

1 简答题

1.1 简答题一

简述IO流的分类以及输入流和输出流作用？

答：

1.2 简答题二

请说明如下程序执行完毕以后结果并说明原因？

io-day01

E:\idea-edu-workplace\io-day01

>

>

>

itcast

itheima

>

out

>

src

io-day01.iml

>

External Libraries

Scratches and Consoles

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

package com.itheima.file;

import ...

public class FileOutputStreamDemo01 {

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

// 创建一个FileOutputStream对象

FileOutputStream fos = new FileOutputStream(new File(pathname: "itheima")) ;

// 通过字节输出流写数据

fos.write("传智播客".getBytes());

// 释放资源

fos.close();

}

答：

1.3 简答题三

如下两段程序：

// 创建基本的字节输入流对象和字节输出流对象

FileInputStream fis = new FileInputStream(name: "D:\\a.txt");

FileOutputStream fos = new FileOutputStream(name: "D:\\b.txt");

// 进行读取操作

byte[] b1 = new byte[16384] ;

int len = 0 ;

while((len = fis.read(b1)) != -1) {

fos.write(b1 , off: 0 , len);

}

// 释放资源

fos.close();

fis.close();

// 创建高效的字节输入和字节输出流对象

BufferedInputStream bis = new BufferedInputStream(new FileInputStream(name: "D:\\a.txt"));

BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream(name: "D:\\b.txt"));

// 进行读取操作

byte[] b1 = new byte[16384] ;

int len = 0 ;

while((len = bis.read(b1)) != -1) {

bos.write(b1 , off: 0 , len);

}

// 释放资源

bos.close();

bis.close();

上述程序是分别采用基本的字节输入和输出流和高效的字节输入和输出流将D:\a.txt文件中的内容复制到D:\b.txt文件中。如果a.txt文件的大小大于8192字节小于16384个字节，那么哪一种方式

复制文件的效率较高并说明原因？

答：

1.4 简答题四

Java中的字节流是可以操作任意类型的数据(音频、视频、文本...),请简述字符流出现的意义？

答：

1.5 简答题五

请将如下程序分别在idea和dos控制台进行执行，分析输出不同结果的原因？如何解决dos控制台结果的输出问题？

// 对"中国"进行编码

String s1 = "中国" ;

byte[] bytes = s1.getBytes();

// 对上述得到的字节数组进行编码

String s2 = new String(bytes , "UTF-8") ;

System.out.println(s2);

答：

2 编程题

2.1 编程题目一

训练目标：掌握java中字节输入流和字节输出流的基本使用，以及理解其在实际开发中的应用

需求背景：为了保证磁盘文件数据的安全性，就需要对原始文件进行加密存储，再使用的时候再对其进行解密处理。加密解密原理：对原始文件中的每一个字节数据进行更改，然后将更改以后的

数据存储到新的文件中。

需求描述：编写一个加密解密程序。

实现提示：

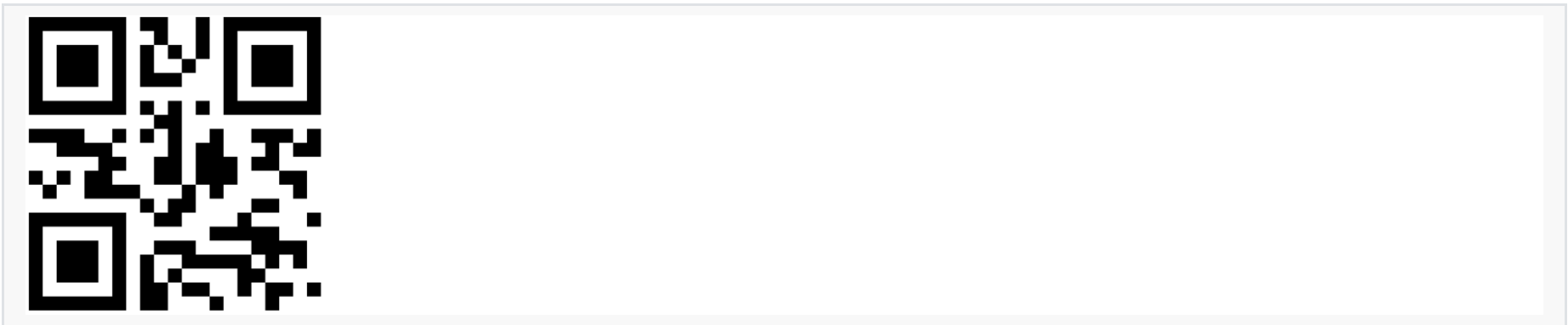
- 1、通过字节输入流读取原始文件中的每一个字节数据然后对其进行处理
- 2、通过字节输出流将处理以后的字节数据写入到另外一个文件中
- 3、对字节数据的处理方式可以使用异或运算符异或一个数
- 4、为了提高加解密的效率可以考虑使用高效的字节输入和输出流

2.2 编程题目二

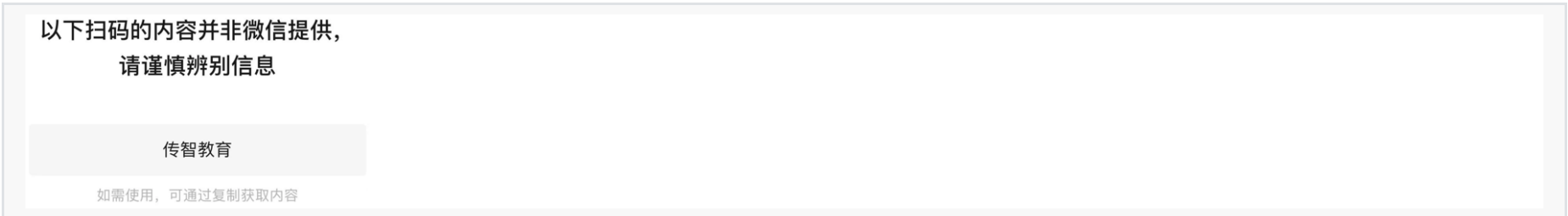
训练目标：掌握java中字节输入流和字节输出流的基本使用，以及理解其在实际开发中的应用

需求背景：二维码又称二维条码，常见的二维码为QR Code，QR全称Quick Response，是一个近几年来移动设备上超流行的一种编码方式，它比传统的Bar Code条形码能存更多的信息，也能

表示更多的数据类型。下方给大家展示的是一个个人信息的二维码：



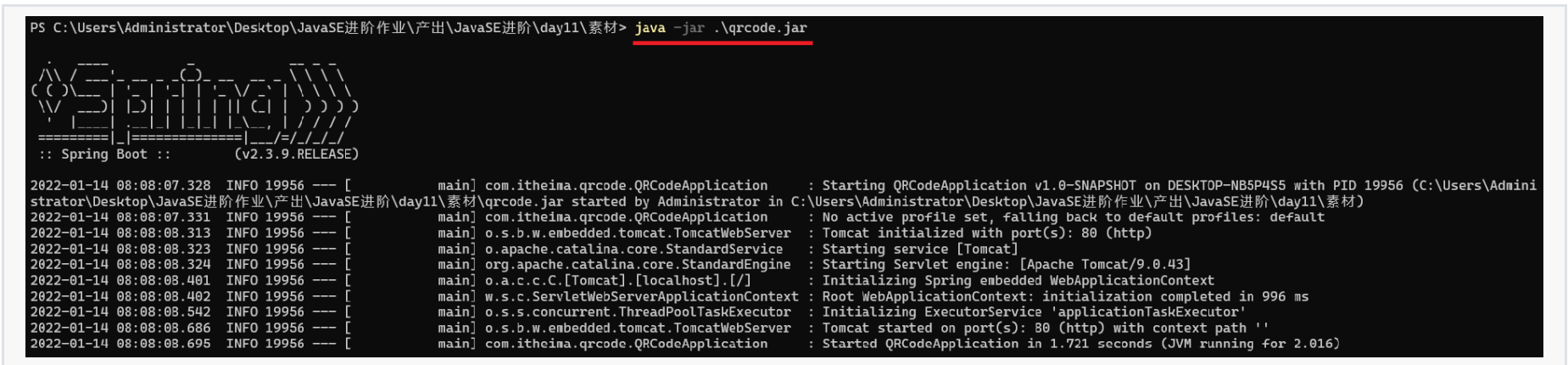
使用微信扫描该二维码，可以得到二维码所承载数据，如下所示：



需求描述：在素材的code项目中指定位置补全代码，获取个人信息二维码保存到本地磁盘。

实现提示：

- 1、生成二维码的程序已经在素材文件中提供好了(qrcode.jar)，在dos控制台通过 `java -jar qrcode.jar` 启动该程序。效果如下所示：



- 2、在code项目中，通过程序请求生成二维码程序获取字节输入流代码已经提供
- 3、保存二维码数据到本地文件，这个本地文件的后缀名可以为png

