

**计算机与信息 学院实验报告**

|  |  |  |
| --- | --- | --- |
| 实验课程： | 面向对象程序设计实验 | |
| 实验编号： | 实验七 | |
| 实验名称： | 抽象类与接口 | |
| 实验人员： | 学号 | 22111302041 |
| 姓名 | 高蝶 |
| 班级 | 22计算机科学与技术1班 |
| 指导教师： | 陶涛 | |
| 实验室： | 学院2号楼202 | |
| 实验日期： | 2023.4.4 | |

# 实验目的

1. 掌握抽象类的定义和使用；

2. 掌握接口的定义和使用；

3. 通过继承实现多态；

4. 通过接口实现多态。

# 实验要求

1. 所编写的程序应符合类的封装、继承、实现接口要求，符合面向对象程序编写规范；

2. 能利用继承性、多态性进行复杂程序的设计和实现；

3. 对问题进行充分的需求分析，设计正确、全面的测试数据，设计类的定义框架及UML图后，再进行编程实现；

4. 所设计的测试数据全部通过后，再进行程序提交，提高程序的通过率。

# 实验内容

**1.验证型实验**

仔细读下面的JAVA语言源程序，自己给出程序的运行结果，再与上机运行程序所得的结果作比较，如果有错，务必弄清楚出错原因。

import java.lang.\*;

interface Common {

double runTimer(doublea, double b, double c);

String getName(); //获取交通工具的名称

}

class Plane implementsCommon {

public doublerunTimer(double a, double b, double c) {

return (a+ b + c);

}

public String getName(){

return"Plane";

}

}

class Car implements Common {

public doublerunTimer(double a, double b, double c) {

return ( a\*b/c );

}

public String getName(){

return"Car";

}

}

public class ComputeTime {

public static void main(Stringargs[]) {

double A=3;

double B=5;

double C=6;

double v,t;

Common d=new Car();

v=d.runTimer(A,B,C);

t=1000/v;

System.out.println(d.getName()+"的平均速度: "+v+" km/h");

System.out.println(d.getName()+"的运行时间："+t+" 小时");

d=newPlane();

v=d.runTimer(10,30,40);

t=1000/v;

System.out.println(d.getName()+"的平均速度: "+v+" km/h");

System.out.println(d.getName()+"的运行时间："+t+" 小时");

}

}

**2.应用型实验**

**2.1面积计算**

设计要求：能实现抽象类的继承关系；程序应包括各个被调用方法的执行结果的显示。

（1）设计一个抽象的形状类 Shape，包含一个 getArea()方法，该方法不包含实际语句。

（2）在 Shape 类基础上设计圆形、矩形、三角形和梯形四个子类，要求根据实际形状重写 getArea()方法。

（3）设计一个 ShapeTest类，包含变量 area（存储总面积）、静态方法 countArea(Shape s)，该方法负责把参数中的形状面积加入到 area 中。在 main 函数中新建（2）中四种类型的对象s1、s2、s3、s4，通过调用 countArea 方法把四个对象面积累加到 area 中，最后输出 area。

输入：圆形的半径，矩形的长与宽，三角形的三条边长，梯形的上底、下底与高。注意：输入的值有的是非法的，则相应的面积值为0。

输出：面积和 ，保留4位小数

样例输入：

1

2 3

3 4 5

2 4 4

样例输出：

27.1416

**2.2 计算平均分**

歌唱比赛计算选手成绩的办法是去掉十个评委的一个最高分和最低分后再计算平均分，而学校考察一个班级的某科目的考试情况时，是计算全班同学的平均成绩。Gymnastics类和School类都实现了ComputerAverage接口，但是实现的方式不同。

ComputerAverage接口中有个方法getAverageScore(double []score);

输入：一个整数c，c为1时表示输入歌唱比赛选手的10个成绩；c为2表示输入一个班级的计算机课程考试成绩，输入班级人数n，n个学生的学号与成绩。已知所有成绩都是合法的。

输出：相应对象的平均分，保留3位小数。

样例输入1：

1

9 8 10 9.6 9.2 8.9 7.5 8.8 9.2 9.1

样例输出1：

8.975

样例输入2：

2

5

s001 95

s002 88

s003 92

s004 97

s005 90

样例输出2：

92.400

**2.3 乘法运算**

定义Mul接口，该接口中只有乘法运算方法multiply（int n）。定义一个新的字符串类MyString，该类的成员变量为String对象，MyString类实现了Mul接口，当对象调用multiply(n)方法时会返回一个新的字符串，其内容为原字符串的内容重复n次。

定义一个MyArray类，它可以完成创建指定元素个数的整型数组，当输出该类对象时，则会输出该对象的整型数组元素，MyArray类也实现了Mul接口，当对象调用multiply(n)方法时会返回一个新的MyArray对象，新对象的整型数组大小为当前对象数组大小的n倍，新对象数组中的内容是将当前对象数组内容重复n次。

输入：一个整数c，c为1表示创建MyString对象并执行相应操作，c为2表示创建MyArray对象并执行相应操作。当c为1时，输入字符串与整数n；当c为2，输入整数n，n个整数。

输出：对于MyString对象，输出对象值，再输出multiply方法的返回值；对于MyArray对象，输出对象值，再输出multiply方法的返回值。

样例输入1：

1

abc

3

样例输出1：

abc

abcabcabc

样例输入2：

2

4

-2 4 5 0

3

样例输出2：

4 -2,4,5,0

12 -2,4,5,0,-2,4,5,0,-2,4,5,0

**3. 设计型实验**

**3.1 贷款申请**

学校中有老师和学生两类人，而在职研究生既是老师又是学生，对学生的管理和对教师的管理在他们身上都有体现。

（1）设计两个信息管理接口StudentInterface和TeacherInterface。其中，StudentInterface接口包括setFee()方法和getFee()方法，分别用于设置和获取学生的学费；TeacherInterface口包括setPay()方法和getPay()方法，分别用于设置和获取教师的工资。

（2）定义一个研究生类Graduate，实现StudentInterface接口和TeacherInterface接口，它定义的成员变量有id（学号）、name(姓名)、sex(性别)、fee(每学期学费)、pay(月工资)，重写接口中的相应方法，再定义一个计算是否需要助学贷款的方法，统计对象的年收入和一年学费，如果收入减去学费不足2000元，则返回“provide a loan”(需要贷款)信息，否则返回空串。

输入：整数n，n个研究生信息

输出：需要助学贷款的研究生的学号、姓名及其需要贷款信息

样例输入：

3

x001 cc male 10000 1200

x003 bb female 12000 1200

x002 aa male 8000 2000

样例输出：

x001,cc,provide a loan

x003,bb,provide a loan

# 程序清单

2.应用型实验

（1）

测试数据

第1组：1

2 3

3 4 5

2 4 4

第2组：2

1 3

3 5 7

2 3 3

第3组：3

1 4

3 4 4

2 4 5

程序代码：**package** prj\_abstractextends;

**import** java.util.Scanner;

**public** **class** Main {

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

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

circle s1 = **new** circle(reader.nextDouble());

juxing s2 = **new** juxing(reader.nextDouble(),reader.nextDouble());

threeangle s3 = **new** threeangle(reader.nextDouble(),reader.nextDouble(),reader.nextDouble());

tixing s4 = **new** tixing(reader.nextDouble(),reader.nextDouble(),reader.nextDouble());

shapetest.*countArea*(s1);

shapetest.*countArea*(s2);

shapetest.*countArea*(s3);

shapetest.*countArea*(s4);

System.***out***.printf("%.4f",shapetest.*area*);

}

}

**abstract** **class** Shape {

**double** area;

**abstract** **void** getArea();

}

**class** circle **extends** Shape {

**double** r;

circle() {

}

circle(**double** r) {

**if**(r>0)

**this**.r = r;

}

**void** getArea() {

**this**.area = Math.***PI*** \* r \* r;

//System.out.println(area);

}

}

**class** juxing **extends** Shape {

**double** a, b;

juxing() {

}

juxing(**double** a, **double** b) {

**if**(a>0&&b>0) {

**this**.a = a;

**this**.b = b;

}

}

**void** getArea() {

area = a \* b;

//System.out.println(area);

}

}

**class** threeangle **extends** Shape {

**double** a,b,c,p;

threeangle() {

}

threeangle(**double** a, **double** b,**double** c) {

**if**(a+b>c||a+c>b||b+c>a) {

**this**.a = a;

**this**.b = b;

**this**.c=c;

p=(a+b+c)/2;

}

}

**void** getArea() {

area =Math.*sqrt*(p\*(p-a)\*(p-b)\*(p-c));

//System.out.println(area);

}

}

**class** tixing **extends** Shape {

**double** h, updi, downdi;

tixing() {

}

tixing( **double** updi, **double** downdi,**double** h) {

**if**(h>0&&updi>0&&downdi>0) {

**this**.h = h;

**this**.updi = updi;

**this**.downdi = downdi;

}

}

**void** getArea() {

area = (updi + downdi) \* h \* 0.5;

//System.out.println(area);

}

}

**class** shapetest {

**static** **double** *area*;

**static** **void** countArea(Shape s) {

s.getArea();

*area*+=s.area;

}

}

（2）

测试数据

第1组：1

9 8 10 9.6 9.2 8.9 7.5 8.8 9.2 9.1

第2组：2

5

s001 95

s002 88

s003 92

s004 97

s005 90

第3组：2

4

s001 95

s002 88

s003 92

s004 97

程序代码：**package** prj\_jisuanaverage;

**import** java.util.Scanner;

**public** **class** Main {

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

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

**int** x=reader.nextInt();

**if**(x==1) {

**double** sco[]=**new** **double**[15];

**for**(**int** i=0;i<10;i++) {

sco[i]=reader.nextDouble();

}

ComputerAverage p=**new** Gymnastics(sco);

System.***out***.printf("%.3f",p.getaverage());

}**else** {

**int** n=reader.nextInt();

String [] no=**new** String[n+10];

**double** sco[]=**new** **double**[n+10];

**for**(**int** i=0;i<n;i++) {

no[i]=reader.next();

sco[i]=reader.nextDouble();

}

ComputerAverage p=**new** School (no,sco,n);

System.***out***.printf("%.3f",p.getaverage());

}

}

}

**interface** ComputerAverage{

**public** **double** getaverage();

}

**class** Gymnastics **implements** ComputerAverage{

**double**[] sco;

Gymnastics(){}

Gymnastics(**double**[] sco){

**this**.sco=sco;

}

**public** **double** getaverage() {

**double** min=sco[0],max=sco[0],sum=0;

**for**(**int** i=0;i<10;i++) {

**if**(sco[i]>max) max=sco[i];

**if**(sco[i]<min) min=sco[i];

sum+=sco[i];

}

sum=sum-min-max;

**return** sum/8;

}

}

**class** School **implements** ComputerAverage{

String [] no;

**double** [] sco;

**int** n;

School(){}

School(String [] no,**double** [] sco,**int** n){

**this**.no=no;

**this**.sco=sco;

**this**.n=n;

}

**public** **double** getaverage() {

**double** sum=0;

**for**(**int** i=0;i<n;i++) {

sum+=sco[i];

}

**return** sum/n;

}

}

（3）

测试数据

第1组：1

abc

3

第2组：2

4

-2 4 5 0

3

第3组：2

2

1 0

3

程序代码：**package** prj\_multijisuan;

**import** java.util.Scanner;

**public** **class** Main {

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

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

**int** c=reader.nextInt();

Text t=**new** Text();

**if**(c==1) {

reader.nextLine();

MyString s=**new** MyString(reader.nextLine());

t.n=reader.nextInt();

t.mul(s);

}**else** {

**int** m=reader.nextInt(),a[]=**new** **int**[m+10];

**for**(**int** i=0;i<m;i++){

a[i]=reader.nextInt();

}

MyArray s=**new** MyArray(m,a);

t.n=reader.nextInt();

t.mul(s);

}

}

}

**interface** Mul{

**public** **void** multiply(**int** n);

}

**class** MyString **implements** Mul{

String s;

MyString(){}

MyString(String s){

**this**.s=s;

}

**public** **void** multiply(**int** n){

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

**for**(**int** i=1;i<=n;i++) {

System.***out***.printf("%s",s);

}

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

}

}

**class** MyArray **implements** Mul{

**int** m,a[]=**new** **int**[10000];

MyArray(){}

MyArray(**int** m,**int**[] a){

**this**.m=m;

**for**(**int** i=0;i<m;i++) {

**this**.a[i]=a[i];

}

//this.a=a;

}

**public** **void** multiply(**int** n){

System.***out***.printf("%d ",m);

**for**(**int** i=0;i<m;i++ ) {

**if**(i!=m-1) {

System.***out***.printf("%d,",a[i]);

}**else** {

System.***out***.printf("%d\n",a[i]);

}

}

System.***out***.printf("%d ",m\*n);

**for**(**int** i=0;i<m\*n;i++ ) {

**if**(i!=m\*n-1) {

System.***out***.printf("%d,",a[i%m]);

}**else** {

System.***out***.printf("%d\n",a[i%m]);

}

}

}

}

**class** Text{

**int** n;

**public** **void** mul(Mul p){

p.multiply(n);

}

}

3.设计型实验

（1）

测试数据

第1组：3

x001 cc male 10000 1200

x003 bb female 12000 1200

x002 aa male 8000 2000

第2组：2

x001 cc male 10000 1200

x003 bb female 12000 1200

第3组：1

x003 bb female 12000 1200

程序代码：**package** prj\_daikuanshenqing;

**import** java.util.Scanner;

**public** **class** Main {

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

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

**int** n=reader.nextInt();

Graduate []stu=**new** Graduate[n+50];

**for**(**int** i=0;i<n;i++) {

stu[i]=**new** Graduate(reader.next(),reader.next(),reader.next(),reader.nextInt(),reader.nextInt());

stu[i].print();

}

}

}

**interface** StudentInterface{

**void** setFee();

**int** getFee();

}

**interface** TeacherInterface{

**void** setPay();

**int** getPay();

}

**class** Graduate **implements** StudentInterface,TeacherInterface{

String no,name,sex;

**int** fee,pay,yearpay,yearfee;

Graduate(){}

Graduate(String no,String name,String sex,**int** fee,**int** pay){

**this**.no=no;

**this**.name=name;

**this**.sex=sex;

**this**.fee=fee;

**this**.pay=pay;

}

**public** **void** setFee() {

}

**public** **int** getFee() {

**return** fee;

}

**public** **void** setPay() {

}

**public** **int** getPay() {

**return** pay;

}

**public** String is() {

yearpay=pay\*12;

yearfee=fee\*2;

**if**(yearpay-yearfee<2000) **return** "provide a loan";

**else** **return** **null**;

}

**public** **void** print() {

**if**(is()!=**null**)

System.***out***.println(no+","+name+","+is());

}

}

# 实验结果及分析

1.验证型实验

（1）分析本程序的执行过程，给出其运行结果

执行过程：实现Common接口的抽象方法的Car类的对象的引用赋给Common接口的变量，接口变量调用car的方法求出速度，进而求出时间；同理将car类换为Plane类重复上述操作。

运行结果：Car的平均速度: 2.5 km/h

Car的运行时间：400.0 小时

Plane的平均速度: 80.0 km/h

Plane的运行时间：12.5 小时

2.应用型实验

（1）

各个类、方法的作用：主类输入调用输出；Shape抽象出重要的方法求面积；circle求圆的面积；juxing求矩形面积；threeangle求三角形面积；tixing求梯形面积；shapetest统一接口调用。

运行结果：1：27.1416

2：29.5616

3：52.8365

（2）

各个类、方法的作用：主类输入判断调用输出；ComputerAverage抽象出计算平均值的重要方法；Gymnastics计算选手评分的平均分；School计算学生的平均分。

运行结果：1：8.975

2：92.400

3：93.000

3.设计型实验

（1）

各个类、方法的作用：主类输入判断调用输出；Mul抽象出乘方的重要方法；MyString定义字符串的乘法；MyArray定义数组的乘法；Text统一接口调用。

运行结果：1：abc

Abcabcabc

2：4 -2,4,5,0

12 -2,4,5,0,-2,4,5,0,-2,4,5,0

3：2 1,0

6 1,0,1,0,1,0

（2）

各个类、方法的作用：主类输入调用输出； StudentInterface抽象出学生管理的重要方法；TeacherInterface抽象出教师管理的重要方法；Graduate实现以上两个接口的方法，增加判断研究生是否需要申请贷款的方法。

运行结果：1：x001,cc,provide a loan

x003,bb,provide a loan

2：x001,cc,provide a loan

x003,bb,provide a loan

3：x003,bb,provide a loan

# 实验小结

[本次实验的反思、收获和体会]

反思：1：今天才知道输入一整行用reader.nextLine()，使用这个还要考虑刚输完整数后的换行，第三题的输入的字符串可能有空格还是同学告诉我的。

2：在写第三题时在其他类中进行输入的方法之前用过结果忘了，学习了之后使用结果oj上显示运行错误，改为主类输入后就没问题了，说实话还是有些不理解。

收获：1：写第四题时才知道接口也可以声明数组，以及返回空串是返回null。

2：对课上说的接口的使用有了进一步的理解，并尝试了用接口实现多态。

体会：随着java的进一步深入学习，感到对有些题目的类和接口的结构设计模模糊糊的，相关知识框架还未搭建好，需要去学习。以及书上的例题和课后习题应该多思考思考。感觉最近看视频学习的效果不是很好，少有自己的思考，可以结合看书做题改善一下学习方法。