学号 E21514033 专业 软件工程 姓名 何铭春

实验日期 **2018.4.26**  教师签字 成绩

实验报告

【实验名称】 Java实验二

【实验目的】

1. 熟悉Java继承和接口机制，能够熟练使用。
2. 能够在子类中使用父类方法，进行代码复用。

【实验原理】

1. 利用super能够在子类使用父类的方法。
2. 接口中函数主动添加了public 和 abstract，但是子类实现时需要注意。

【实验内容】

题目一：图形类继承

源代码：

Shape类：

public class Shape {

String color;

public Shape() {

}

public Shape(String aColor) {

color = aColor;

}

public void show(){

System.out.println(color);

}

}

圆形类：

public class Circle extends Shape {

double r;

public Circle(){}

public Circle(String aColor,double aR){

super(aColor);

r = aR;

}

public void show(){

System.out.println("颜色为"+color);

System.out.println("半径为"+r);

}

}

矩形类：

public class Rectangle extends Shape{

double a;

double b;

public Rectangle(){}

public Rectangle(String aColor,double theA,double theB){

super(aColor);

a = theA;

b = theB;

}

public void show(){

System.out.println("颜色为"+color);

System.out.println("长为"+a);

System.out.println("宽为"+b);

}

}

测试类：

public class TestShape {

public static void main(String[] args) {

Circle test = new Circle("蓝色",5.23);

test.show();

System.out.println("");

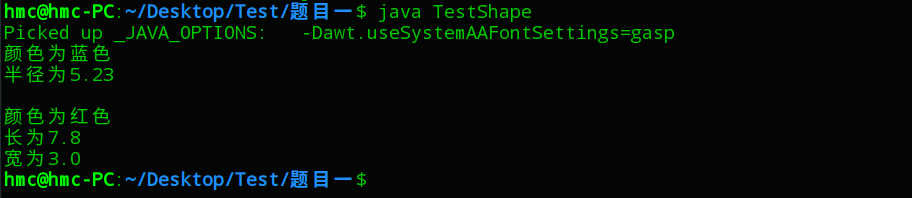
test = new Rectangle("红色",7.8,3);

test.show();

}

}

实验截图：



第二题：定义抽象类Figure，实现area()方法

Figure类：

abstract class Figure {

abstract public double area();

}

Circle类：

public class Circle extends Figure {

private double r;

public Circle(){}

public Circle(double theR){

r = theR;

}

public double area(){

return r\*r\*Math.PI;

}

}

Square类：

public class Square extends Figure {

double border;

public Square(){}

public Square(double aBorder){

border = aBorder;

}

public double area() {

return border\*border;

}

}

测试类：

public class Test{

public static void main(String[] args) {

Figure test = new Circle(4.56);

System.out.println("圆形的面积为"+test.area());

System.out.println("");

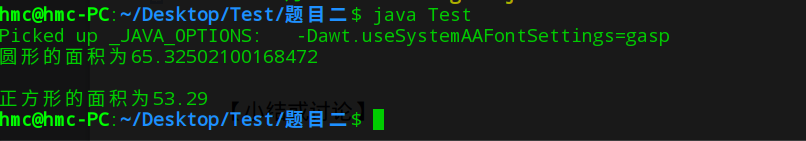
test = new Square(7.3);

System.out.println("正方形的面积为"+test.area());

}

}

实验截图：



题目三：在题目二基础上增加三角形类

新增类：

import java.math.\*;

public class Triangle extends Figure {

double a;

double b;

double c;

public Triangle(){}

public Triangle(double tempA,double tempB,double tempC){

a = tempA;

b = tempB;

c = tempC;

}

public double area(){

double p = a + b + c;

p = p/2;

double s = p\*(p-a)\*(p-b)\*(p-c);

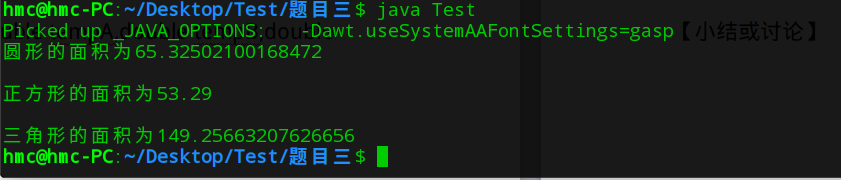
s = Math.sqrt(s);

return s;

}

}

实验截图：



题目四：公约数和公倍数

Father类：

public class Father {

public int f(int a,int b){

int max = 1;

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

if (a % i == 0 && b % i ==0) {

max = i;

}

}

return max;

}

}

Son类：

public class Son extends Father {

public int f(int a,int b){

int max = super.f(a,b);

return (a \* b) / max;

}

}

测试类：

public class Test {

public static void main(String[] args) {

int a =15;

int b =12;

Father Hacky = new Father();

int max = test.f(a,b);

System.out.println("公约数为"+max);

System.out.println("");

Hacky = new Son();

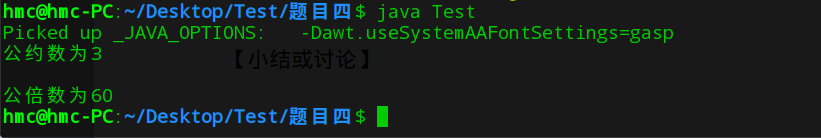
int min = Hacky.f(a,b);

System.out.println("公倍数为"+min);

}

}

实验截图：



题目五：改上一题使用接口

F类：

public abstract class F {

public abstract int f(int a,int b);

}

Max类：

public class Max extends F {

public int f(int a,int b) {

int max = 1;

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

if (a % i == 0 && b % i ==0) {

max = i;

}

}

return max;

}

}

Min类：

public class Min extends Max {

public int f(int a,int b){

int max = super.f(a,b);

return (a \* b) / max;

}

}

测试类：

class Test {

public static void main(String[] args) {

int a =15;

int b =12;

F Hacky = new Max();

int max = Hacky.f(a,b);

System.out.println("公约数为"+max);

System.out.println("");

Hacky = new Min();

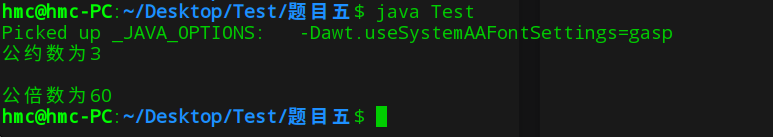
int min = Hacky.f(a,b);

System.out.println("公倍数为"+min);

}

}

实验截图：



题目六：使用接口Shape实现第二题

Shape类：

interface Shape{

double area();

}

子类代码：

public class Square implements Shape{

double border;

public Square(){}

public Square(double aBorder){

border = aBorder;

}

public double area() {

return border\*border;

}

}

public class Circle implements Shape{

private double r;

public Circle(){}

public Circle(double theR){

r = theR;

}

public double area(){

return r\*r\*Math.PI;

}

}

import java.math.\*;

public class Triangle implements Shape{

double a;

double b;

double c;

public Triangle(){}

public Triangle(double tempA,double tempB,double tempC){

a = tempA;

b = tempB;

c = tempC;

}

public double area(){

double p = a + b + c;

p = p/2;

double s = p\*(p-a)\*(p-b)\*(p-c);

s = Math.sqrt(s);

return s;

}

}

测试类：

public class Test{

public static void main(String[] args) {

Shape[] spring = new Shape[3];

spring[0] = new Circle(4.56);

spring[1] = new Square(7.3);

spring[2] = new Triangle(12.3,31.1,25.3);

System.out.println("圆形的面积为"+spring[0].area());

System.out.println("");

System.out.println("正方形的面积为"+spring[1].area());

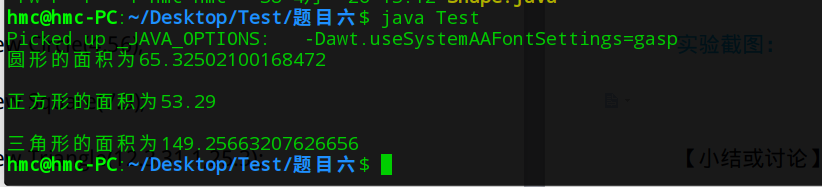
System.out.println("");

System.out.println("三角形的面积为"+spring[2].area());

}

}

实验截图：



【小结或讨论】

1. 使用父类的函数能够极大的简化子类的实现。
2. 接口类规定了必须实现的函数，有利于规范子类。
3. 使用多态能够使用父类数组来指向子类对象。