### Java第一阶段—DAY17-JAVA案例

1. 制造并抛出一个异常。

public class Demo02Exception {  
  
 public static void main(String[] args) {  
 *//创建数组* int[] arr = {1,2,3,4};  
 *//根据索引找对应的元素* int index = 4;  
 int element = getElement(arr, index);  
  
 System.out.println(element);  
 }  
  
 public static int getElement(int[] arr,int index){  
*// 当数组下表越界时抛出下标越界异常* if(index<0 || index>arr.length-1){  
 throw new ArrayIndexOutOfBoundsException("blb提醒您，数组下标已越界。");  
 }  
 return arr[index];  
 }  
}

1. 制造一个异常，捕获异常，并打印出现的异常信息。

public class Demo03Exception {  
  
 public static void main(String[] args) {  
 try {  
*// 制造一个异常。* System.out.println(5/0);  
 String s = null;  
 System.out.println(s.toString());  
 }catch (ArithmeticException e){  
 System.out.println("发生了算术运算的异常。");  
 e.printStackTrace();  
 }catch (NullPointerException e){  
 System.out.println("发生了空指针的异常。");  
 e.printStackTrace();  
 }  
 }  
}

1. 写程序读取D盘中a.txt文件的内容。

public class Demo08IO {  
  
 public static void main(String[] args) {  
 FileReader fileReader = null;  
 try {  
*// 读取D盘中a.txt文件的内容* File f = new File("D:\\a.txt");  
*// 创建对应的输入流* fileReader = new FileReader(f);  
 int c ;  
 while((c = fileReader.read())!=-1){  
 System.out.println( (char)c);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 try {  
*// 关闭流* if (fileReader != null) {  
 fileReader.close();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 }  
}

1. 将文本“昨夜雨疏风骤，浓睡不消残酒。试问卷帘人，却道海棠依旧。知否，知否？应是绿肥红瘦。"通过程序写到D盘下的blb.txt中。

public class Demo09IO {  
  
 public static void main(String[] args) {  
 File f = new File("D:\\blb.txt");  
 FileWriter fileWriter = null;  
 try {  
*// 创建文件输出流* fileWriter = new FileWriter(f,true);*// 第2个参数表示是追加还是覆盖* fileWriter.write("昨夜雨疏风骤，浓睡不消残酒。试问卷帘人，却道海棠依旧。知否，知否？应是绿肥红瘦。");  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }finally {  
 try {  
 if(fileWriter!=null){  
 fileWriter.close();  
 }  
 } catch (Exception e2) {  
 e2.printStackTrace();  
 }  
 }  
 }  
}

1. 读取D盘中a.txt文件的内容。

public class Demo10IO {  
  
 public static void main(String[] args) throws IOException {  
 FileReader fileReader = null;  
 BufferedReader bufferdReader = null;  
 bufferdReader = new BufferedReader(fileReader);  
*// 读取D盘中a.txt文件的内容* File f = new File("D:\\blb.txt");  
*// 创建对应的输入流* fileReader = new FileReader(f);  
 String s = null;  
  
 while((s = bufferdReader.readLine())!=null){  
 System.out.println(s);  
 }  
*// 关闭流* bufferdReader.close();  
 fileReader.close();  
  
 }  
}

1. 将“床前明月光，疑是地上霜。举头望明月，低头思故乡。”文本写入到d盘下的blb.txt文件中。

public class Demo11IO {  
  
 public static void main(String[] args) throws IOException {  
 File f = new File("D:\\blb.txt");  
 FileWriter fileWriter = null;  
 BufferedWriter bufferedWriter = null;  
 fileWriter = new FileWriter(f,true);*// 第2个参数表示是追加还是覆盖* bufferedWriter = new BufferedWriter(fileWriter);  
  
 bufferedWriter.newLine();  
 bufferedWriter.write("床前明月光");  
 bufferedWriter.newLine();  
 bufferedWriter.write("疑是地上霜");  
 bufferedWriter.newLine();  
 bufferedWriter.write("举头望明月");  
 bufferedWriter.newLine();  
 bufferedWriter.write("低头思故乡");  
  
 bufferedWriter.flush();  
  
 bufferedWriter.close();  
 fileWriter.close();  
 }  
}

1. 自定义Person类，创建其对象，通过对象流写入跟读取。

**class** Person **implements** Serializable {  
*// 序列化版本号，必须保证序列化跟反序列化的版本号一致* **private static final long** serialVersionUID = -1213754818242979380L;  
  
 **public** String name ;  
 *//设置age字段为transient，表示此字段不需要序列化* **public transient int** age = 1 ;  
  
 **public** String sex ;  
  
 @Override  
 **public** String toString() {  
 **return "Person{"** +  
 **"name='"** + name + **'\''** +  
 **", age="** + age +  
 **", sex='"** + sex + **'\''** +  
 **'}'**;  
 }  
}  
  
  
**public class** Demo12Serializable {  
  
 **public static void** main(String[] args) **throws** IOException, ClassNotFoundException {  
*// saveObject();// 序列化* readObject();*// 反序列化* }  
  
*// 序列化* **public static void** saveObject() **throws** IOException{  
*// 准备要序列化的对象* Person person = **new** Person();  
 person.name = **"张三"**;  
 person.age = 18 ;  
 person.sex = **"男"**;  
*// 准备对象流* FileOutputStream fos = **new** FileOutputStream(**"d:/blb.txt"**);  
 ObjectOutputStream out = **new** ObjectOutputStream(fos);  
*// 序列化对象* out.writeObject(person);  
*// 关闭流* out.close();  
 fos.close();  
 }  
  
*// 反序列化* **public static void** readObject() **throws** IOException, ClassNotFoundException {  
 *// 准备对象流* FileInputStream fis = **new** FileInputStream(**"d:/blb.txt"**);  
 ObjectInputStream in = **new** ObjectInputStream(fis);  
 *// 反序列化对象* Person person = (Person) in.readObject();  
 System.out.println(person);  
 *// 关闭流* in.close();  
 fis.close();  
 }  
  
}

1. 创建文件blb.properties，将name、age、sex信息通过程序写入文件中。

public class Demo13IO {  
  
 public static void main(String[] args) throws IOException {  
*// 创建对应的属性文件* File file = new File("d:/blb.properties");  
 if(!file.exists()){  
 file.createNewFile();  
 }  
*// 创建Properties对象，存储数据* Properties p = new Properties();  
 p.setProperty("name", "张三");  
 p.setProperty("age", "22");  
 p.setProperty("sex", "男");  
  
*// 通过OutputStreamWriter包装一层流，可以处理乱码问题* FileOutputStream fos = new FileOutputStream(file);  
 OutputStreamWriter osw = new OutputStreamWriter(fos,"utf-8");  
  
*// 将Properties中的数据直接通过流写出去* p.store(osw,"");  
*// 关闭流* osw.close();  
 fos.close();  
 }  
}

1. 将blb.properties文件中的信息通过程序读取并显示。

public class Demo14Properties {  
  
 public static void main(String[] args) throws IOException {  
*// 创建对应的属性文件* File file = new File("d:/blb.properties");  
  
*// 建立到文件之间的流* FileInputStream fis = new FileInputStream(file);  
 InputStreamReader read = new InputStreamReader(fis,"utf-8");  
  
*// 通过流把配置文件中的数据加载到Properties对象中。* Properties properties = new Properties();  
 properties.load(read);  
*// 关闭流* read.close();  
 fis.close();  
  
 System.out.println(properties.getProperty("name"));  
 System.out.println(properties.getProperty("age"));  
 System.out.println(properties.getProperty("sex"));  
  
*// 遍历所有内容* Set<String> keys = properties.stringPropertyNames();  
 for (String key:keys) {  
 System.out.println( key+ " = "+ properties.getProperty(key));  
 }  
  
 }  
}