《Java技术》课程上机报告

第 3 次上机

学 号：­­­22009200439

姓 名：赵宇阳

提交日期：2023.5.25

## 任务简要描述

1.请设计 3 个类 , 分别是学生类 Student, 本科生类Undergaduate, 研究生类 Postgraduate, 其中 Student 类是一个抽象类 , 它包含一些基本的学生信息如姓名、所学课程、课程成绩等 , 而Undergraduate 类和 Postgraduate 都是 Student 类的子类 , 它们之间的主要差别是计算课程成绩等级的方法有所不同 , 研究生的标准要比本科生的标准高一些。假设某班级里既有本科生也有研究生 , 请编写程序统计出全班学生的成绩等级并显示出来。

   2.下面程序模拟了饲养两只宠物的内容，补充完整程序代码，并回答问题。

// 所有宠物的父类

**class** Pet {

    // 补充定义宠物属性，至少包括宠物的名字、宠物的食物偏好索引号、宠物的食

// 量，三个成员变量。成员变量访问权限必须定义为私有权限

    // 初始化默认的宠物实例对象

**public** Pet() {

       Name           = "MyPet";

       AmountNeedEat = 1;

       FavoriteIndex = 1;

    }

    // 补充定义根据输入参数，初始化宠物的实例对象

**public** Pet(String Name, **int** FavoriteIndex, **int** AmountNeedEat) {

    }

    // 补充定义返回宠物偏好的食物索引号

**public** **int** getFavoriteIndex() {

}

    // 补充定义返回宠物的名字

**public** String toString() {

    }

    // 补充定义宠物进食。投喂的食物符合偏好，就吃，且饱食度减一，返回已进食；

// 投喂不符合偏好，或饱食度为零时，拒食，返回未进食

**public** **boolean** eat(**int** foodindex) {

    }

    // 补充定义宠物饥饿判断。饱食度为零返回不饿，否则返回饿

   // 宠物根据自己的状态可以干点什么了。

**public** String doSomething() { **return** "doSomething";}

}

// 宠物狗类

**class** Dog **extends** Pet {

// 补充定义需要的成员变量。成员变量的访问权限必须定义为私有权限

    // 根据输入参数，初始化狗的实例对象

**public** Dog(String Name, **int** AmountNeedEat, **int** FavoriteIndex) {

    }

    // 补充定义宠物狗的干事方法。如果没吃饱，返回still want to eat more food.

    // 如果吃饱了，返回Let’s go outside for a wallow.

**public** String doSomething() {

}

}

// 宠物猫类

**class** Cat **extends** Pet {

// 补充定义需要的成员变量。成员变量的访问权限必须定义为私有权限

    // 补充定义根据输入参数，初始化猫的实例对象

**public** Cat(String Name, **int** AmountNeedEat, **int** FavoriteIndex) {

    }

    // 补充定义宠物猫的干事方法。如果没吃饱，返回still want to eat more food.

    // 如果吃饱了，返回Let’s go to bed for a sleep.

**public** String doSomething() {

    }

}

// 宠物饲养程序

**public** **class** PetRaise {

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

       // 随机购买了10种食物，食物偏好索引为1,2

**int** food[] = **new** **int**[10];

**for** (**int** i = 0;i < 10;i ++)

           food[i] = (**int**)(Math.random()\*2) + 1;

       // 补充程序：随机领养了两只宠物

       // 第一个宠物的名字：Tom；食物偏好1；饭量：3

       // 第二个宠物的名字：Pluto；食物偏好2；饭量：6

       // 补充程序：给领养的两个宠物喂食，直到宠物吃饱或无合适食物

       // 补充程序：如果有某个宠物未吃饱，则输出是哪种食物不足

**for** (**int** i = 0;i < 2;i ++) {

       }

       // 补充程序：使用增强for循环带每个宠物活动（doSomething）

    }

}

1)       设计Pet类的目的是什么？

2)       以程序为例说明多态的作用

3．编写一个程序，实现复数运算，要求如下：（3、4二选一）

1)    设计并实现复数类。

2)    设计并实现依据各种基本数值类型值构建复数对象实例的方法。

3)    设计并实现复数的加、减、乘、除运算。以提供任意数值类型数据与复数对象实例自身的运算功能。

4)    参考Integer类源码，重写复数的equals方法、toString方法及hashCode方法。

         5)    构建测试类，该类实现与用户的交互，接收用户键入的数据，并完成类功能的测试。

## 二、问题及解决

任务1:

首先设计抽象类Student,并写好构造函数和默认抽象方法getGradeRank()和printFradeRank()，再新建一个子类Undergraduate继承Student类，并实现抽象方法，同理，新建子类Postgraduate继承实现Student类，实现抽象方法，最后编写多态实现测试类PolymorphicTest类，对两个子类的方法进行测试校验。

任务2：

首先将代码补全，实现Pet基类，再实现Dog,Cat子类，重写部分方法。再在PetRaise类中补全测试代码。

对于1）设计Pet父类一是为了减少代码的重写冗余，实现代码复用，提高效率；二是提取类特征，使得架构层次分明；三是减少耦合，修改一个子类不会影响到另一个子类。

对于2）我认为使用多态有这些优点：

1. 提高了代码的可维护性；
2. 提高了代码的扩展性；
3. 降低代码耦合度。

任务3：

这里3，4中选择3来实现。

先编写ComplexNumber类，实现借助double实例化复数，实部和虚部的计算值都是double，再分别实现复数与实数、复数之间的加减乘除运算，每个方法返回一个复数对象。

再编写复数操作测试类CpxNumTest，在其中打印交互信息，并进行计算、打印结果。