一、实验目的

掌握Java中抽象类和抽象方法的定义；

掌握Java中接口的定义，熟练掌握接口的定义形式以及接口的实现方法；

了解异常的使用方法，并在程序中根据输入情况做异常处理；

二、实验要求

说明：某学校为了给学生提供勤工俭学机会，也减轻授课教师的部分压力，准许博士研究生参与课程的助教工作。此时，该博士研究生有双重身份：学生和助教教师。

内容：

设计两个管理接口：学生管理接口和教师管理接口。学生接口必须包括缴纳学费、查学费的方法；教师接口包括发放薪水和查询薪水的方法。

设计博士研究生类，实现上述的两个接口，该博士研究生应具有姓名、性别、年龄、每学期学费、每月薪水等属性。

编写测试类，并实例化至少两名博士研究生，统计他们的年收入和学费。根据两者之差，算出每名博士研究生的年应纳税金额。

在博士研究生类中实现各个接口定义的抽象方法;

对年学费和年收入进行统计，用收入减去学费，求得纳税额；

国家最新纳税标准（系数），属于某一时期的特定固定值，与实例化对象没有关系，考虑如何用static final修饰定义

实例化研究生类时，可采用运行时通过main方法的参数args一次性赋值，也可采用Scanner类实现运行时交互式输入

根据输入情况，要在程序中做异常处理。

三、实验里的核心代码

//interface 接口

public interface TeacherInterface {

public abstract void setPay(float pay);// 设置教师的工资

public abstract void getPay(float pay);// 获取教师的工资

}

public interface StudentInterface {

public abstract void setFee(float fee);// 设置学生的学费

public abstract void getFee(float fee);// 获取学生的学费

}

public Graduate(String name,String sex,int age){ //有参构造方法

this.name=name;

this.sex=sex;

this.age=age;

}

Graduate() { //无参构造方法

}

//实现属性的封装 (例如)

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

// 实现接口中的抽象方法

public void setPay(float pay) {

this.pay = pay \* 12; //月收入\*12=年收入

System.out.println("年收入为：" + this.pay);

}

public void getPay(float pay) {

this.pay = pay \* 12;

System.out.println("年收入为：" + this.pay);

}

//boolean是java中的布尔型（逻辑型）数据类型

public boolean paytaxes(){

if ((this.pay - this.fee) < 5000) {

System.out.println("纳税金额为:"+(this.pay -this.fee)\*0.03f+"\n");

return true; // true 代表“真”

}

else {

System.out.println("纳税金额为:"+(this.pay -this.fee)\*0.03f+"\n");

return false; //false 代表“假”

}

}

}

1. 实验感想

通过这次实验掌握Java中接口的定义、抽象类和抽象方法的定义，熟练掌握接口的定义形式以及接口的实现方法

接口可以被多重实现（implements）,抽象类只能被单一继承（extends）

理解异常是如何捕获以及抛出的，但对整体的异常理解并不透彻，比如有没有一种异常定义是，定义所有的异常内，然后在通过各种异常判断把最终的异常确定下来