**每日作业卷**

**JavaSE第9天**

传智播客.黑马程序员

# 基础题

## 练习一:字节输出流写出字节数据

描述:利用字节输出流一次写一个字节的方式，向D盘的a.txt文件输出字符‘a’。

答案

操作步骤:

1. 创建字节输出流FileOutputStream对象并指定文件路径。
2. 调用字节输出流的write(int byte)方法写出数据

代码:

**public class** Test01\_01 {

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

// 1.创建字节输出流FileOutputStream对象并指定文件路径。

FileOutputStream fos = **new** FileOutputStream("d:/a.txt");

// 2.调用字节输出流的write(int byte)方法写出数据

fos.write(97);

// 3.关闭流

fos.close();

}

}

## 练习二:字节输出流写出字节数组数据

描述:利用字节输出流一次写一个字节数组的方式向D盘的b.txt文件输出内容:"i love java"。

答案

操作步骤:

1. 创建字节输出流FileOutputStream对象并指定文件路径。
2. 调用字节输出流的write(byte[] buf)方法写出数据。

代码:

**public class** Test01\_02 {

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

// 1.创建字节输出流FileOutputStream对象并指定文件路径。

FileOutputStream fos = **new** FileOutputStream("d:/b.txt");

// 2.调用字节输出流的write(byte[] buf)方法写出数据。

**byte**[] buf = "i love java".getBytes();

fos.write(buf);

// 3.关闭资源

fos.close();

}

}

## 练习三:文件的续写和换行输出

描述:在D盘下，有一c.txt 文件中内容为：HelloWorld

在c.txt文件原内容基础上，添加五句 I love java，而且要实现一句一行操作(注：原文不可覆盖)。

利用字节输出流对象往C盘下c.txt文件输出5句：”i love java”

答案

操作步骤:

1. 利用两个参数的构造方法创建字节输出流对象，参数一指定文件路径，参数二指定为true
2. 调用字节输出流的write()方法写入数据，在每一行后面加上换行符:”\r\n”

代码:

**public class** Test01\_03 {

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

// 1.创建字节输出流FileOutputStream对象并指定文件路径,并追加方式

FileOutputStream fos = **new** FileOutputStream("c:/c.txt",**true**);

// 2.调用字节输出流的write方法写出数据

// 2.1 要输出的字符串

String content = "i love java \r\n";

**for** (**int** i = 0; i< 5; i++) {

fos.write(content.getBytes());

}

// 3.关闭流

fos.close();

}

}

## 练习四:字节输入流一次读取一个字节数据

描述:利用字节输入流读取D盘文件a.txt的内容，文件内容确定都为纯ASCII字符

,使用循环读取，一次读取一个字节，直到读取到文件末尾。将读取的字节输出到控制台

答案

操作步骤:

1. 创建字节输入流对象指定文件路径。
2. 调用read(byte b)方法循环读取文件中的数据
3. 直到读取到-1时结束读取

代码:

**public class** Test01\_04 {

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

// 创建字节输入流对象并关联文件

FileInputStream fis = **new** FileInputStream("d:/a.txt");

// 定义变量接收读取的字节

**int** len = -1;

// 循环从流中读取数据

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

System.***out***.print((**char**)len);

}

// 关闭流

fis.close();

}

}

## 练习五:字节输入流一次读取一个字节数组数据

描述:利用字节输入流读取D盘文件b.txt的内容，文件内容确定都为纯ASCII字符

,使用循环读取，一次读取一个字节数组，直到读取到文件末尾，将读取到的字节数组转换成字符串输出到控制台。

答案

操作步骤:

1. 创建字节输入流对象指定文件路径。
2. 定义一个字节数数组，用来存放读取的字节数
3. 调用read(byte[] buf)方法传入字节数组，循环读取文件中的数据
4. 直到读取到-1时结束读取

代码:

**public class** Test01\_05 {

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

// 创建字节输入流对象并关联文件

FileInputStream fis = **new** FileInputStream("d:/b.txt");

// 定义字节数组存放读取的字节数

**byte**[] buffer = **new byte**[1024];

// 定义变量接收读取的字节

**int** len = -1;

// 循环从流中读取数据

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

System.***out***.print(**new** String(buffer,0,len));

}

// 关闭流

fis.close();

}

}

## 练习六:字节流复制文件

描述:利用字节流将E盘下的a.png图片复制到D盘下(文件名保存一致)

要求：

一次读写一个字节的方式

答案

操作步骤:

1. 创建字节输入流对象关联文件路径：E盘下的a.png
2. 创建字节输出流对象关联文件路径：D盘下的a.png
3. 使用循环不断从字节输入流读取一个字节，每读取一个字节就利用输出流写出一个字节。
4. 关闭流，释放资源

代码:

**public class** Test01\_06 {

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

// 创建字节输入流对象并关联文件

FileInputStream fis = **new** FileInputStream("e:/a.png");

// 创建字节输出流对象并关联文件

FileOutputStream fos = **new** FileOutputStream("d:/a.png");

// 定义变量接收读取的字节数

**int** len = -1;

// 循环读取图片数据

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

// 每读取一个字节的数据就写出到目标文件中

fos.write(len);

}

// 关闭流

fis.close();

fos.close();

}

}

## 练习七:字符输出流写出字符数据

项目需求：请用户从控制台输入信息，程序将信息存储到文件Info.txt中。可以输入多条信息，每条信息存储一行。当用户输入：”886”时，程序结束。

答案

操作步骤:

1. 创建MainAPP类,并包含main()方法
2. 按照上述要求实现程序

代码:

p**ublic class** Test01\_07 {

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

//1. 指定输出流， 对应的文件Info.txt

FileWriter bw**= new** FileWriter("Info.txt");

//2.采用循环的方式，把每条信息存储一行到Info.txt中

Scanner sc= **new** Scanner(System.in);

while(true){

//获取键盘输入的一行内容

System.out.print("请输入内容：");

String str= sc.nextLine();

//当用户输入：”886”时，程序结束。

if ("886".equals(str)) {

break;//跳出循环

}

//把内容写入到Info.txt文件中

bw.write(str);

//换行

bw.write(System.lineSeparator());

}

//关闭流

bw.close();

}

}

## 练习八:IO对象Properties结合使用,设置properties文件

描述:

我有一个文本文件score.txt，我知道数据是键值对形式的，但是不知道内容是什么。

请写一个程序判断是否有"lisi"这样的键存在，如果有就改变其实为"100"

score.txt文件内容如下：

zhangsan = 90

lisi = 80

wangwu = 85

答案

操作步骤:

1. 创建一个空的Properties集合
2. 读取数据到集合中
3. 遍历集合，获取到每一个key
4. 判断当前的key 是否为 "lisi"，如果是就把"lisi"的值设置为100
5. 把集合中所有的信息，重新存储到文件中

提示信息：

把集合中的信息，存储到文件中，可以用如下方法。

java.util 类 Properties

|  |  |
| --- | --- |
| void | **[store](mk:@MSITStore:C:\\Users\\admin\\Desktop\\参考手册\\JDK_API_1_6_zh_CN.CHM::/java/util/Properties.html" \l "store(java.io.OutputStream, java.lang.String))**([OutputStream](mk:@MSITStore:C:\\Users\\admin\\Desktop\\参考手册\\JDK_API_1_6_zh_CN.CHM::/java/io/OutputStream.html" \o "java.io 中的类) out, [String](mk:@MSITStore:C:\\Users\\admin\\Desktop\\参考手册\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) comments)           以适合使用[load(InputStream)](mk:@MSITStore:C:\\Users\\admin\\Desktop\\参考手册\\JDK_API_1_6_zh_CN.CHM::/java/util/Properties.html" \l "load(java.io.InputStream))方法加载到Properties表中的格式，将此Properties表中的属性列表（键和元素对）写入输出流。 |

参数：

out - 输出流。

comments - 属性列表的描述。

代码:

**public class** Test02\_06 {  
**public static void** main(String[] args) **throws** IOException {  
*//1：创建一个空的集合*Properties prop = **new** Properties();  
*//2:读取数据到集合中*prop.load(**new** FileInputStream(**"score.txt"**));  
*//3:遍历集合，获取到每一个key*Set<String> keys = prop.stringPropertyNames();  
*//获取到每一个key***for** (String key : keys) {  
*//4:判断当前的key 是否为 "lisi"***if** (**"lisi"**.equals(key)) {  
*//把"lisi"的值设置为100*prop.setProperty(key, **"100"**);  
 }  
 }  
*//把集合中所有的信息，重新存储到文件中*prop.store(**new** FileOutputStream(**"score.txt"**), **"haha"**);  
 }  
}

# 扩展题

## 练习一:字节输入流使用

描述:

在D盘下有一个文本文件test.txt(里面的内容由数字和字母组成)

定义一个方法统计test.txt文件中’a’字符出现的次数。

比如a字符在文件中出现了10次则调用方法传入a后，方法内部输出：a出现10次

答案

操作步骤:

1. 创建字节输入流对象，循环从文件中读取一个字节
2. 定义一个整数变量用来统计字符出现的次数。
3. 将读取的字节转换字符跟传入的字符进行比较，相同则计数加一。
4. 输出结果。

代码:

import java.io.FileInputStream;

import java.io.IOException;

public class Test02\_01{

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

// 调用方法

calcuteCount('a');

}

/\*

\* 统计字符在文件中出现的次数

\*/

public static void calcuteCount(char ch) throws IOException {

// 创建字节输入流

final FileInputStream fis = new FileInputStream("d:/test.txt");

try (fis) {

// 定义一个计数变量，统计字符出现的次数

int count = 0;

// 循环读取数据

int len = -1;

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

// 判断字符c和外界传入的字符是否相同

if (len == ch) {

count++;

}

}

System.out.println(ch + "出现了" + count + "次");

} catch (IOExceptione) {

e.printStackTrace();

}

}

}

## 练习二:字节输出流写出数据

描述:

从控制台循环接收用户录入的学生信息，输入格式为：学号-学生名字

将学生信息保存到D盘下面的stu.txt文件中，一个学生信息占据一行数据。

当用户输入end时停止输入。

答案

操作步骤:

1. 使用Scanner类进行键盘录入数据
2. 创建字节输入流对象关联目标文件
3. 使用死循环不停的接收用户输入的学生数据
4. 接收用户输入的学生数据
5. 判断输入的内容是否是end，是则终止循环，停止输入。否则就数据写出到文件中。

代码:

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Scanner;

public class Test02\_02{

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

// 创建字节输出流对象

final FileOutputStream fos = new FileOutputStream("d:/stu.txt");

// 创建键盘录入对象

Scanner sc = new Scanner(System.in);

System.out.println("请输入学生信息，格式：学号-姓名");

try (fos; sc) {

// 使用死循环不停的接收用户输入的学生数据

while (true) {

// 接收用户输入的学生数据

String line = sc.nextLine();

// 判断输入的内容是否是end，是则终止循环

if ("end".equals(line)) {

break;

}

// 将数据写出到文件中。

fos.write(line.getBytes());

// 写出换行符号

fos.write(System.lineSeparator().getBytes());

}

} catch (IOExceptione) {

e.printStackTrace();

}

}

}

## 练习三:字符输出流写出字符数据并存到集合中

需求说明：从控制台接收3名学员的信息，每条信息存储到一个Student对象中，将多个Student对象存储到一个集合中。输入完毕后，将所有学员信息存储到文件Student.txt中。每名学员信息存储一行，多个属性值中间用逗号隔开。

答案

操作步骤:

1. 创建Student类，有如下属性：

学号、姓名、性别、年龄

全部属性使用String类型。要求有无参，全参构造方法。所有属性私有，并提供公有get/set方法。

1. 创建MainApp类，包含main()方法
2. 在main()方法中：
3. 定义一个存储Student对象的集合；
4. 循环3次，从控制台接收3名学员信息，每条信息封装一个Student对象，将每个Student对象存储到集合中。
5. 遍历集合，获取每个Student对象，取出所有属性值，输出到文件Test2\_2.txt中。每名学员信息占一行。

代码:

public class Task02\_03 {

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

// 1.定义学生类, 定义存学生的集合

ArrayList<Student> list = new ArrayList<Student>();

// 2.通过3次循环，完成如下操作

Scanner sc = new Scanner(System.in);

for (int i = 1; i<= 3; i++) {

// 键盘输入学生的信息，

System.out.print("请输入第" + i + "名学生的学号：");

String id = sc.next();

System.out.print("请输入第" + i + "名学生的姓名：");

String name = sc.next();

System.out.print("请输入第" + i + "名学生的性别：");

String sex = sc.next();

System.out.print("请输入第" + i + "名学生的年龄：");

String age = sc.next();

// 把信息封装到Student对象中

Student s = new Student(id, name, sex, age);

// 把Student对象存到集合里

list.add(s);

}

// 3.将所有学员信息存储到文件Student.txt中。

FileWriter out = new FileWriter("Student.txt");

// 每名学员信息存储一行，多个属性值中间用逗号隔开。

for (int i = 0; i<list.size(); i++) {

// 1.获取集合中每一个学生对象

Student s = list.get(i);

// 2.获取对象中的每一个属性值,多个属性值中间用逗号隔开

String line = s.getId() + "," + s.getName() + "," + s.getSex() + "," + s.getAge();

// 3.按照指定的格式把对象的属性值，写入到文件中

out.write(line);

out.write(System.lineSeparator());

}

out.close();// 关闭流

}

}