实验二 **Java**中的类继承机制、接口

实验日期： 2019 年 5 月 10 日 班级： 软件1702

学号（后四位）： 0224 姓名： 刘文博 成绩：

1. 实验目的

1．能够根据问题需求，编写父类、子类，并合理地应用父类与子类之间的继承关系；

2.能够对类进一步抽象出接口，提高程序的重用性和可扩展性。

二、支撑的课程目标

本实验项目可以支撑“课程目标1：能根据软件工程领域的复杂工程问题，进行研究，运用Java的编程思想设计项目开发方案，并按照合理步骤和软件工程流程进行方案的实施”、“课程目标2：掌握Java语言的基本特点、语法和面向对象的编程思想，能应用Java语言的编程原理、方法和技术，针对软件工程领域中复杂工程问题的设计解决方案要求进行软件开发”。

本实验通过“任务驱动”方法，使学生掌握面向对象的基本思想，能够根据项目需求合理提取类，设置类之间的继承关系和合理应用接口，并能够根据Java的编程思想和软件工程的流程进行方案设置，并能根据方案进行软件开发。

1. 实验内容

编写能够满足如下条件的程序：

1. a 声明一个Person类，有name(String类型)、age(int类型)、sex(char类型)属性。

通过构造方法进行赋值。

一个show方法，返回String类型，内容如下：

某某 男（女） 年龄

b 声明一个Student类，继承Person类，增加id(int，学号)属性，通过构造方法，利用super调用父类构造方法来进行变量赋值。Override父类的show方法，返回String类型，内容如下：

某某 男（女） 年龄 学号

提示：利用super调用父类的show方法得到除学号部分的String，然后加上学号的信息。

c 声明一个Teacher类，继承Person，增加course(String，所教课程)属性，通过构造方法，利用super调用父类构造方法来进行变量赋值。Override父类的show方法，返回String类型，内容如下：

某某 男（女）年龄 所教课程

提示：利用super调用父类的show方法得到除所教课程部分的String，然后加上所教课程的信息。

d 声明PersonApp类，在其中的main方法中分别声明Person、Student、Teacher类型的变量，并通过构造方法初始化，然后显示各自的信息。

**public** **class** People {

**private** String name;

**private** **int** age;

**private** **char** sex;

**public** People() {}

**public** People(String name, **int** age, **char** sex) {

**super**();

**this**.name = name;

**this**.age = age;

**this**.sex = sex;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **char** getSex() {

**return** sex;

}

**public** **void** setSex(**char** sex) {

**this**.sex = sex;

}

**p**ublic String show() {

return ("Name:" + this.getName() +"\n"+

"Sex:" + this.getSex() + "\n"+

"Age:" + this.getAge() + "\n");

}

}

**public** **class** Student **extends** People {

**private** String ID;

**public** Student() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Student(String name, **int** age, **char** sex) {

**super**(name, age, sex);

// **TODO** Auto-generated constructor stub

}

**public** Student(String name, **int** age, **char** sex, String iD) {

**super**(name, age, sex);

**this**.ID = iD;

}

**public** String getID() {

**return** ID;

}

**public** **void** setID(String iD) {

ID = iD;

}

**public** String show() {

**return** (**super**.show()+

"ID:" + **this**.getID()+"\n");

}

}

**public** **class** Teacher **extends** People {

**private** String course;

**public** Teacher(String name, **int** age, **char** sex, String course) {

**super**(name, age, sex);

**this**.course = course;

}

**public** Teacher(String course) {

**super**();

**this**.course = course;

}

**public** Teacher() {

**super**();

}

**public** String getCourse() {

**return** course;

}

**public** **void** setCourse(String course) {

**this**.course = course;

}

**public** String show() {

**return** (**super**.show()+

"Scurse:" + **this**.getCourse() + "\n");

}

}

**public** **class** PersonApp {

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

// **TODO** Auto-generated method stub

People hum1 = **new** People("Xiaoming",15,'m');

Student stu1 = **new** Student("Xiaowang",16,'m',"20171604001");

Teacher teac1 = **new** Teacher("Mrs.Dind",35,'f',"English");

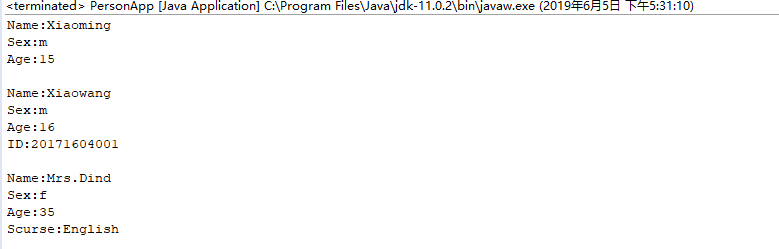
System.*out*.println(hum1.show());

System.*out*.println(stu1.show());

System.*out*.println(teac1.show());

}

}



1. 声明一个Shape接口，其中有计算面积（area）、周长（perimeter）的方法，有以下几个实现：Circle(圆)，Rectangle(矩形)，Triangle(三角形)，都有计算面积、周长的方法。

**public** **interface** shapable {

**double** calcPerimeter();

**double** calcArea();

}

**public** **class** Circle **implements** shapable {

**double** r;

**public** Circle(**double** r) {

**super**();

**this**.r = r;

}

**public** Circle(){

}

@Override

**public** **double** calcPerimeter() {

**return** 2 \* Math.*PI* \* **this**.r;

}

@Override

**public** **double** calcArea() {

**return** Math.*pow*(**this**.r, 2) \* Math.*PI*;

}

}

**public** **class** Rectangle **implements** shapable {

**double** l;

**double** w;

**public** Rectangle(**double** l, **double** w) {

**super**();

**this**.l = l;

**this**.w = w;

}

**public** **double** getL() {

**return** l;

}

**public** **void** setL(**double** l) {

**this**.l = l;

}

**public** **double** getW() {

**return** w;

}

**public** **void** setW(**double** w) {

**this**.w = w;

}

@Override

**public** **double** calcPerimeter() {

**return** 2 \* (**this**.l + **this**.w);

}

@Override

**public** **double** calcArea() {

// **TODO** Auto-generated method stub

**return** **this**.l \* **this**.w;

}

}

**public** **class** Triangle **implements** shapable {

**double** r1;

**double** r2;

**double** r3;

**public** Triangle(**double** r1, **double** r2, **double** r3) {

**super**();

**this**.r1 = r1;

**this**.r2 = r2;

**this**.r3 = r3;

}

**public** **double** getR1() {

**return** r1;

}

**public** **void** setR1(**double** r1) {

**this**.r1 = r1;

}

**public** **double** getR2() {

**return** r2;

}

**public** **void** setR2(**double** r2) {

**this**.r2 = r2;

}

**public** **double** getR3() {

**return** r3;

}

**public** **void** setR3(**double** r3) {

**this**.r3 = r3;

}

@Override

**public** **double** calcPerimeter() {

**return** **this**.r1 + **this**.r2 + **this**.r3;

}

@Override

**public** **double** calcArea() {

**double** p = (**this**.r1 + **this**.r2 + **this**.r3)/2.0;

**return** Math.*sqrt*(p \* (p-**this**.r1) \* (p - **this**.r2) \* (p - **this**.r3));

}

}

**public** **class** ShapeTest {

/\*\*

\* **@param** args

\*/

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

Circle c1 = **new** Circle(3.1);

System.*out*.printf("Circle's perimeter is:%.2f\n" ,c1.calcPerimeter());

System.*out*.printf("Circle's area is:%.2f\n", c1.calcArea());

Rectangle r1 = **new** Rectangle(5.0, 4.0);

System.*out*.printf("Rectangle's perimeter is:%.2f\n" ,r1.calcPerimeter());

System.*out*.printf("Rectangle's area is:%.2f\n", r1.calcArea());

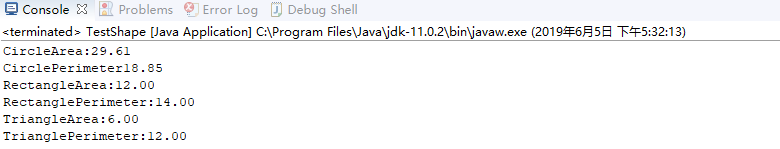
Triangle t1 = **new** Triangle(3.0, 4.0, 5.0);

System.*out*.printf("Triangle's perimeter is:%.2f\n" ,t1.calcPerimeter());

System.*out*.printf("Triangle's area is:%.2f\n", t1.calcArea());

}

}



四 在本次实验中的问题和解决方法

无

五 心得体会

接口中的方法必须在实现类中实现

想要不实例化就调用类中的方法，需要将方法设置为静态方法，使用static关键字