（Filename filter是个接口，被称作文件过滤器，当中定义了一个抽象方法accept（File dir， String name）在调用list（）方法时，需要实现文件过滤器FilenameFilter，并在accept（）方法中作出判断，从而获得指定类型的文件。）
  + 删除文件及目录
    - 可以通过FIle类的delete()方法删除文件，但是本方法只能删除一个指定的文件，如果是目录就需要递归。