|  |
| --- |
| 专业班级 信息173班 学号 201711010228 姓名 苏栋 |

**实验代码：**

package book;

public class Book

{

//定义Book的四个属性， id name author price

int id;

String name;

String author;

float price;

//定义Book类的全参构造方法

public Book(int id,String name, String author,float price)

{

this.id = id;

this.name = name;

this.author = author;

this.price = price;

}

}

package book;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

public class Ex6\_17

{

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

{

//创建Book类的对象，并为其赋初值

Book book=new Book(201710, "Java", "S\_Tong", 25);

//构建对象输出流

ObjectOutputStream ooStream=new ObjectOutputStream(new FileOutputStream("E:/book.dat"));

//写入指定的对象book

ooStream.writeObject(book);

//关闭输出流

ooStream.close();

//将book对象的值置空

book=null;

//构建对象输入流

ObjectInputStream oisInputStream=new ObjectInputStream(new FileInputStream("E:/book.dat"));

//读取并返回一个对象book

book=(Book)oisInputStream.readObject();

//关闭输入流管道

oisInputStream.close();

//原样输出ID is:及book属性id的值

System.out.println("ID is:"+book.id);

//原样输出name is:及book属性name的值

System.out.println("name is:"+book.name);

//原样输出author is:及book属性author的值

System.out.println("author is:"+book.author);

//原样输出price is:及book属性price的值

System.out.println("price is:"+book.price);

}

}

package demo;

import java.io.\*;

public class Blip3

{

//定义类Blip3的属性i 和 s

int i;

String s;

public Blip3(String string, int j)

{

// TODO Auto-generated constructor stub

}

//构造类Blip3的无参构造方法

public Blip3()

{

//原样输出Blip3 Constructor

System.out.println("Blip3 Constructor");

}

//构造类Blip3的全参构造方法

public void Blip31(String x,int a)

{

//原样输出Blip3(String x, int a)

System.out.println("Blip3(String x,int a)");

//将x的值赋给属性s

s=x;

//将a的值赋给属性i

i=a;

}

//定义toString()方法

public String toString()

{

return "Blip3 [i=" + i + ", s=" + s + "]";

}

//实现接口Externalizable中的方法writeExternal

public void writeExternal(ObjectOutput out) throws IOException

{

System.out.println("Blip3.writeExternal");

out.writeObject(s);

out.writeInt(i);

}

//实现接口Externalizable中的方法readExternal

public void readExternal(ObjectInput in) throws ClassNotFoundException, IOException

{

System.out.println("Blip3.readExternal");

s=(String)in.readObject();

i=in.readInt();

}

@SuppressWarnings("resource")

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

{

//原样输出Constructing objects:

System.out.println("Constructing objects:");

//创建类Blip3的对象b3

Blip3 b3=new Blip3("A String",47);

//输出对象b3，即调用其toString()方法

System.out.println(b3);

//构建对象输出流

ObjectOutputStream objectOutputStream=new ObjectOutputStream(new FileOutputStream("Blip3.out"));

//原样输出Saving object:

System.out.println("Saving object:");

//写入指定的对象b3

objectOutputStream.writeObject(b3);

//关闭输出流管道

objectOutputStream.close();

//构建对象输入流

ObjectInputStream inputStream=new ObjectInputStream(new FileInputStream("Blip3.out"));

//原样输出Recovering b3:

System.out.println("Recovering b3:");

//读取并返回一个对象b3

b3=(Blip3)inputStream.readObject();

//输出对象b3，即调用toString()方法

System.out.println(b3);

}

}

package demo;

import java.util.zip.\*;

import java.io.\*;

public class Ex6\_15

{

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

{

//构建压缩流，生成压缩文件test.zip

ZipOutputStream out=new ZipOutputStream(new BufferedOutputStream(new FileOutputStream("E:/test.zip")));

//遍历main函数的String数组参数

for(int i=0;i<args.length;i++)

{

//原样输出Writing file以及对应的参数

System.out.println("Writing file"+args[i]);

//构建字节输入流的缓冲流

BufferedInputStream inputStream=new BufferedInputStream(new FileInputStream(args[i]));

out.putNextEntry(new ZipEntry(args[i]));

//定义int类型变量c

int c;

//read()方法读取一个字节，转化为[0,255]的之间的一个整数，返回一个int，如果读到了文件末尾，则返回-1

while((c=inputStream.read())!=-1)

{

//write()方法写一个字节的低8位

out.write(c);

}

//关闭输入流管道

inputStream.close();

}

//关闭输出流管道

out.close();

//原样输出Reading file

System.out.println("Reading file");

//构建压缩流，将生成的test.zip压缩文件解压恢复原状

ZipInputStream in2=new ZipInputStream(new BufferedInputStream(new FileInputStream("E:/test.zip")));

//定义ZipEntry类型变量zEntry

ZipEntry zEntry;

//判断：当每次获取到的ZIP文件是否为null

while((zEntry=in2.getNextEntry())!=null)

{

System.out.println("Reading file"+zEntry.getName());

int x;

while((x=in2.read())!=-1)

{

System.out.println(x);

}

}

//关闭输入流管道

in2.close();

}

}

package employee;

public class Employee

{

//定义Employee类的两个属性 name 和 age 其中姓名不超过8个字符

char name[]= {'\u0000','\u0000','\u0000','\u0000','\u0000','\u0000','\u0000','\u0000'};

int age;

//定义Employee类的全参构造方法

public Employee(String name,int age)

{

//判断将String类型的参数转为字符数组，数组的长度是否>8

if(name.toCharArray().length>8)

{

System.arraycopy(name.toCharArray(), 0, this.name, 0,8);

}

else

{

System.arraycopy(name.toCharArray(), 0, this.name, 0, name.toCharArray().length);

}

this.age=age;

}

}

package employee;

import java.io.\*;

public class Ex6\_18

{

//定义类的属性Filename

String Filename;

//定义类Ex6\_18的全参构造方法

public Ex6\_18(String Filename)

{

this.Filename=Filename;

}

//定义写对象的方法

public void writeEmployee(Employee e,int n) throws Exception

{

//创建RandomAccessFile类的对象 对文件执行的操作类型是读写

RandomAccessFile randomAccessFile=new RandomAccessFile(Filename, "rw");

//移动文件位置，从偏离文件开头的n\*20个字节数的位置开始

randomAccessFile.seek(n\*20);

for(int i=0;i<8;i++)

{

//循环8次写入4个字节的int型数字

randomAccessFile.writeInt(e.age);

//关闭随机访问流

randomAccessFile.close();

}

}

//定义读对象的方法

public void readEmployee(int n) throws Exception

{

//创建8个字符大小的字符数组

char buf[]=new char[8];

//创建RandomAccessFile类的对象 对文件执行的操作类型是只读

RandomAccessFile randomAccessFile=new RandomAccessFile(Filename, "r");

//移动文件位置，从偏离文件开头的n\*20个字节数的位置

randomAccessFile.seek(n\*20);

for(int i=0;i<8;i++)

{

//循环8次读取字符

buf[i]=randomAccessFile.readChar();

}

System.out.println("name");

System.out.println("buf");

System.out.println("age:"+randomAccessFile.readInt());

//关闭随机访问流

randomAccessFile.close();

}

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

{

//创建类Ex6\_18的对象tEx6\_18

Ex6\_18 tEx6\_18=new Ex6\_18("E:/1.txt");

//创建三个Employee类的对象

Employee e1=new Employee("张三",23);

Employee e2=new Employee("李四",44);

Employee e3=new Employee("赵六",36);

//写入对象数据

tEx6\_18.writeEmployee(e1, 0);

tEx6\_18.writeEmployee(e3, 2);

System.out.println("第一个雇员的信息");

//读取第一个雇员的信息

tEx6\_18.readEmployee(0);

System.out.println("第三个雇员的信息");

//读取第三个雇员的信息

tEx6\_18.readEmployee(2);

//写入对象数据

tEx6\_18.writeEmployee(e2, 1);

System.out.println("第二个雇员信息");

//读取第二个雇员的信息

tEx6\_18.readEmployee(1);

}

}

package Student;

public class SortedIntArray

{

//定义类SortedIntArray的三个私有属性 capacity rep size

private int capacity;

private Integer[] rep;

private int size;

//定义类SortedIntArray的带参构造方法

public SortedIntArray(int n)

{

capacity=n;

rep=new Integer[capacity];

}

//定义类SortedIntArray的无参构造方法

public SortedIntArray()

{

this(100);

}

private int search(int i, int lower, int upper)

{

int index=lower;

if (upper>=lower)

{

int middle=(upper+lower)/2;

int current=rep[middle].intValue();

if (current==i)

{

index=middle;

}

else if(current<i)

{

index=search(i, middle+1, upper);

}

else

{

index=search(i, lower, middle-1);

}

}

return index;

}

//定义公共的查询方法

public int search(int i)

{

return search(i, 0, size-1);

}

// 定义公共的插入方法

public SortedIntArray insert(int i)

{

int index=search(i);

for(int j=size;j>index;--j)

{

rep[j]=rep[j-1];

}

rep[index]=new Integer(i);

++size;

return this;

}

//定义公共的移除方法

public SortedIntArray remove(int i)

{

int index = search(i);

if (rep[index].intValue()==i)

{

--size;

for (int j=index;j<size;++j)

{

rep[j]=rep[j+1];

}

}

return this;

}

//定义toString()方法

public String toString()

{

String toReturn="";

for (int i=0;i<size;++i)

{

toReturn+=rep[i].toString()+ ",";

}

return toReturn;

}

public static void main(String[] args)

{

//创建类SortedIntArray的对象

SortedIntArray anArray = new SortedIntArray();

//插入元素

anArray.insert(3).insert(2).insert(5).insert(6).insert(4). insert(7).insert(1).insert(8);

//调用toString()方法

System.out.println(anArray);

//移除元素

anArray.remove(3).remove(5).remove(7).remove(1);

//调用toString()方法

System.out.println(anArray);

}

}

package Student;

import java.io.\*;

public class SortTester

{

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

{

//创建Student类的对象并申请空间

Student1 students[]=new Student1[5];

//构建文件输入流

FileInputStream fileOutputStream=new FileInputStream("stu.ser");

//构建对象输入流

ObjectInputStream stream=new ObjectInputStream(fileOutputStream);

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

{

//循环5次读取students[]对象

students[i]=(Student1)stream.readObject();

}

//关闭输入流管道

stream.close();

//创建类StudentClass的对象，并对其初始化

StudentClass1 aClass=new StudentClass1("软件0201", 5);

aClass.setStudents(students);

//调用toString()方法

System.out.println(aClass);

//调用选择排序方法

aClass.selectionSort();

System.out.println("选择排序后的结果:\n");

//调用toString()方法

System.out.println(aClass);

}

}

package Student;

public class Student1

{

//定义Student类的6个私有属性 id name eng math comp sum

private String id;

private String name;

private int eng;

private int math;

private int comp;

private int sum;

//定义Student类的全参构造方法

public Student1(String id,String name,int eng,int math,int comp)

{

this.id=id;

this.name=name;

this.eng=eng;

this.math=math;

this.comp=comp;

sum();

}

//定义Student类的构造方法

public Student1(Student1 s, int math, int comp)

{

this.id=s.id;

this.name=new String(s.name);

this.eng=s.eng;

this.math=math;

this.comp=comp;

sum();

}

//生成所有属性的get和set方法

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 int getEng()

{

return eng;

}

public void setEng(int eng)

{

this.eng = eng;

}

public int getMath()

{

return math;

}

public void setMath(int math)

{

this.math = math;

}

public int getComp()

{

return comp;

}

public void setComp(int comp)

{

this.comp = comp;

}

public int getSum()

{

return sum;

}

public void setSum(int sum)

{

this.sum = sum;

}

//定义sum方法

void sum()

{

this.sum=eng+math+comp;

}

//定义toString()方法

@Override

public String toString()

{

return "Student [id=" + id + ", name=" + name + ", eng=" + eng + ", math=" + math + ", comp=" + comp + ", sum="

+ sum + "]";

}

//判断参数对象是否为Student类的对象

public boolean equals(Object x)

{

//若当前对象不是Student类的对象

if(this.getClass()!=x.getClass())

//直接返回false

return false;

//将参数对象转为Student类型

Student1 bStudent=(Student1)x;

//返回当前对象的id是否和参数对象的id相等

return (this.getId().equals(bStudent.getId()));

}

//比较成绩大小，当前对象成绩比参数对象成绩大时返回1 相等返回0 其他返回-1

public int compare(Student1 A)

{

if(this.getSum()>A.getSum())

return 1;

else if(this.getSum()==A.getSum())

return 0;

else return -1;

}

}

package Student;

public class StudentClass1

{

//定义StudentClass类的4个属性 3个私有属性name Students[] size 静态属性capacity

private String name;

static int capacity=40;

private Student1 students[];

private int size;

//定义StudentClass类的构造方法

public StudentClass1(String name, int size)

{

super();

this.name = name;

this.size = size;

}

//定义所有属性的get和set方法

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public static int getCapacity()

{

return capacity;

}

public static void setCapacity(int capacity)

{

StudentClass1.capacity = capacity;

}

public Student1[] getStudents()

{

return students;

}

public void setStudents(Student1[] students)

{

for(int i=0;i<size;i++)

{

this.students[i] = new Student1(students[i], i, i);

}

}

public int getSize()

{

return size;

}

public void setSize(int size)

{

this.size = size;

}

//定义toString()方法

@Override

public String toString()

{

String s;

s= " 班级:" + name +"\t" + "容量:" + capacity + "\t" + "实际人数:" + size +"\n\n";

s= s + "学号"+"\t" + "姓名"+"\t" + "英语"+"\t"+"数学"+"\t"+ "计算机" +"\t" + "总成绩\n";

for (int i=0; i<size; i++)

s=s+students[i].getId()+"\t"+students[i].getName()+"\t"+students[i].getEng()+"\t"+students[i].getMath()+"\t"+students[i].getComp()+"\t"+students[i].getSum()+"\n";

return s;

}

//已知学生的学号，查找此学号是否存在 如果存在，返回其在数组中的下标位置 若不存在，返回-1

public int find(String id)

{

for (int i=0;i<size;i++)

if (students[i].getId().equals(id)) return i;

return -1;

}

//在数组的末尾添加一个学生对象 增加之前需先判断数组中是否还有空间，并且在数组中查找将要增加的学号是否已经存在

//增加成功，返回true 否则返回false

public boolean add(Student1 aStudent)

{

if (size==capacity) return false;

if (find(aStudent.getId())>=0)

return false;

this.students[size]=new Student1(newString(aStudent.getId()),new String(aStudent.getName()),

aStudent.getEng(), aStudent.getMath(),aStudent.getComp() );

size++;

return true;

}

//已知一个Student对象，将此对象从数组中删除

public boolean del(Student1 aStudent)

{

int pos=find(aStudent.getId());

if (pos==-1) return false;

for (int i=pos+1;i<size;i++)

students[i-1]=students[i];

size--;

return true;

}

//已知一个学号，删除一个学生

public boolean del(String id)

{

int pos=find(id);

if (pos==-1) return false;

for (int i=pos+1;i<size;i++)

students[i-1]=students[i];

size--;

return true;

}

//使用选择排序的方法将学生按总成绩从高到低排序

public void selectionSort()

{

Student1 temp;

for (int i=0;i<size-1;i++)

for (int j=i+1;j<size;j++)

if (students[j].compare(students[i])>0)

{

temp = students[i];

students[i]=students[j];

students[j]=temp;

}

}

//插入排序方法

public void insertSort()

{

Student1 temp;

for (int i=1;i<size;i++)

{

temp=students[i];

int j=i-1;

while (j>-1 && temp.compare(students[j])>0)

{

students[j+1]=students[j];

j--;

}

students[j+1]=temp;

}

}

}

package Student;

import java.io.\*;

import java.util.Scanner;

public class Tester1

{

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

{

//声明Student类型的数组对象

Student1 students[];

//创建StudentClass类的对象

StudentClass1 aClass=new StudentClass1("信息0228", 9);

//为声明的数组对象申请空间

students=new Student1[5];

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

//初始化数组对象

students[i]=new Student1(getAStudent(i+1), i, i);

}

//

aClass.setStudents(students);

//调用toString()方法

System.out.println(aClass);

//定义文件输出流

FileOutputStream fOutputStream=new FileOutputStream("stu.ser");

//定义对象输出流

ObjectOutputStream sObjectOutputStream=new ObjectOutputStream(fOutputStream);

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

//写入指定的数组对象

sObjectOutputStream.writeObject(students[i]);

//关闭输出流管道

sObjectOutputStream.close();

}

}

public static Student1 getAStudent(int i){

//声明Student类的对象

Student1 studenti;

System.out.println("输入第" + i + "个学生的信息:");

@SuppressWarnings("resource")

Scanner scanner=new Scanner(System.in);

System.out.print("学号:");

//从键盘获取id

String id =scanner.next();

System.out.print("姓名:");

//从键盘获取name

String name = scanner.next();

System.out.print("英语成绩:");

//从键盘获取eng

int eng = scanner.nextInt();

System.out.print("数学成绩:");

//从键盘获取math

int math = scanner.nextInt();

System.out.print("计算机成绩:");

//从键盘获取comp

int comp = scanner.nextInt(); ;

//初始化声明的对象studenti

studenti = new Student1(id,name,eng,math,comp);

return studenti;

}

}