**每日作业卷**

**JavaSE第12天**

# 关卡1

## 训练案例1

### 训练知识点

1.字符流输出流一次写一个字符到文件中

### 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。

### 操作步骤描述

1. 创建键盘录入对象
2. 接收用户输入的一行字符串
3. 创建字符输出流对象并关联目标文件
4. 遍历字符串获得每一个字符
5. 调用输出流对象方法将字符写出到文件中。
6. 关闭流释放资源。

**public** **class** Test {

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

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

String str = scanner.nextLine();

**char**[] chs = str.toCharArray();

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

**for** (**int** i = 0; i < chs.length; i++) {

fileWriter.write(chs[i]);

}

fileWriter.close();

}

}

## 训练案例2

### 训练知识点

1. 字符流输出流一次写一个字符数组到文件中

### 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。

### 操作步骤描述

1. 创建键盘录入对象
2. 接收用户输入的一行字符串
3. 创建字符输出流对象并关联目标文件
4. 将字符串转换成字符数组
5. 调用输出流对象方法一次性将字符数组中的内容写出到文件中。

**public** **class** Test {

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

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

String str = scanner.nextLine();

**char**[] chs = str.toCharArray();

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

fileWriter.write(chs);

fileWriter.close();

}

}

## 训练案例3

### 训练知识点

1. 字符输入流一次读取一个字符
2. 字符输入流一次读取一个字符数组

### 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。

### 操作步骤描述

1. 创建字符输入流对象并关联目标文件
2. 使用循环读取文件中的内容，直到读到文件末尾为止。
3. 关闭流释放资源

**public** **class** Test {

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

FileReader fileReader = **new** FileReader(**new** File("c:/c.txt"));

**int** c = -1;

**while**((c=fileReader.read())!=-1){

System.***out***.println((**char**)c);

}

fileReader.close();

System.***out***.println("-----------");

FileReader fr2 = **new** FileReader("c:/c.txt");

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

**int** len=-1;

**while**((len=fr2.read(arr))!=-1) {

String s = **new** String(arr,0,len);

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

}

fr2.close();

}

}

## 训练案例4

### 训练知识点

1.转换输出流的使用

### 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。

### 操作步骤描述

1. 创建文件字节输出流关联目标文件
2. 根据文件字节输出流创建转换输出流对象，并指定编码字符集为：uft8
3. 调用流对象的方法将字符串写出到文件中。
4. 关闭流并释放资源。

**public** **class** Test {

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

FileOutputStream fileOutputStream = **new** FileOutputStream(**new** File("c:/a.txt"));

OutputStreamWriter outputStreamWriter = **new** OutputStreamWriter(fileOutputStream, "utf-8");

outputStreamWriter.write("我爱java");

outputStreamWriter.close();

}

}

## 训练案例5

### 训练知识点

1. 转换输入流的使用

### 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。

### 操作步骤描述

1. 创建字节输入流对象指定文件路径。
2. 根据字节输入流对象创建转换输入流对象并指定字符集编码为：utf8
3. 调用转换输入流对象的读取方法读取内容
4. 关闭流释放资源

**public** **class** Test {

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

FileInputStream fileInputStream = **new** FileInputStream(**new** File("c:/a.txt"));

InputStreamReader inputStreamReader = **new** InputStreamReader(fileInputStream, "utf-8");

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

**int** len = -1;

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

System.***out***.println(**new** String(chs,0,len));

}

inputStreamReader.close();

}

}

## 训练案例6

### 训练知识点

1. 对象的序列化

### 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。

### 操作步骤描述

1. 定义学生类并实现序列化接口和测试类
2. 在测试类中创建学生对象
3. 创建文件字节输出流对象并关联目标文件
4. 根据文件字节输出流对象创建对象输出流对象
5. 调用对象输出流对象的方法将学生对象保存到文件中
6. 关闭流释放资源

**public** **class** Test {

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

Student student = **new** Student("aa", 10, "男");

FileOutputStream fileOutputStream = **new** FileOutputStream("stu.txt");

ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(student);

objectOutputStream.close();

}

}

## 训练案例7

### 训练知识点

1.对象的反序列化

### 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。

### 操作步骤描述

1. 创建文件字节输入流对象关联目标文件
2. 根据文件字节输入流对象创建对象输入流对象
3. 调用对象输入流对象的方法从文件中获取学生对象
4. 关闭流释放资源。

**public** **class** Test {

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

FileInputStream fileInputStream = **new** FileInputStream(**new** File("stu.txt"));

ObjectInputStream objectInputStream = **new** ObjectInputStream(fileInputStream);

Student student = (Student)objectInputStream.readObject();

System.***out***.println(student.getName());

System.***out***.println(student.getAge());

System.***out***.println(student.getSex());

objectInputStream.close();

}

}

## 训练案例8

### 训练知识点

1.字节打印流的使用

### 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

### 操作步骤描述

1. 创建字节打印流对象并关联文件路径
2. 调用字节打印流对象的打印方法将内容输入到目标文件中
3. 关闭流释放资源

**public** **class** Test {

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

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

String str = scanner.nextLine();

PrintStream printStream = **new** PrintStream(**new** File("c:/d.txt"));

printStream.println(str);

printStream.close();

}

}

## 训练案例9

### 训练知识点

1. 字符打印流的基本使用

### 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。

### 操作步骤描述

1. 创建字符打印流对象关联文件路径
2. 调用字符打印流对象的方法将字符串保存到文件中
3. 关闭流并释放资源。

**public** **class** Test {

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

String str = "我爱java";

PrintStream printStream = **new** PrintStream(**new** File("c:/e.txt"));

printStream.println(str);

printStream.close();

}

}

## 训练案例10

### 10.1训练知识点

1.打印流的使用

### 10.2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。

### 10.3操作步骤描述

1. 创建打印流对象关联目标文件
2. 遍历集合
3. 调用打印流对象的方法将字符串保存到文件中
4. 关闭流释放资源

**public** **class** Test {

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

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

list.add("aaaa");

list.add("bbbb");

list.add("ccc");

PrintStream printStream = **new** PrintStream("array.txt");

**for**(String s:list){

printStream.println(s);

}

printStream.close();

}

}

# 关卡2

## 训练案例1

### 训练知识点

1. 转换流的使用

### 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

### 操作步骤描述

1. 创建转换输入流对象关联test.txt文件，并指定编码为gbk
2. 创建转换输出流对象关联test2.txt文件，并指定编码为gbk
3. 利用转换输入流循环读取test.txt文件内容，每读取一次就利用转换输出流对象将读取的内容保存到目标文件test2.txt文件中。
4. 关闭流释放资源

**public** **class** Test {

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

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

InputStreamReader inputStreamReader = **new** InputStreamReader(fileInputStream,"UTF-8");

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

OutputStreamWriter outputStreamWriter = **new** OutputStreamWriter(fileOutputStream,"UTF-8");

**int** len = -1;

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

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

outputStreamWriter.write(**new** String(chs,0,len));

}

inputStreamReader.close();

outputStreamWriter.close();

}

}

## 训练案例2

### 训练知识点

1. 对象的序列化
2. 对象输出流的使用

### 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。

### 操作步骤描述

1. 创建集合对象用来存放学生
2. 创建多个学生对象添加到集合中
3. 创建对象输出流关联目标文件
4. 调用对象输出流的方法将集合对象保存到文件中
5. 关闭流并释放资源

**public** **class** Test {

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

Student s1 = **new** Student("aa", 10, "男");

Student s2 = **new** Student("bb", 10, "男");

Student s3 = **new** Student("cc", 10, "男");

Student s4 = **new** Student("dd", 10, "男");

ArrayList<Student> students = **new** ArrayList<>();

students.add(s1);

students.add(s2);

students.add(s3);

students.add(s4);

FileOutputStream fileOutputStream = **new** FileOutputStream("stus.txt");

ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(students);

objectOutputStream.close();

}

}

## 训练案例3

### 训练知识点

1. 对象的反序列
2. 对象输入流的使用

### 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

### 操作步骤描述

1. 创建对象输入流并关联目标文件stus.txt
2. 调用read方法从文件中读取集合对象
3. 遍历集合，将集合中学生对象输出在控制台
4. 关闭流并释放资源

**public** **class** Test2 {

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

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

ObjectInputStream objectInputStream = **new** ObjectInputStream(fileInputStream);

ArrayList<Student> arrayList = (ArrayList<Student>) objectInputStream.readObject();

**for**(Student s:arrayList){

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

}

}

}

## 训练案例4

### 训练知识点

1. 字符打印流的使用

### 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

### 操作步骤描述

1. 创建字符输入流对象并关联c:/ccc.txt文件
2. 创建字符打印流对象并关联d:/ccc.txt文件
3. 通过输入流对象循环从文件中读取内容，每读取到内容就利用打印流对象将内容输出到目标文件中
4. 关闭流释放资源

**public** **class** Test {

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

FileReader fileReader = **new** FileReader(**new** File("c:/ccc.txt"));

PrintStream printStream = **new** PrintStream(**new** File("d:/ccc.txt"));

**int** len = -1;

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

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

printStream.println(**new** String(chs,0,len));

}

fileReader.close();

printStream.close();

}

}

## 训练案例5

### 训练知识点

1. 字节打印流的使用

### 训练描述

将C盘下的a.png图片复制到D盘下

### 操作步骤描述

1. 创建字节输入流对象并关联c:/a.png文件
2. 创建字节打印流对象并关联d:/a.png文件
3. 通过输入流对象循环从文件中读取内容，每读取到内容就利用打印流对象将内容输出到目标文件中
4. 关闭流释放资源

**public** **class** Test {

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

FileInputStream fileInputStream = **new** FileInputStream(**new** File("c:/a.png"));

PrintStream printStream = **new** PrintStream(**new** File("d:/a.png"));

**int** c=-1;

**while**((c=fileInputStream.read())!=-1) {

printStream.print((**char**)c);

}

fileInputStream.close();

printStream.close();

}

}

## 训练案例6

### 训练知识点

1.打印流的使用

### 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)

### 操作步骤描述

1. 定义变量记录平均分，最大值和最小值
2. 遍历数组，求和，最大值和最小值
3. 求平均分
4. 创建打印流对象并关联文件
5. 遍历数组，将大于平均值的元素输出到文件中
6. 关闭流释放资源

**public** **class** Test {

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

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

**int** sum = 0;

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

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

**for** (**int** i = 0; i < arr.length; i++) {

sum+=arr[i];

**if**(max<arr[i]){

max= arr[i];

}

**if**(min>arr[i]){

min =arr[i];

}

}

**int** average = sum/(arr.length);

System.***out***.println("平均分:"+average);

System.***out***.println("最大值:"+max);

System.***out***.println("最小值:"+min);

PrintStream printStream = **new** PrintStream("number.txt","gbk");

**for** (**int** i = 0; i < arr.length; i++) {

**if**(arr[i]>average){

printStream.append("'"+arr[i]+"'");

}

}

printStream.close();

}

}

# 关卡3

## 训练案例1

### 训练知识点

1.对象的序列化和反序列化

### 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

**public** **class** Test {

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

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

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

**if**(!file.exists()){

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

System.***out***.println("请输入学号：");

String id = scanner.next();

System.***out***.println("请输入姓名：");

String name = scanner.next();

System.***out***.println("请输入身份证号：");

String idenfyId = scanner.next();

System.***out***.println("请输入Java成绩：");

**int** javaScore = scanner.nextInt();

System.***out***.println("请输入数学成绩：");

**int** mathScore = scanner.nextInt();

System.***out***.println("请输入英语成绩：");

**int** engScore = scanner.nextInt();

Student s = **new** Student(id,name,**null**,javaScore,mathScore,engScore);

FileOutputStream fileOutputStream = **new** FileOutputStream(file);

ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(s);

objectOutputStream.close();

}**else** {

FileInputStream fileInputStream = **new** FileInputStream(file);

ObjectInputStream objectInputStream = **new** ObjectInputStream(fileInputStream);

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

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

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

System.***out***.println(s.getId()+" "+s.getName()+" "+s.getIdentfyId()

+" "+s.getJavaScore()+" "+s.getMathScore()

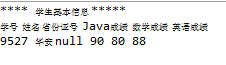
+" "+s.getEngScore());

objectInputStream.close();

}

}

}



## 训练案例2

### 训练知识点

1. 打印流的使用

### 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。

**public** **class** Test {

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

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

String name = scanner.nextLine();

File file = **new** File(name);

*copy*(file);

}

**public** **static** **void** copy(File file) **throws** Exception {

File[] files = file.listFiles();

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

**if**(f.isFile()){

FileInputStream fileInputStream = **new** FileInputStream(f);

PrintStream printStream = **new** PrintStream("d:/temp/"+f.getName());

**int** c = -1;

**while**((c=fileInputStream.read())!=-1){

printStream.write((**char**)c);

}

fileInputStream.close();

printStream.close();

}**else**{

File file2 = **new** File("d:/temp/"+f.getName());

file2.mkdir();

*copy*(f);

}

}

}

}

## 训练案例3

### 训练知识点

1. 转换流的使用

### 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。

**public** **class** Test {

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

FileInputStream fileInputStream = **new** FileInputStream(**new** File("c:/a.txt"));

InputStreamReader inputStreamReader = **new** InputStreamReader(fileInputStream,"GBK");

FileInputStream fileInputStream2 = **new** FileInputStream(**new** File("c:/b.txt"));

InputStreamReader inputStreamReader2 = **new** InputStreamReader(fileInputStream2,"utf-8");

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

OutputStreamWriter osw = **new** OutputStreamWriter(fos,"UTF-8");

**int** c = -1;

**while**((c=inputStreamReader.read())!=-1){

osw.write(c);

}

inputStreamReader.close();

**int** ch = -1;

**while**((ch=inputStreamReader2.read())!=-1){

osw.write(ch);

}

inputStreamReader2.close();

osw.close();

}

}