## 3735304o第7章 IO（输入输出）

一、填空题

1．Java中的I/O流，按照传输数据的不同，可分为\_\_字节流\_\_和\_\_字符流\_\_。

2．File类中\_io\_\_包中唯一代表磁盘文件本身的对象，它定义了一些与平台无关的方法用于操作文件。

3．计算机中，无论是文本、图片、音频还是视频，所有文件都是以\_\_进制\_\_形式存在的。

4．IO提供两个带缓冲的字节流，分别是\_BufferedInputStream\_\_和\_BufferOutputStream\_\_。

5．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\_\_OutputStreamWrite\_。

二、判断题

1．如果一个File表示目录下有文件或者子目录，调用delete()方法也可以将其删除。（×）

2．File类提供了一系列方法，用于操作其内部封装的路径指向的文件或者目录，boolean exists()方法是判断文件或目录是否存在。（√）

3．JDK提供了两个抽象类InputStream和OutputStream，它们是字节流的顶级父类，所有的字节输入流都继承自OutputStream，所有的字节输出流都继承自InputStream。（×）

4．InputStreamReader是Reader的子类，它可以将一个字节输出流转换成字符输出流。（×）

三、选择题

1．下列选项中，哪些些是标准输入输出流？（ AB ）（多选）

A、System.In

B、System.Out

C、InputStream

D、OutputStream

2．File类中以字符串形式返回文件绝对路径的方法是哪一项？（ C ）

A、 getParent()

B、 getName()

C、 getAbsolutePath()

D、 getPath()

3．下列选项中，那个流使用到了缓冲技术？（A ）

A、 BuffereOutputStream

B、 FileIutputStream

C、 DataOutputStream

D、 FileReader

4．在程序开发中，经常需要对文本文件的内容进行读取，如果想从文件中直接读取字符便可以使用字符输入流（ C ）

A、Reader

B、Writer

C、FileReader

D、FileWriter

5．File类提供了一系列方法，用于操作其内部封装的路径指向的文件或者目录，当File对象对应的文件不存在时，使用哪个方法将新建的一个File对象指定到新文件中。（ C ）

A、String getAbsolutePath()

B、boolean canRead()

C、boolean createNewFile()

D、boolean exists()

四、简答题

1．简述字符流与字节流的区别。 字节流的两个基类是 InputStream 和 OutputStream，字符流的两个基类是 Reader 和Writer，它们都是 Object 类的直接子类，字节流是处理以8位字节为基本单位的字节流类；Reader 和 Writer 类是专门处理 16 位字节的字符流类。

2．简述InputStreamReader类与OutputStreamWriter类的作用。 Input Stream Reader类的作用是将字节流转换为字符流，而Output Stream Writer类的作用是将字符流转换为字节流。

五、编程题

1. 编写一个程序,分别使用字节流和字符流拷贝一个文本文件。要求如下:

（1）使用FileInputStream、FileOutputStreaem和FileReader、FileWriter分别进行拷贝。

（2）使用字节流拷贝时,定义一个1024长度的字节数组作为缓冲区，使用字符流拷贝，

import java.io.\*;

public class Main {

public static void main(String[] args)throws Exception {

FileInputStream in = new FileInputStream("E:/src.txt");

FileOutputStream out = new FileOutputStream("E:/des1.txt");

byte[] buf = new byte[1024];

int len;

while ((len=in.read(buf))!=-1){

out.write(buf,0,len);

}

in.close();

out.close();

BufferedReader bf=new BufferedReader(new FileReader("E:/src.txt"));

BufferedWriter bw=new BufferedWriter(new FileWriter("E:/des2.txt"));

String str;

while ((str=bf.readLine())!=null){

bw.write(str);

bw.newLine();

}

bf.close();

bw.close();

}

}