003.字节流和字符流的基类方法

IO体系的基类（InputStream/Reader，OutputStream/Writer）

字节流和字符流的操作方式基本一致，
只是操作的数据单元不同——字节流的操作单元是字节，字符流的操作单元是字符。
所以字节流和字符流就整理在一起了。

InputStream/Reader，OutputStream/Writer是所有输入/输出流的抽象基类，
本身并不能创建实例来执行输入/输出，
但它们将成为所有输入/输出流的模板，
所以它们的方法是所有输入/输出流都可使用的方法。

在InputStream里面包含读数据的方法。

- int read()；从输入流中读取单个字节，返回所读取的字节数据（字节数据可直接转换为int类型）。
- int read(byte[] b)从输入流中最多读取b.length个字节的数据，并将其存储在字节数组b中，返回实际读取的字节数。
- int read(byte[] b,int off,int len)；从输入流中最多读取len个字节的数据，并将其存储在数组b中，放入数组b中时，并不是从数组起点开始，而是从off位置开始，返回实际读取的字节数。

在Reader中包含也有如下3个方法。

- int read()；从输入流中读取单个字符，返回所读取的字符数据（字符数据可直接转换为int类型）。
- int read(char[] b)从输入流中最多读取b.length个字符的数据，并将其存储在字节数组b中，返回实际读取的字符数。
- int read(char[] b,int off,int len)；从输入流中最多读取len个字符的数据，并将其存储在数组b中，放入数组b中时，并不是从数组起点开始，而是从off位置开始，返回实际读取的字符数。

对比InputStream和Reader所提供的方法，就不难发现这两个基类的功能基本是一样的。

程序即可以通过read()方法每次读取一个”水滴“，
也可以通过read（char[] chuf）或者read（byte[] b）方法来读取多个“水滴”。
里面的数组就像舀水的瓢。

InputStream和Reader提供的一些移动指针的方法：

- void mark(int readlimit)；在记录指针当前位置记录一个标记（mark）。
- boolean markSupported()；判断此输入流是否支持mark()操作，即是否支持记录标记。
- void reset()；将此流的记录指针重新定位到上一次记录标记（mark）的位置。
- long skip(long n)；记录指针向前移动n个字节/字符。

readlimit 参数给出当前输入流在标记位置变为非法前允许读取的字节数。

这句话的意思是说：mark就像书签一样，用于标记，以后再调用reset时就可以再回到这个mark过的地方。mark方法有个参数，通过这个整型参数，你告诉系统，希望在读出这么多个字符之前，这个mark保持有效。比如说mark(10)，那么在read()10个以内的字符时，reset（）操作后可以重新读取已经读出的数据，如果已经读取的数据超过10个，那reset()操作后，就不能正确读取以前的数据了，因为此时mark标记已经失效。

OutputStream和Writer：

OutputStream和Writer的用法也非常相似，两个流都提供了如下三个方法：

- void write(int c)；将指定的字节/字符输出到输出流中，其中c即可以代表字节，也可以代表字符。
- void write(byte[]/char[] buf)；将字节数组/字符数组中的数据输出到指定输出流中。
- void write(byte[]/char[] buf, int off,int len)；将字节数组/字符数组中从off位置开始，长度为len的字节/字符输出到输出流中。

因为字符流直接以字符作为操作单位，所以Writer可以用字符串来代替字符数组，即以String对象作为参数。

Writer里面还包含如下两个方法：

- void write(String str)；将str字符串里包含的字符输出到指定输出流中。
- void write (String str, int off, int len)；将str字符串里面从off位置开始，长度为len的字符输出到指定输出流中。