## 案例一：追加文件内容

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)。

提示：

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”

代码实现：

|  |
| --- |
| //异常应该使用try-catch-finally处理。  **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("d:/hello.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\n";  //2.2 遍历写出  **for** (**int** i = 0; i < 5; i++) {  fos.write(content.getBytes());  }  // 3.关闭流  fos.close();  } |

## 案例二：字节流复制文件

描述:利用字节流将D盘下的a.png图片复制到E盘下(文件名保存一致)

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **public** **static** **void** main(String[] args) **throws** IOException {  // 创建字节输入流对象并关联文件  File srcFile = **new** File("d:/a.png");  FileInputStream fis = **new** FileInputStream("d:/a.png");    // 创建字节输出流对象并关联文件  File destFile = **new** File("e:/",srcFile.getName());  FileOutputStream fos = **new** FileOutputStream(destFile);  // 定义byte[]接收读取的字节数  **byte**[] buffer = **new** **byte**[1024];  **int** len;  // 循环读取图片数据  **while** ((len = fis.read(buffer)) != -1) {  // 将每次读取的数据写出到目标文件中  fos.write(buffer,0,len);  }  // 关闭流  fis.close();  fos.close();  } |

## 案例三：保存控制台输入的数据

描述：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。

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **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();  } |

## 案例四：Properties与IO流综合使用

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"Tom"这样的键存在，如果有就改变其实为"100"。

现score.txt文件内容如下：

Jerry=90

Tom=80

Jack=85

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **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String类型的key  //也可以考虑调用keySet()获取所有key构成的集合，只不过类型是Object  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 "Tom"  **if** ("Tom".equals(key)) {  // 把"Tom"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"), "songhk update the properties");  } |

## 案例五：统计指定字符出现的次数

描述:

在D盘下有一个文本文件test.txt，

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

代码实现：

|  |
| --- |
| **public** **static** **void** main(String[] args) **throws** Exception {    **int** count = *calculateCharCount*("test.txt",'a');  System.***out***.println(count);  }  // 异常应该使用try-catch-finally处理。  **public** **static** **int** calculateCharCount(String file,**char** c) **throws** Exception {  **int** count = 0;  FileReader fr = **new** FileReader(file);  **int** cbuf;  **while**((cbuf = fr.read()) != -1) {  **if**(c == cbuf) {  count++;  }  }  fr.close();  **return** count;  } |

## 案例六：获取文本上字符出现的次数,把数据写入文件

思路：

1.遍历文本每一个字符

2.字符出现的次数存在Map中

Map<Character,Integer> map = new HashMap<Character,Integer>();

map.put('a',18);

map.put('你',2);

3.把map中的数据写入文件

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **public** **static** **void** main(String[] args) **throws** Exception {  // 1.创建Map集合  Map<Character, Integer> map = **new** HashMap<Character, Integer>();  // 2.遍历每一个字符,每一个字符出现的次数放到map中  FileReader fr = **new** FileReader("dbcp.txt");  **int** c = 0;  **while** ((c = fr.read()) != -1) {  // int 还原 char  **char** ch = (**char**) c;  // 判断char是否在map中第一次出现  **if** (map.get(ch) == **null**) {  map.put(ch, 1);  } **else** {  map.put(ch, map.get(ch) + 1);  }  }  // 3.把map中数据存在文件count.txt  // 3.1 创建Writer  BufferedWriter bw = **new** BufferedWriter(**new** FileWriter("wordcount.txt"));  // 3.2 遍历map,再写入数据  Set<Map.Entry<Character, Integer>> entrySet = map.entrySet();  **for** (Map.Entry<Character, Integer> entry : entrySet) {  **switch** (entry.getKey()) {  **case** ' ':  bw.write("空格=" + entry.getValue());  **break**;  **case** '\t':// \t表示tab 键字符  bw.write("tab键=" + entry.getValue());  **break**;  **case** '\r'://  bw.write("回车=" + entry.getValue());  **break**;  **case** '\n'://  bw.write("换行=" + entry.getValue());  **break**;  **default**:  bw.write(entry.getKey() + "=" + entry.getValue());  **break**;  }  bw.newLine();  }  fr.close();  bw.close();  } |

## 案例七：保存对象到文件中

描述：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students.txt中。每名学员信息存储一行，多个属性值中间用逗号隔开。

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **public** **static** **void** main(String[] args) **throws** Exception {  // 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：");  **int** id = sc.nextInt();  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 gender = sc.next();  System.***out***.print("请输入第" + i + "名学生的年龄：");  **int** age = sc.nextInt();  // 把信息封装到Student对象中  Student s = **new** Student(id, name, gender, 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("students.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.getGender() + "," + 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  } |

## 案例八：保存验证码，并验证

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **public** **static** **void** main(String[] args) **throws** Exception {  // 键盘录入3个字符串并写入项目根路径下的data.txt文件中  *writeStringToFile*();  // 验证码验证  *verifyCode*();  }  /\*\*  \* 键盘录入3个字符串并写入项目根路径下的data.txt文件中  \*  \* **@throws** Exception  \*/  **private** **static** **void** writeStringToFile() **throws** Exception {  // 创建高效字符缓冲输出流对象并和data.txt文件关联  BufferedWriter bw = **new** BufferedWriter(**new** FileWriter(**new** File("data.txt")));  String line = **null**;  // 创建键盘录入对象  Scanner sc = **new** Scanner(System.***in***);  **for** (**int** i = 0; i < 3; i++) {  System.***out***.println("请输入第" + (i + 1) + "个字符串验证码");  // 读取用户键盘录入的一行验证码信息  line = sc.nextLine();  // 将读取到的一行验证码写入到文件中  bw.write(line);  // 写入换行符  bw.newLine();  }  // 关闭流对象  bw.close();  }  /\*\*  \* 验证码验证  \*  \* **@throws** Exception  \*/  **private** **static** **void** verifyCode() **throws** Exception {  // 创建ArrayList集合，用于存储文件中的3个验证码  ArrayList<String> list = **new** ArrayList<>();  // 创建高效字符缓冲输入流对象,并和data.txt文件关联  BufferedReader br = **new** BufferedReader(**new** FileReader(**new** File("data.txt")));  String line = **null**;  // 循环读取每一行  **while** (**null** != (line = br.readLine())) {  // 将读到的每一行信息存入到list集合中  list.add(line);  }  // 关闭流对象  br.close();  // 创建键盘录入对象  Scanner sc = **new** Scanner(System.***in***);  // 提示用户输入验证码  System.***out***.println("请输入一个验证码");  String code = sc.nextLine();  **if** (list.contains(code)) {  System.***out***.println("验证成功");  } **else** {  System.***out***.println("验证失败");  }  } |

## 案例九：键盘读入内容，打印流输出

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

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **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("print.txt");  // 将字符串写出到文件中  ps.println(line);  // 关流  ps.close();  } |

## 案例十：文本内容反转

描述:

项目根路径下有text.txt文件，内容如下：

青春正好

恋爱趁早

利用IO流的知识读取text.txt文件的内容反转后写入text1.txt文件中，内容如下：

恋爱趁早

青春正好

代码实现：

|  |
| --- |
| // 异常应该使用try-catch-finally处理。  **public** **static** **void** main(String[] args) **throws** IOException {  // 创建集合  ArrayList<String> list = **new** ArrayList<>();  // 读取关联文件  BufferedReader br = **new** BufferedReader(**new** FileReader("text.txt"));  String line;  **while** ((line = br.readLine()) != **null**) {  // 每读取一行放入集合  list.add(line);  }  // 关闭流  br.close();  // 反转集合  Collections.*reverse*(list);  // 写入关联文件  BufferedWriter bw = **new** BufferedWriter(**new** FileWriter("text1.txt"));  **for** (**int** i = 0; i < list.size(); i++) {  // 逐行写入  bw.write(list.get(i));  bw.newLine();// 换行  bw.flush();// 刷新  }  bw.close();// 关闭流  } |