**河南工业大学 Java程序设计实验 实验报告**

专业班级： 计科2101 学号： 211040100127 姓名： 翟向阳 指导老师： 赵晨阳

1. **实验题目**：实验（四）：Java中的类继承机制
2. **实验目的**：1. 理解子类和父类；2. 理解类继承机制；3.理解多态。
3. **实验要求：1. 每个题目，写出实验思路；2. 源代码实现，不要代码截图；3. 运行结果截图。**
4. **实验内容**：

1.声明一个父类Person类，类中成员变量为name(String类型)、age(int类型)、gender(char类型)属性，要求通过构造方法进行赋值。类中声明show方法，返回String类型，内容如下：某某，男（女），年龄。

声明一个子类Student类，继承Person类，增加id(int，学号)属性，编写构造方法，利用super调用父类构造方法来进行变量初始化。重写父类的show方法，返回String类型，内容如下：某某，男（女）， 年龄， 学号。

2.声明一个Shape抽象类，其中有计算面积（area）、周长（perimeter）的方法，有以下几个子类实现：Circle(圆)，Rectangle(矩形)，Triangle(三角形)，重写计算面积、周长的方法。请编写程序并测试，以多态的方式表示出来。

3.设计一个动物声音“模拟器”，希望模拟器可以模拟许多动物的叫声。要求如下：

（1）编写抽象类Animal

Animal抽象类有2个抽象方法cry()和getAnimaName()，即要求各种具体的动物给出自己的叫声和种类名称。

（2）编写模拟器类Simulator

该类有一个playSound(Animal animal)方法，该方法的参数是Animal类型。即参数animal可以调用Animal的子类重写的cry()方法播放具体动物的声音、调用子类重写的getAnimalName()方法显示动物种类的名称。

（3）编写Animal类的子类：Dog，Cat类

图是Simulator、Animal、Dog、Cat的UML图。

UML类图

|  |
| --- |
| **Dog** |
|  |
| cry():void e |

|  |
| --- |
| **Cat** |
|  |
| cry():void |

|  |
| --- |
| ***Animal*** |
|  |
| cry():void  getAnimalName():String |

|  |
| --- |
| **Simulator** |
|  |
| playSound(Animal):void |

（4）编写主类Application（用户程序），要求实现多态，并测试上述功能。

4.编写一个Java程序：（1）设计一个abstract类Employee，其中有一个abstract方法earnings()；（2）设计Employee的二个子类YearWorker和MonthWorker，分别对应按年领取薪水（年薪为100000元）、按月领取薪水（月薪为5000元），子类必须重写父类的earnings()方法；（3）设计一个Company类，该类用Employee对象数组作为成员，Employee对象数组中的元素可以是YearWorker对象或MonthWorker对象的上转型对象；（4）编写一个测试类，假设某公司共有10名员工，两种类型的员工各为5人（其中数组下标为奇数的员工为MonthWorker，其他为YearWorker），计算并输出该公司一年需要支付的薪水总额。

5.假定根据学生的3门学位课程的分数决定是否可以拿到学位，对于本科生，如果3门课程的平均分数超过60分即表示通过，而对于研究生，则需要平均超过80分才能通过。请完成以下java类的设计：（1）设计一个抽象类student，描述学生的共同特征；（2）设计一个描述本科生的类Undergraduate，该类继承并扩展Student类；（3）设计一个描述研究生的类Graduate，该类继承并扩展Student类；（4）设计一个测试类StudentDemo，分别创建本科生和研究生这两个对象，要求实现多态，并输出相关信息。

6.定义一个复数类Complex，要求（1）显示该复数；（2）求两个复数的和，返回复数类对象；（3）求两个复数的差，返回复数类对象；（4）求两个复数的乘积，返回复数类对象；（5）判断两个复数是否相等，返回布尔类型。

1. **实验执行：**

1. 声明一个父类Person类，类中成员变量为name(String类型)、age(int类型)、gender(char类型)属性，要求通过构造方法进行赋值。类中声明show方法，返回String类型，内容如下：某某，男（女），年龄。

声明一个子类Student类，继承Person类，增加id(int，学号)属性，编写构造方法，利用super调用父类构造方法来进行变量初始化。重写父类的show方法，返回String类型，内容如下：某某，男（女）， 年龄， 学号。

（1）设计思路

按照题目的实验要求来写就好，一步一步实现。

（2）代码实现（源代码粘贴，不要截图）

public class test1 {

public static void main(String[] args) {

Person person = new Person("李四",40,'男');

Student student = new Student(10001,"张三",18, '男');

person.show();

student.show();

}

}

class Person{

private String name;

private int age;

private char gender;

Person(String name,int age,char gender){

this.age=age;

this.name=name;

this.gender=gender;

}

Person(){}

public void show(){

System.out.println("Person信息如下：");

System.out.println("姓名："+this.name+"\t性别："+this.gender+"\t年龄："+this.age);

}

public int getAge() {

return age;

}

public String getName() {

return name;

}

public char getGender() {

return gender;

}

}

class Student extends Person{

private int id;

Student(int id,String name,int age,char gender){

super(name, age, gender);

this.id=id;

}

public int getId() {

return id;

}

@Override

public void show() {

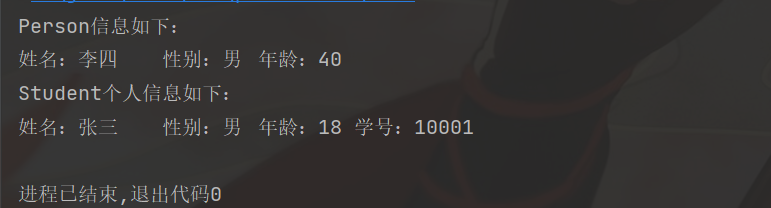
System.out.println("Student个人信息如下：");

System.out.println("姓名："+getName()+"\t性别："+getGender()+"\t年龄："+getAge()+"\t学号："+getId());

}

}

（3）运行结果截图



2.声明一个Shape抽象类，其中有计算面积（area）、周长（perimeter）的方法，有以下几个子类实现：Circle(圆)，Rectangle(矩形)，Triangle(三角形)，重写计算面积、周长的方法。请编写程序并测试，以多态的方式表示出来。

（1）

先声明一个抽象的Shape类，里边放GetArea()和GetPerimeter()两个抽象函数，然后再声明三个类分别为Circle，Rectangle,Triangle类，在这三个类中定义各自的成员变量，之后重写Shape类中的两个抽象函数，最后在测试类中分别创建对象，调用相应的方法即可

（2）

public class test2 {

public static void main(String[] args) {

Circle circle = new Circle(2);

System.out.println("输出圆的面积:\t\t"+circle.GetArea()+"\t周长:"+circle.GetPerimeter());

Rectangle rectangle = new Rectangle(2,3);

System.out.println("输出矩形的面积:\t"+rectangle.GetArea()+"\t\t周长:"+rectangle.GetPerimeter());

Triangle triangle = new Triangle(3,4,5);

System.out.println("输出三角形的面积:\t"+triangle.GetArea()+"\t\t周长:"+triangle.GetPerimeter());

}

}

abstract class Shape{

abstract double GetArea();

abstract double GetPerimeter();

}

class Circle extends Shape{

private double r;

Circle(int r){

this.r=r;

}

Circle(){}

public void setR(double r) {

this.r = r;

}

double GetArea(){

return 3.14\*r\*r;

}

double GetPerimeter(){

return 2\*3.14\*r;

}

}

class Triangle extends Shape{

double a,b,c;

Triangle(double a,double b,double c){

this.a=a;

this.b=b;

this.c=c;

}

double GetArea(){

double p = GetPerimeter()/2;

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

}

double GetPerimeter(){

return a+b+c;

}

}

class Rectangle extends Shape{

double a,b;

Rectangle(double a,double b){

this.a=a;

this.b=b;

}

double GetArea(){

return a\*b;

}

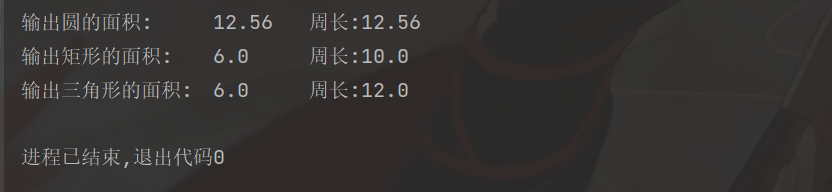
double GetPerimeter(){

return (a+b)\*2;

}

}

（3）



3.设计一个动物声音“模拟器”，希望模拟器可以模拟许多动物的叫声。要求如下：

（1）编写抽象类Animal

Animal抽象类有2个抽象方法cry()和getAnimaName()，即要求各种具体的动物给出自己的叫声和种类名称。

（2）编写模拟器类Simulator

该类有一个playSound(Animal animal)方法，该方法的参数是Animal类型。即参数animal可以调用Animal的子类重写的cry()方法播放具体动物的声音、调用子类重写的getAnimalName()方法显示动物种类的名称。

（3）编写Animal类的子类：Dog，Cat类

（4）编写主类Application（用户程序），要求实现多态，并测试上述功能。

图是Simulator、Animal、Dog、Cat的UML图。

UML类图

|  |
| --- |
| **Dog** |
|  |
| cry():void e |

|  |
| --- |
| **Cat** |
|  |
| cry():void |

|  |
| --- |
| ***Animal*** |
|  |
| cry():void  getAnimalName():String |

|  |
| --- |
| **Simulator** |
|  |
| playSound(Animal):void |

（1）

按照题目要求先声明抽象类Animal类，里面包含cry,getAnimalName两个抽象函数，再声明Simulator类,在其中的playSound函数中调用animal中声明的cry和getAnimalName函数，最后声明Dog和Cat两个具体的动物类，分别在其中实现Animal中声明的两个抽象函数。在测试类中测试即可。

（2）

public class test3 {

public static void main(String[] args) {

Dog dog = new Dog();

Cat cat = new Cat();

Simulator simulator = new Simulator();

simulator.playSound(dog);

simulator.playSound(cat);

}

}

abstract class Animal{

abstract void cry();

abstract String getAnimalName();

}

class Simulator{

public void playSound(Animal animal){

System.out.println("name："+animal.getAnimalName());

animal.cry();

}

}

class Dog extends Animal{

public void cry() {

System.out.println("汪汪汪~~~~");

}

public String getAnimalName() {

return "Dog";

}

}

class Cat extends Animal{

@Override

void cry() {

System.out.println("喵喵喵~~~~");

}

@Override

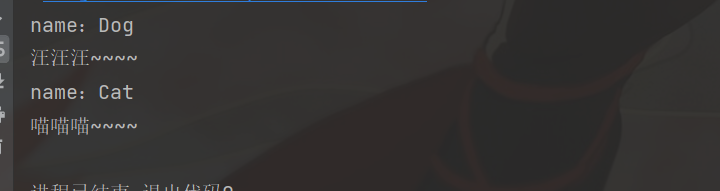
String getAnimalName() {

return "Cat";

}

}

（3）



4.编写一个Java程序：（1）设计一个abstract类Employee，其中有一个abstract方法earnings()；（2）设计Employee的二个子类YearWorker和MonthWorker，分别对应按年领取薪水（年薪为100000元）、按月领取薪水（月薪为5000元），子类必须重写父类的earnings()方法；（3）设计一个Company类，该类用Employee对象数组作为成员，Employee对象数组中的元素可以是YearWorker对象或MonthWorker对象的上转型对象；（4）编写一个测试类，假设某公司共有10名员工，两种类型的员工各为5人（其中数组下标为奇数的员工为MonthWorker，其他为YearWorker），计算并输出该公司一年需要支付的薪水总额。

（1）

利用上转型对象，在Company类中声明Employee对象数组，初始值为11，在Company中的构造函数中将下标为偶数的new成YearWorker（），将下标奇数的new成MonthWorker，最后在totalmoney函数中利用上转型对象所调用的方法是子类中所重写的实例方法，分别调用各自的方法来计算该公司一年的薪水总额。

（2）

public class test4 {

public static void main(String[] args) {

Company company = new Company();

System.out.println("该公司一年需要支付的薪水总额:"+company.totalMoney());

}

}

abstract class Employee{

abstract double earnings();

}

class YearWorker extends Employee{

@Override

double earnings() {

return 100000;

}

}

class MonthWorker extends Employee{

@Override

double earnings() {

return 5000\*12;

}

}

class Company{

Employee[] employees = new Employee[11];

Company(){

for(int i=1;i<11;i++){

if(i%2==0){

employees[i] = new YearWorker();

}

else

employees[i] = new MonthWorker();

}

}

public double totalMoney(){

double sum=0;

for(int i=1;i<11;i++){

sum+=employees[i].earnings();

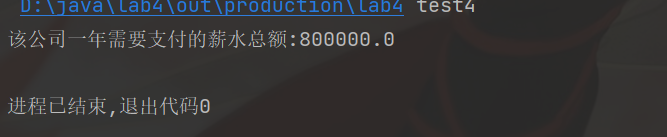
}

return sum;

}

}

（3）



5.假定根据学生的3门学位课程的分数决定是否可以拿到学位，对于本科生，如果3门课程的平均分数超过60分即表示通过，而对于研究生，则需要平均超过80分才能通过。请完成以下java类的设计：（1）设计一个抽象类student，描述学生的共同特征；（2）设计一个描述本科生的类Undergraduate，该类继承并扩展Student类；（3）设计一个描述研究生的类Graduate，该类继承并扩展Student类；（4）设计一个测试类StudentDemo，分别创建本科生和研究生这两个对象，要求实现多态，并输出相关信息。

（1）

将本科生和研究生所共有的东西声明在抽象类student\_5中，（math,java,English,abstract IsPass(),GetAverage()）,之后分别在Underground类和Ground类中实现IsPass()这个抽象方法，最后在test5类中测试。

（2）

import java.util.Scanner;

public class test5 {

public static void main(String[] args) {

System.out.println("本科生：");

Student\_5 undergraduate = new Undergraduate();

undergraduate.IsPass();

System.out.println("研究生：");

Student\_5 graduate = new Graduate();

graduate.IsPass();

}

}

abstract class Student\_5{

double math;

double java;

double english;

abstract void IsPass();

double GetAverage() {

return (math+java+english)/3;

}

Student\_5(){

Scanner sc = new Scanner(System.in);

System.out.println("请按顺序输入你的各科成绩：(math,java,english)");

math = sc.nextDouble();

java = sc.nextDouble();

english = sc.nextDouble();

}

}

class Undergraduate extends Student\_5{

Undergraduate(){

super();

}

public void IsPass(){

System.out.println("平均成绩："+GetAverage());

if(GetAverage()>=60){

System.out.println("恭喜你本科生，可以拿到学位证");

}

else {

System.out.println("很遗憾本科生，不能拿到学位证");

}

}

}

class Graduate extends Student\_5{

Graduate(){

super();

}

public void IsPass(){

System.out.println("平均成绩："+GetAverage());

if(GetAverage()>=80){

System.out.println("恭喜你研究生，可以拿到学位证");

}

else {

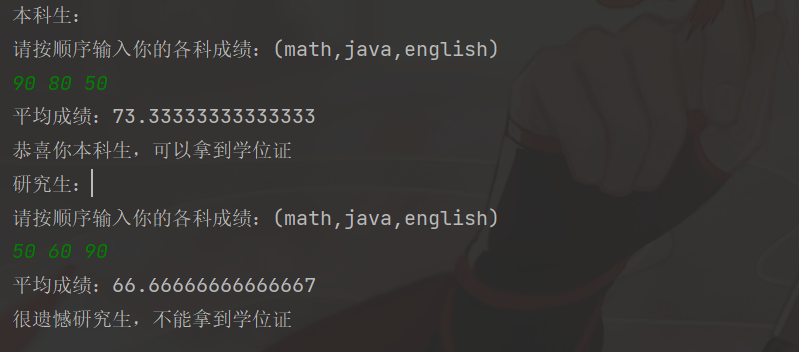
System.out.println("很遗憾研究生，不能拿到学位证");

}

}

}

（3）



6.定义一个复数类Complex，要求（1）显示该复数；（2）求两个复数的和，返回复数类对象；（3）求两个复数的差，返回复数类对象；（4）求两个复数的乘积，返回复数类对象；（5）判断两个复数是否相等，返回布尔类型。

（1）

按照题目的要求操作，在show方法中，对于不同的参数，有不同的显示效果，比如加减号的显示等，复数的和就是两个复数的实部相加，虚部相加；差就是实部相减，虚部相减；乘积则是将按照乘法规则进行，之后相加即可。判断两个复数是否相等则分别判断其实部和虚部是否分别相等。

（2）

public class test6 {

public static void main(String[] args) {

Complex x = new Complex(3,2);

Complex y = new Complex(2,3);

Complex c =new Complex();

System.out.print("x:");

x.show();

System.out.print("y:");

y.show();

System.out.print("c:");

c.show();

System.out.print("加法:");

c = c.add(x,y);

c.show();

System.out.print("减法:");

c = c.sub(x,y);

c.show();

System.out.print("乘法:");

c = c.mul(x,y);

c.show();

System.out.print("是否相等:"+x.IsEqual(y));

}

}

class Complex{

private int a;//实部

private int b;//虚部

public int getA() {

return a;

}

public int getB() {

return b;

}

public void show(){

if(this.getB()>0)

System.out.println(getA()+"+"+getB()+"i");

else if(this.getB()==0)

System.out.println(this.getA());

else

System.out.println( getA()+""+getB()+"i");

}

Complex(int a,int b){

this.a=a;

this.b=b;

}

Complex(){

a=0;

b=0;

}

public Complex add(Complex x,Complex y){

Complex c = new Complex();

c.a = x.getA()+y.getA();

c.b = x.getB()+y.getB();

return c;

}

public Complex sub(Complex x,Complex y){

Complex c =new Complex();

c.a = x.getA()-y.getA();

c.b = x.getB()-y.getB();

return c;

}

public Complex mul(Complex x,Complex y){

Complex c = new Complex();

c.a = x.getA()\*y.getA()-x.getB()\*y.getB();

c.b = x.getA()\*y.getB()+x.getB()\*y.getA();

return c;

}

public boolean IsEqual(Complex x){

return x.getA() == this.getA() && x.getB() == this.getB();

}

}

（3）



1. **实验测试中遇到的问题和自己的解决方案**

在这次测试中加深了对java中继承机制的了解，还有通过继承来实现多态，调用同一个函数却可以做出不同的行为，还有java中的封装性，在这次实验中都有更深的了解，强化了自己的动手能力，刚开始写的写的时候一步一步挺慢的，到后来越来越熟练。