JAVA IO

赵耀

Read file

通常我们经常见到有代码这样去读一个文件：

```
strEncoding =“UTF-8"
FileInputStream fis;
fis = new FileInputStream("input.txt");
InputStreamReader isr;
isr = new InputStreamReader(fis, strEncoding);
BufferedReader br = new BufferedReader(isr);
```

甚至简化成这样：

```
BufferedReader br = new BufferedReader(new InputStreamReader(new
                        FileInputStream(fileName), "UTF-8"));
```

为什么要这么写？

一定要这么写吗？

什么时候才这么写？

什么叫流

流是一组有顺序的，有起点和终点的字节集合，是对数据传输的总称或抽象。即数据在两设备间的传输称为流，**流的本质是数据传输，根据数据传输特性将流抽象为各种类，方便更直观的进行数据操作。**

字节流与字符流

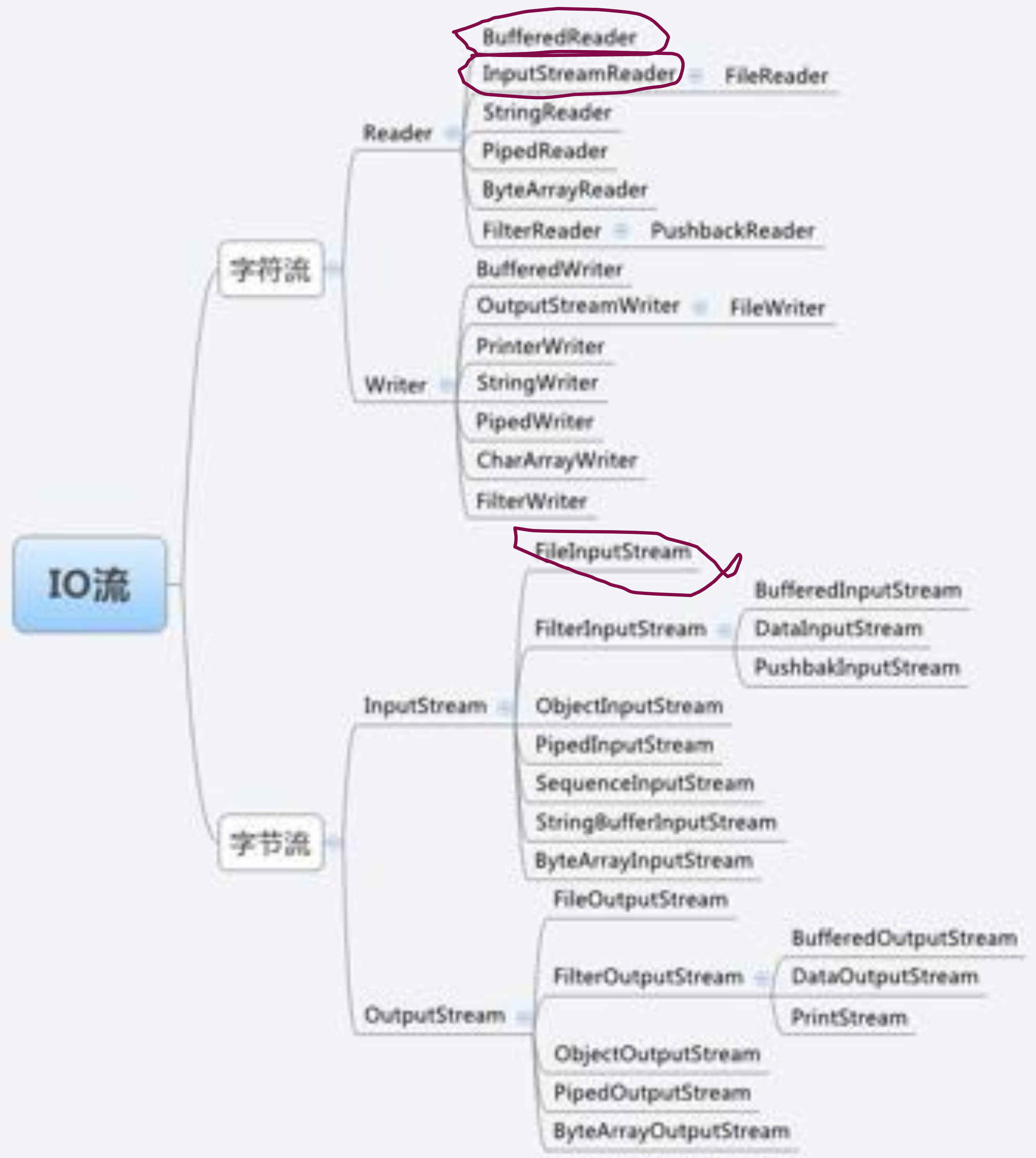
- 字节流

字节流以字节（8bit）为单位。

- 字符流

本质其实就是基于字节流读取时，去查了指定的码表（比如utf-8,utf-16）。 因为处理文本数据时，同样的字节会因为数据编码的不同，而代表不同含义的字符。字符流对象能对字符做高效操作。

JAVA IO流类图结构



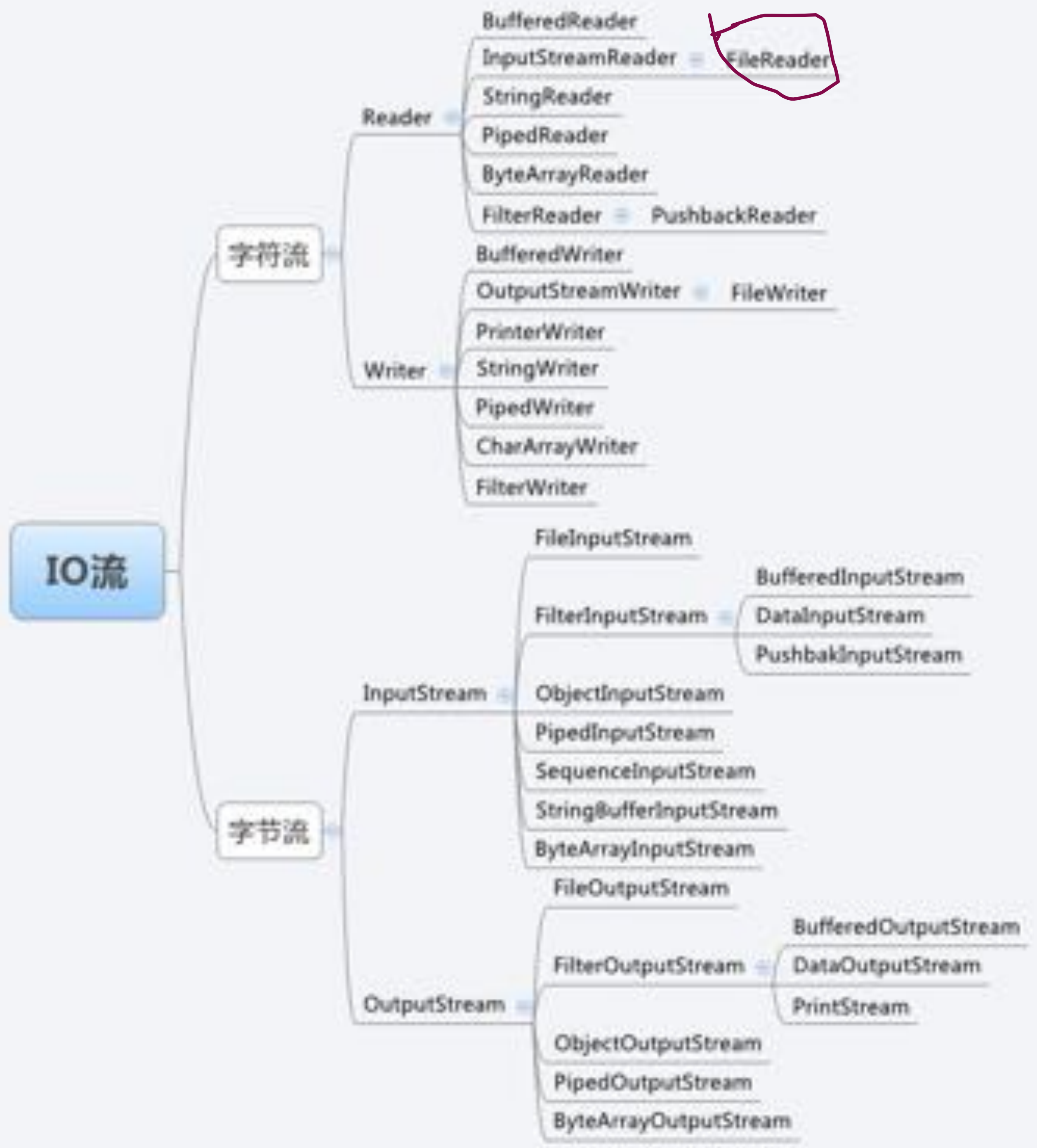
FileInputStream

- 其作用就是本地文件中以字节为单位读取数据
- 其父类： InputStream
- 其兄弟： ByteArrayInputStream、StringBufferInputStream、FileInputStream 是三种基本的介质流，它们分别从Byte 数组、StringBuffer、和本地文件中读取数据。PipedInputStream 是从与其它线程共用的管道中读取数据。

InputStreamReader

- 是一个连接字节流和字符流的桥梁，它将字节流转变为字符流。

能不能直接从文件读字符呢？



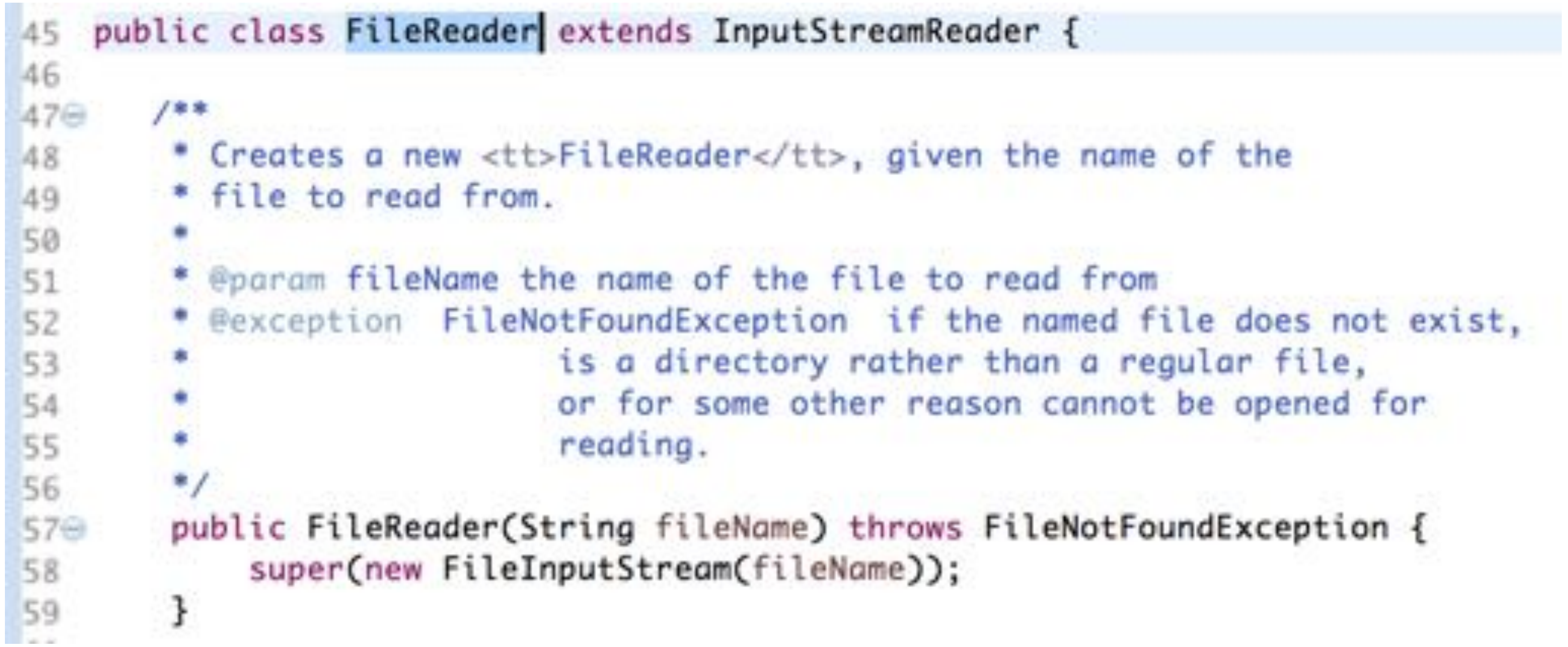
FileReader

- FileReader fr = new FileReader(”xxx.txt");
- 其实跟下面是相等的：

InputStreamReader is = new InputStreamReader(new FileInputStream(”xxx.txt"));

- 可以看下FileReader源代码

FileReader的继承及构造函数



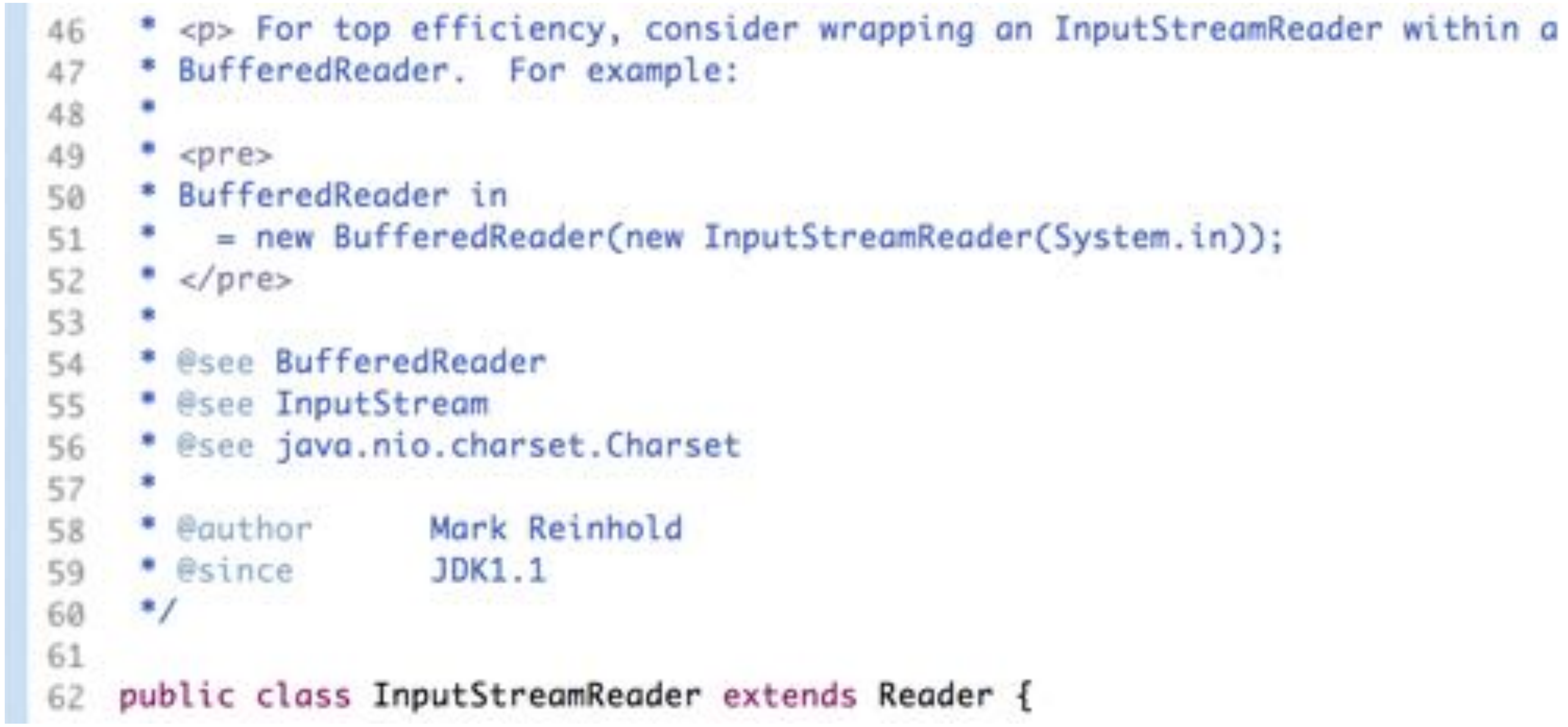
```java
public class FileReader extends InputStreamReader {

    /**
     * Creates a new <tt>FileReader</tt>, given the name of the
     * file to read from.
     *
     * @param fileName the name of the file to read from
     * @exception  FileNotFoundException  if the named file does not exist,
     *                   is a directory rather than a regular file,
     *                   or for some other reason cannot be opened for
     *                   reading.
     */
    public FileReader(String fileName) throws FileNotFoundException {
        super(new FileInputStream(fileName));
    }
```

FileReader

- 思考一下FileReader的局限性

BufferReader

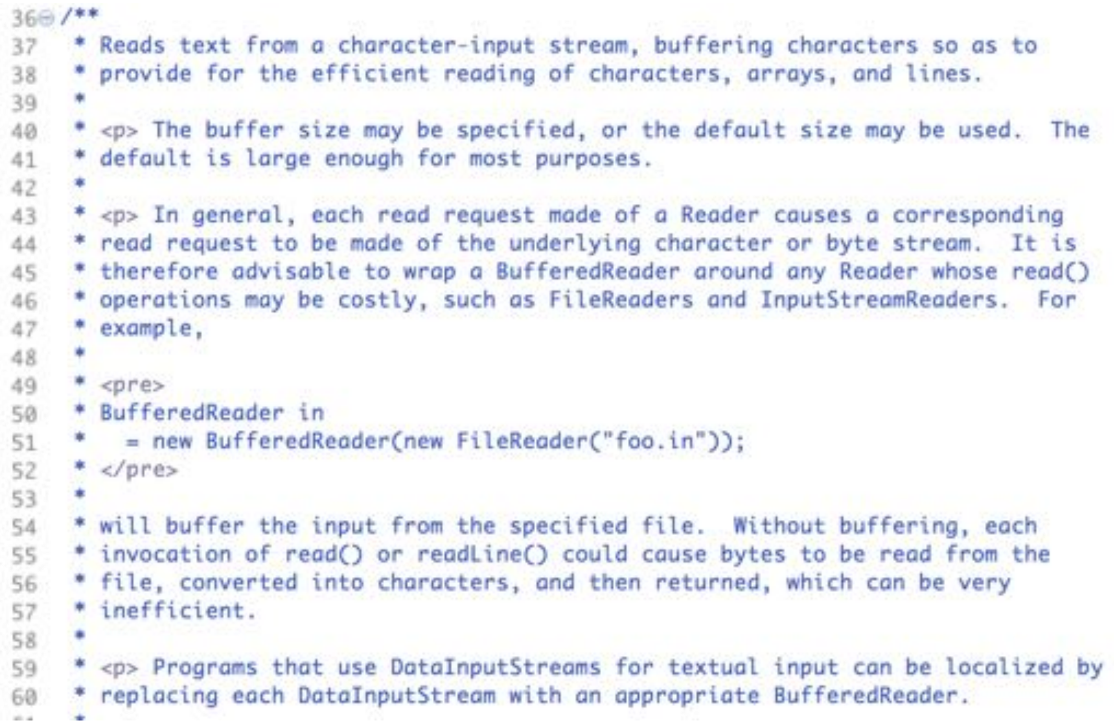
- 关于BufferReader的作用，其实InputStreamReader的注释给出了解释：



```
 * <p> For top efficiency, consider wrapping an InputStreamReader within a
 * BufferedReader.  For example:
 *
 * <pre>
 * BufferedReader in
 *   = new BufferedReader(new InputStreamReader(System.in));
 * </pre>
 *
 * @see BufferedReader
 * @see InputStream
 * @see java.nio.charset.Charset
 *
 * @author      Mark Reinhold
 * @since       JDK1.1
 */

public class InputStreamReader extends Reader {
```

BufferReader

- 关于BufferReader的作用，更详细的解释在于类定义源代码的注释中：



```
/**
 * Reads text from a character-input stream, buffering characters so as to
 * provide for the efficient reading of characters, arrays, and lines.
 *
 * <p> The buffer size may be specified, or the default size may be used.  The
 * default is large enough for most purposes.
 *
 * <p> In general, each read request made of a Reader causes a corresponding
 * read request to be made of the underlying character or byte stream.  It is
 * therefore advisable to wrap a BufferedReader around any Reader whose read()
 * operations may be costly, such as FileReaders and InputStreamReaders.  For
 * example,
 *
 * <pre>
 * BufferedReader in
 *   = new BufferedReader(new FileReader("foo.in"));
 * </pre>
 *
 * will buffer the input from the specified file.  Without buffering, each
 * invocation of read() or readLine() could cause bytes to be read from the
 * file, converted into characters, and then returned, which can be very
 * inefficient.
 *
 * <p> Programs that use DataInputStreams for textual input can be localized by
 * replacing each DataInputStream with an appropriate BufferedReader.
```

小练习

- 尝试用字节流读入一个文件，将文件内容保存在byte[]，用debug的方式看byte[]中的内容
- 尝试用字符流(传入的编码格式是正确的)读入一个文件，将文件内容保存在char[]，用debug的方式看char[]中的内容
- 尝试用字符流(传入的编码格式故意用错误的)读入一个文件，将文件内容保存在char[]，用debug的方式看char[]中的内容
- 传入一个大文件，分别使用BufferReader和不用BufferReader读入，比较效率的差别。