**每日作业卷答案**

**JavaSE第12天**

# 关卡1

## 训练案例1

### 训练描述

从键盘接收用户输入的一行字符串，将这行字符串写入到C盘下a.txt文件中

要求：一次写出一个字符。

### 操作步骤答案

**public** **class** Test01\_01 {

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

// 创建键盘录入对象

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

System.***out***.println("请输入一行字符串：");

// 接收用户输入的内容

String line = sc.nextLine();

// 创建字符流输出流对象并关联目标文件

FileWriter fw = **new** FileWriter("c:/a.txt");

// 使用循环一次写出一个字符

**for** (**int** index = 0; index < line.length(); index++) {

fw.write(line.charAt(index));

}

// 关闭流释放资源

fw.close();

}

}

## 训练案例2

### 训练描述

键盘接收用户输入的一行字符串，将这行字符串写入到C盘下b.txt文件中

要求：一次写出一个字符数组。

### 操作步骤答案

**public** **class** Test01\_02 {

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

// 创建键盘录入对象

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

System.***out***.println("请输入一行字符串：");

// 接收用户输入的内容

String line = sc.nextLine();

// 创建字符流输出流对象并关联目标文件

FileWriter fw = **new** FileWriter("c:/b.txt");

// 将字符串写入到文件中

fw.write(line.toCharArray());

// 关闭流释放资源

fw.close();

}

}

## 训练案例3

### 训练描述

在C盘下创建一个c.txt文件，往该文件中输入多行文本数据，利用字符输入流读取该文件的内容并输入到控制台上。

要求使用两种方式读取：

\* 一次读取一个字符

\* 一次读取一个字符数组。

### 操作步骤答案

**public** **class** Test01\_03 {

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

// 一次读取一个字符

*readChar*();

// 一次读取一个字符数组。

*readCharArray*();

}

/\*

\* 一次读取一个字符数组。

\*/

**public** **static** **void** readCharArray() **throws** IOException {

// 创建字符串输入流对象

FileReader fr = **new** FileReader("c.txt");

// 定义变量接收实际读取的字符数

**int** len = -1;

// 定义字符数组存放读取的字符

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

// 循环读取内容

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

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

}

// 关闭流释放资源

fr.close();

}

/\*

\* 一次读取一个字符

\*/

**public** **static** **void** readChar() **throws** IOException {

// 创建字符串输入流对象

FileReader fr = **new** FileReader("c.txt");

// 定义变量接收读取的字符数据

**int** len = -1;

// 循环读取内容

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

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

}

// 关闭流释放资源

fr.close();

}

}

## 训练案例4

### 训练描述

现有一字符串：”我爱Java”。将该字符串保存到当前项目根目录下的a.txt文件中。

要求：使用utf8编码保存。

### 操作步骤答案

**public** **class** Test01\_04 {

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

// 要保存的字符串

String content = "我爱Java";

// 创建字节输出流对象

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

// 创建转换输出流对象

OutputStreamWriter osw = **new** OutputStreamWriter(fos, "utf8");

// 调用方法写出数据

osw.write(content);

// 关闭流释放资源

osw.close();

}

}

## 训练案例5

### 训练描述

利用转换输入流将当前项目根目录下使用utf8编码的a.txt文件的内容读取出来，并打印在控制台上。

要求：不能出现乱码的情况。

### 操作步骤答案

**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("a.txt");

// 创建转换输入流对象

InputStreamReader isr = **new** InputStreamReader(fis,"utf8");

// 定义字符数组存放读取的内容

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

// 定义变量接收读取的字符个数

**int** len = -1;

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

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

}

// 关闭流

isr.close();

}

}

## 训练案例6

### 训练描述

定义一个学生类，包含姓名，年龄，性别等成员变量，提供setters和getters方法以及构造方法。在测试类中创建一个学生对象，给学生对象的三个成员变量赋值。然后将该对象保存到当前项目根目录下的stu.txt文件中。

### 操作步骤答案

**public** **class** Test01\_06 {

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

// 创建学生对象

Student s = **new** Student("jack",20,"男");

// 创建对象输出流对象

ObjectOutputStream oos = **new** ObjectOutputStream(**new** FileOutputStream("stu.txt"));

// 将学生对象保存文件中

oos.writeObject(s);

// 关闭流并释放资源

oos.close();

}

}

**class** Student **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 4983659706961705248L;

**private** String name;

**private** **int** age;

**private** String gender;

**public** Student() {

}

**public** Student(String name, **int** age, String gender) {

**this**.name = name;

**this**.age = age;

**this**.gender = gender;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

}

## 训练案例7

### 训练描述

将上一题保存到stu.txt文件中的学生对象读取出来。

### 操作步骤答案

**public** **class** Test01\_07 {

**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("stu.txt");

// 创建对象输入流对象

ObjectInputStream ois = **new** ObjectInputStream(fis);

// 读取学生对象

Student s = (Student) ois.readObject();

System.***out***.println(s);

// 关闭流

ois.close();

}

}

## 训练案例8

### 训练描述

从键盘录入一行字符串，利用字节打印流将该行字符串保存到当前项目根目录下的d.txt文件中

### 操作步骤答案

**public** **class** Test01\_08 {

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

// 创建键盘录入对象

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

System.***out***.println("请输入一行字符串：");

// 接收用户输入的数据

String line = sc.nextLine();

// 创建字节打印流对象

PrintStream ps = **new** PrintStream("d.txt");

// 将字符串写出到文件中

ps.println(line);

// 关流

ps.close();

}

}

## 训练案例9

### 训练描述

利用字符打印流将字符串”我爱Java” 保存到当前项目的根目录下的e.txt文件中。

### 操作步骤答案

**public** **class** Test01\_09 {

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

// 创建字符打印流对象并关联文件

PrintWriter pw = **new** PrintWriter("e.txt");

// 要保存的内容

String content = "我爱Java";

// 调用打印流的方法保存字符串

pw.println(content);

// 刷新缓冲区

pw.flush();

// 关闭流

pw.close();

}

}

## 训练案例10

### 训练描述

定义一个集合用户存放多个字符串，利用打印流将集合中的字符串输出到当前项目根目录下的array.txt文件中，要求一个字符串占据一行。

### 操作步骤答案

**public** **class** Test01\_10 {

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

// 创建集合存放字符串

ArrayList<String> strs = **new** ArrayList<String>();

// 添加多个字符串

strs.add("zhangsan");

strs.add("lisi");

strs.add("wangwu");

// 创建打印流对象

PrintWriter pw = **new** PrintWriter(**new** FileOutputStream("array.txt"), **true**);

// 遍历集合，将集合元素输出到文件中

**for** (String str : strs) {

pw.println(str);

}

// 关闭流释放资源

pw.close();

}

}

# 关卡2

## 训练案例1

### 训练描述

在当前项目根目录下准备好一个test.txt 的文本文件，要求该文本文件是使用GBK编码，其内容如下：

窗前明月光

疑是地上霜

举头望明月

低头思故乡

利用字节流+桥转换读入这个文本文件，以UTF8的编码方式将读取的内容写到test2.txt文件中

### 操作步骤答案

**public** **class** Test02\_01 {

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

// 创建字节输入流对象关联要读取的文件

FileInputStream in = **new** FileInputStream("test.txt");

// 创建转换输入流对象

InputStreamReader isr = **new** InputStreamReader(in);

// 创建字节输出流对象关联要保存的文件

FileOutputStream os = **new** FileOutputStream("test2.txt");

// 创建转换输出流对象

OutputStreamWriter osw = **new** OutputStreamWriter(os,"utf8");

// 定义字符数组存放读取数据

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

// 定义变量接收读取的字符数

**int** len = -1;

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

// 利用输出流将读取内容保存到文件中

osw.write(buffer, 0, len);

}

// 关闭流释放资源

isr.close();

osw.close();

}

}

## 训练案例2

### 训练描述

定义一个学生类，成员变量有姓名，年龄，性别，提供setters和getters方法以及构造方法

定义一个测试类，在测试类创建多个学生对象保存到集合中，然后将集合存储到当前项目根目录下的stus.txt文件中。

### 操作步骤答案

**public** **class** Test02\_02 {

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

// 创建集合对象，存放学生对象

List<Student> stus = **new** ArrayList<Student>();

// 添加学生对象

stus.add(**new** Student("jack",20,"男"));

stus.add(**new** Student("rose",18,"女"));

stus.add(**new** Student("laowang",20,"男"));

// 创建对象输出流对象

ObjectOutputStream oos = **new** ObjectOutputStream(**new** FileOutputStream("stus.txt"));

// 将学生对象保存文件中

oos.writeObject(stus);

// 关闭流并释放资源

oos.close();

}

}

## 训练案例3

### 训练描述

将上一题保存到stus.txt文件中集合对象读取出来并遍历集合将学生对象输出在控制台

### 操作步骤答案

**public** **class** Test02\_03 {

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

// 创建字节输入流对象并关联文件

FileInputStream fis = **new** FileInputStream("stus.txt");

// 创建对象输入流对象

ObjectInputStream ois = **new** ObjectInputStream(fis);

// 读取集合对象

List<Student> stus = (List<Student>) ois.readObject();

// 关闭流

ois.close();

// 遍历集合

**for** (Student stu : stus) {

System.***out***.println(stu);

}

}

}

## 训练案例4

### 训练描述

利用字符打印流将C盘下的ccc.txt文件复制到d盘下

### 操作步骤答案

**public** **class** Test02\_04 {

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

// 创建字符打印流对象关联目标文件

PrintWriter pw = **new** PrintWriter("d:/ccc.txt");

// 创建高效字符输出流对象

BufferedReader br = **new** BufferedReader(**new** FileReader("c:/ccc.txt"));

// 定义字符串接收读取的数据

String line = **null**;

// 从输入流中循环读取数据

**while**((line = br.readLine()) != **null**) {

// 利用打印流将数据写出到目标文件中

pw.println(line);

}

// 关闭资源

pw.close();

br.close();

}

}

## 训练案例5

### 训练描述

将C盘下的a.png图片复制到D盘下

### 操作步骤答案

**public** **class** Test02\_05 {

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

// 创建字节打印流对象关联目标文件

PrintStream ps = **new** PrintStream("b.png");

// 创建高效字节输出流对象

BufferedInputStream bis = **new** BufferedInputStream(**new** FileInputStream("a.png"));

// 定义字节数组接收读取的数据

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

**int** len = -1;

// 从输入流中循环读取数据

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

// 利用打印流将数据写出到目标文件中

ps.write(buffer,0,len);

}

// 关闭资源

ps.close();

bis.close();

}

}

## 训练案例6

### 训练描述

已知数组int arr[] = {2,3,4,2,4};

(1)计算出该数组的平均分、最大值、最小值

(2)并将大于平均分的元素放到当前根目录下number.txt中(使用打印流,以追加的方式添加)

### 操作步骤答案

**public** **class** Test02\_06 {

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

// 已知数组i

**int** arr[] = { 2, 3, 4, 2, 4 };

// 计算平均分需要先算出总分

// 统计总分

**int** sum = 0;

// 记录最大值

**int** max = arr[0];

// 记录最小值

**int** min = arr[0];

// 遍历数组

**for** (**int** num : arr) {

// 求和

sum += num;

// 如果num比max大,修改max为num

**if** (max < num) {

max = num;

}

// 如果min > num, 修改min为num

**if** (min > num) {

min = num;

}

}

// 计算平均值

**int** avg = sum / arr.length;

System.***out***.println("平均值=" + avg + ",最大值=" + max + ",最小值=" + min);

// (2)并将大于平均分的元素放到当前根目录下number.txt中(使用打印流,以追加的方式添加)

// 创建打印流

PrintWriter pw = **new** PrintWriter(**new** FileWriter("number.txt", **true**));

// 遍历数组

**for** (**int** num : arr) {

**if** (num > avg) {

pw.println(num);

}

}

// 关闭流,释放资源

pw.close();

}

}

# 关卡3

## 训练案例1

### 训练描述

有学生类包含学号，姓名，省份证号，Java成绩，数学成绩，英语成绩等成员变量，提供构造方法和setter和getter方法。

要求：

\* 学生信息及成绩保存到C盘的save.txt文件中

\* 学生身份证号码不能保存到文件中。

\* 程序运行时如果save.txt不存在，则

从键盘录入1个学生信息，信息录入格式如下：

\*\*\*\*\* 录入学生信息 \*\*\*\*\*

请输入学号：9527

请输入姓名：华安

请输入身份证号：2203919831234543

请输入Java成绩：90

请输入数学成绩：80

请输入英语成绩：88

根据录入的信息创建学生对象并将学生对象保存到C盘下的save.txt文件中。

\* 如果程序运行时，save.txt文件已经存在，则显示学生信息。格式如下：

\*\*\*\* 学生基本信息 \*\*\*\*\*

学号 姓名 省份证号 Java成绩 数学成绩 英语成绩

9527 华安 null 90 80 88

### 操作步骤答案

**public** **class** Test03\_01 {

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

// 根据文件路径创建文件对象

File file = **new** File("c:/save.txt");

**if**(file.exists()) { // 文件存在

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\* 学生基本信息 \*\*\*\*\*\*\*\*\*\*\*\*");

// 创建对象输入流对象

ObjectInputStream ois = **new** ObjectInputStream(**new** FileInputStream(file));

Student stu = (Student) ois.readObject();

System.***out***.println("学号\t姓名\t省份证号\tJava成绩\t数学成绩\t英语成绩");

System.***out***.println(stu.getId()+"\t"+stu.getName()+"\t"+ stu.getIdCard()+"\t " + stu.getJavaScore()+"\t "

+ stu.getMathScore()+"\t "+ stu.getEnglishScore()+"\t ");

// 关闭流

ois.close();

} **else** { // 文件不存在

// 创建文件

file.createNewFile();

// 创建键盘录入对象

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

// 录入学生信息

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\* 录入学生信息 \*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.print("请输入学号：");

String id = sc.nextLine();

System.***out***.print("请输入姓名：");

String name = sc.nextLine();

System.***out***.print("请输入身份证号：");

String idCard = sc.nextLine();

System.***out***.print("请输入Java成绩：");

String javaScore = sc.nextLine();

System.***out***.print("请输入数学成绩：");

String mathScore = sc.nextLine();

System.***out***.print("请输入英语成绩：");

String englishScore = sc.nextLine();

// 创建学生对象

Student stu = **new** Student(id, name, idCard, javaScore, mathScore, englishScore);

System.***out***.println(stu);

// 创建对象输出流

ObjectOutputStream oos = **new** ObjectOutputStream(**new** FileOutputStream(file));

oos.writeObject(stu);

// 关闭流

oos.close();

}

}

}

**class** Student **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** String id;

**private** String name;

**private** **transient** String idCard;

**private** String javaScore;

**private** String mathScore;

**private** String englishScore;

**public** Student(String id, String name, String idCard, String javaScore, String mathScore, String englishScore) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.idCard = idCard;

**this**.javaScore = javaScore;

**this**.mathScore = mathScore;

**this**.englishScore = englishScore;

}

**public** Student() {

**super**();

}

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getIdCard() {

**return** idCard;

}

**public** **void** setIdCard(String idCard) {

**this**.idCard = idCard;

}

**public** String getJavaScore() {

**return** javaScore;

}

**public** **void** setJavaScore(String javaScore) {

**this**.javaScore = javaScore;

}

**public** String getMathScore() {

**return** mathScore;

}

**public** **void** setMathScore(String mathScore) {

**this**.mathScore = mathScore;

}

**public** String getEnglishScore() {

**return** englishScore;

}

**public** **void** setEnglishScore(String englishScore) {

**this**.englishScore = englishScore;

}

@Override

**public** String toString() {

**return** "Student [id=" + id + ", name=" + name + ", idCard=" + idCard + ", javaScore=" + javaScore

+ ", mathScore=" + mathScore + ", englishScore=" + englishScore + "]";

}

}

## 训练案例2

### 训练描述

从键盘输入一个文件夹路径，利用打印流将该文件夹下的所有文件(包括子文件夹)复制到D盘下temp文件夹下。

### 操作步骤答案

**public** **class** Test03\_02 {

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

// 获得文件夹对象

File dir = *inputFile*();

// 创建目标文件夹对象

File destDir = **new** File("d:/temp");

*copyDir*(dir,destDir);

}

/\*

\* 利用打印流将该文件夹下的所有文件(包括子文件夹)复制到D盘下temp文件夹下。

\*/

**public** **static** **void** copyDir(File srcDir,File destDir) **throws** IOException{

// 创建文件夹

destDir.mkdirs();

// 获得源文件夹下所有的文件和子文件夹

File[] files = srcDir.listFiles();

// 遍历文件数组

**for** (File file : files) {

File destFile = **new** File(destDir,file.getName());

// 判断file是否是文件夹

**if**(file.isDirectory()) {

// 递归调用当前方法

*copyDir*(file, destFile);

} **else** { // 如果是文件,拷贝到目标文件夹destDir下

*copyFile*(file, destFile);

}

}

}

/\*

\* 字节流缓冲区流读写字节数组

\*/

**private** **static** **void** copyFile(File srcFile, File destFile) **throws** IOException {

// 创建字节缓冲输入流

BufferedInputStream bis = **new** BufferedInputStream(**new** FileInputStream(srcFile));

// 创建打印流对象

PrintStream ps = **new** PrintStream(destFile);

**int** len = 0;

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

// 循环读取数据

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

// 利用输出流往输出目的写数据

ps.write(buffer,0,len);

}

// 关闭资源

bis.close();

ps.close();

}

/\*

\* 根据用户输入的路径字符串创建文件对象

\*/

**private** **static** File inputFile() {

// 创建键盘输入对象

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

System.***out***.println("请输入一个文件夹路径：");

// 接收用户输入的路径

String filePath = sc.nextLine();

// 根据路径字符串创建文件对象

File dir = **new** File(filePath);

**if**(!dir.exists()) **throw** **new** RuntimeException("你输入的文件路径不存在！！！");

// 判断是否是文件夹

**if**(!dir.isDirectory()){

**throw** **new** RuntimeException("输入的路径不是文件夹路径！！！");

}

**return** dir;

}

}

## 训练案例3

### 训练描述

C盘下有两个文本文件，分别为a.txt和b.txt，其中a.txt编码方式是gbk,而b.txt的编码方式是utf8。要求将使用转换流实现如下功能：

\* 将a.txt和b.txt文件的内容复制到c.txt文件中，保证内容不乱码。

### 操作步骤答案

**public** **class** Test03\_03 {

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

// 将a.txt文件的内容复制到c.txt文件中，使用gbk方法读取a.txt文件

*copyFile*("c:/a.txt","c:/c.txt","gbk");

// 将b.txt文件的内容复制到c.txt文件中，使用utf8方法读取b.txt文件

*copyFile*("c:/b.txt","c:/c.txt","utf8");

}

**private** **static** **void** copyFile(String src,String dest,String encoding) **throws** FileNotFoundException, IOException {

// 创建转换输入流对象：使用默认的编码方式gbk读取a.txt文件

InputStreamReader isr = **new** InputStreamReader(**new** FileInputStream(src),encoding);

// 创建转换输出流对象:使用默认的编码方式：gbk

OutputStreamWriter osw = **new** OutputStreamWriter(**new** FileOutputStream(dest,**true**));

**char**[] cbuf = **new** **char**[1024];

**int** len = -1;

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

osw.write(cbuf, 0, len);

}

osw.write("\n");

// 关闭流释放资源

isr.close();

osw.close();

}

}