**实验五 继承、多态与接口**

学生姓名： 黄晨箬 学 号： 6109119066 专业班级： 计算机193班

实验类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期： 2021.5.21 实验成绩：

**一、实验目的**

1、掌握继承和多态的概念与实现方法

2、掌握如何从已有的类中派生子类并继承父类。

3、掌握方法的重写（覆盖）和重载。

4、掌握抽象类和接口的定义和实现；

5、掌握数组列表ArrayList类的使用。

**二、实验内容**

**习题1. 编写程序模拟中国人、美国人是人，北京人是中国人，完成相应类别人群的基本信息输出。**除主类外，程序中还有4 个类：People、ChinaPeople、AmericanPeople 和BeijingPeople 类。要求如下：

 People 类有权限是protected 的double 型成员变量height 和weight，以及public void speakHello()、public void averageHeight()和public void averageWeight()方法。

 ChinaPeople 类是People 的子类，新增了public void chinaGongfu()方法。要求ChinaPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。

 AmericanPeople 类是People 的子类，新增public void americanBoxing()方法。要求AmericanPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。

 BeijingPeople 类是ChinaPeople 的子类，新增public void beijingOpera()方法。要求BeijingPeople 重写父类的public void speakHello()、public void averageHeight()和public void averageWeight()方法。

请画出People、ChinaPeople、AmericanPeople 和BeijingPeople 类的UML图及相应关系，然后程序先依次输出ChinaPeople、AmericanPeople 和BeijingPeople共有行为和属性结果，再输出各类特有行为和属性的输出结果，并得到下列参考运行实例效果：



其中People类源文件参考如下：

//People.java

public class People {

protected double weight,height;

public void speakHello() {

System.out.println("您好");

}

public void averageHeight() {

height=173;

System.out.println("average height:"+height);

}

public void averageWeight() {

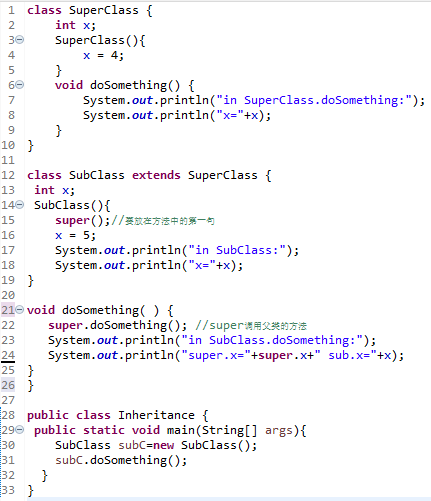
weight=70;

System.out.println("average weight:"+weight);

}

}

**习题2. 阅读以下程序，获得运行结果，并回答下列问题（各问题不串联，请独立分析）。**



问题1：将本程序第13行及第16行删去，程序的输出结果有何变化？为什么？

问题2：使用多行注释，将第21行至第25行内容不执行，该程序能获得输出结果吗？若能，给出结果，并进行适当分析。若不能，请分析原因。

问题3：将第30行、31行分别改为

SuperClass superA = new SubClass(); //第30行

superA.doSomething(); //第31行

程序的输出结果有变化吗？为什么？

**习题3.** 有来自4类（鸟类、昆虫类、爬行类和鱼类）的100个动物聚在一起开会，商议和另一个动物部落进行运动会事宜，会议要求每个动物都要报告自己所属的动物类别和自己的天赋，以便选拔人才、组织队伍参赛。

设计：

1.    用Animal作为基类，类Animal中要有属性成员编号、类别，以及showMessage()方法将所有属性成员信息输出。

2.    设计Talent接口，接口中有抽象方法showTalent()。

3.    鸟类、昆虫类、爬行类和鱼类均继承Animal并实现Talent接口，同时对每个子类至少要添加一个描述的动物特有的行为和一个特有的属性，以更准确地描述子类对象。

使用：用循环随机生成这100个动物装入动物数组，用循环让每个参会的动物报告自己的类别和天赋。 设计时运用继承、重写并设计多态机制。

**习题4-5. 完成课本下列章节的编程题内容。**

**习题4：第11章 课本389页11.13（去掉重复元素）**

**习题5：第11章 课本390页11.17（代数：完全平方）**

**三、实验要求**

1、要求学生在实验前一定要非常清楚并灵活运用涉及章所讲过的内容；

2、在上机实验前编写好实验内容要求的程序，以便上机实验时调试、修改等。

3、上机实验后，每位学生必须对该次实验内容写一份实验报告，包括程序内容、调试过程、运行情况和结果等。

**四、实验环境**

1、PC微机；

2、DOS操作系统或 Windows 操作系统；

3、Eclipse程序集成环境。

**五、实验步骤**

1、根据题目要求，画出UML类图，并标明各类间关系；

2、给出本程序的java数据结构；

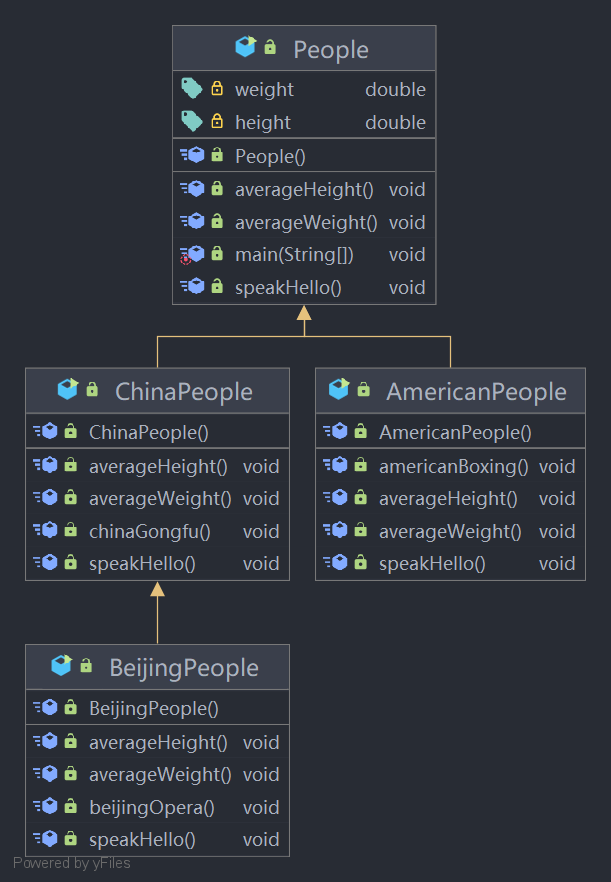
3、编写出程序；

4、调试程序：给出一些测试数据，检查输出结果。

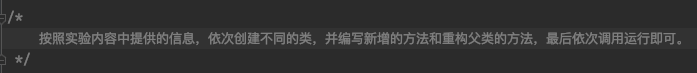
**六、实验数据及处理结果**

**习题一：**

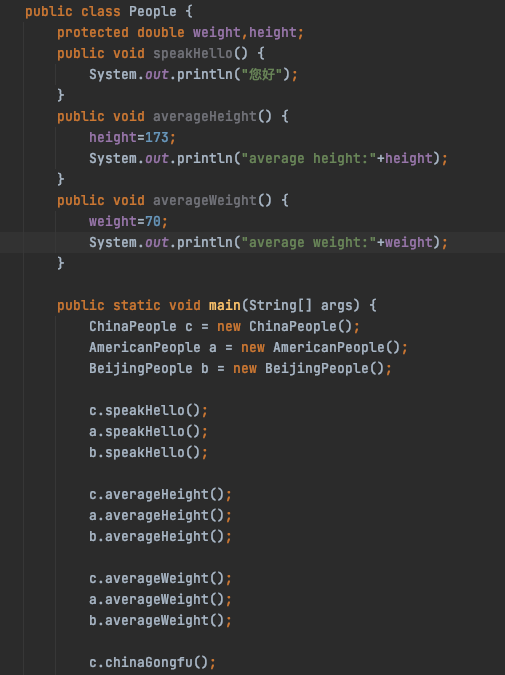
[程序流程图]



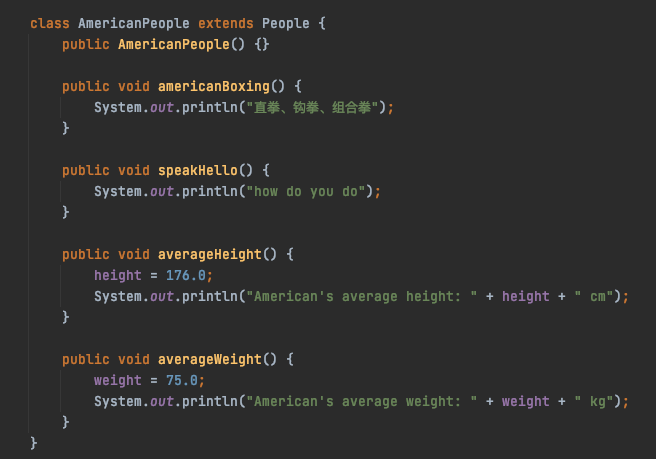
[数据结构设计]

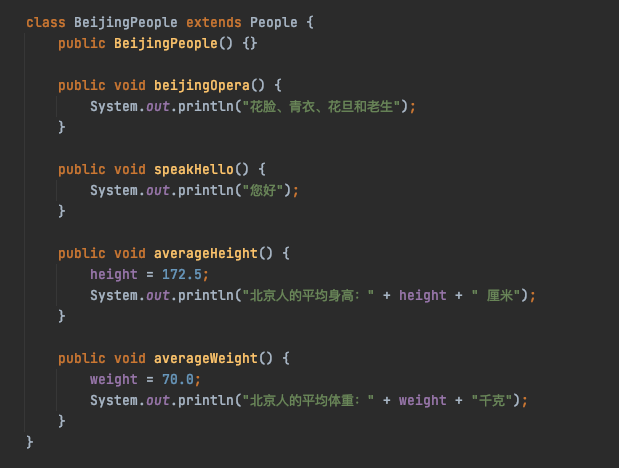


[程序源代码]

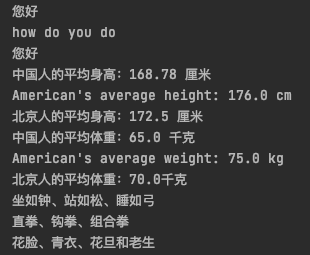






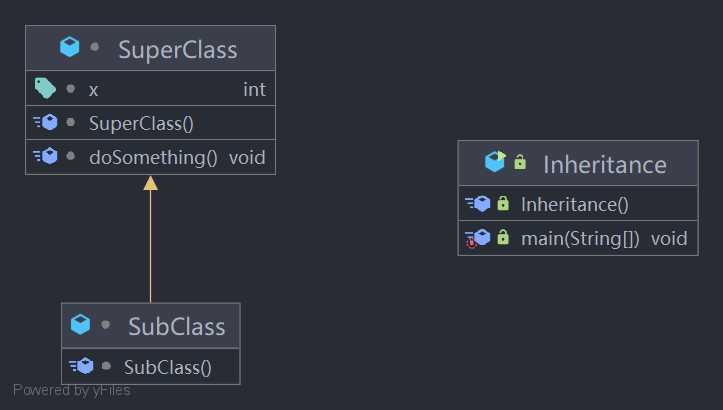


[程序运行结果]



**习题二：**

[程序流程图]



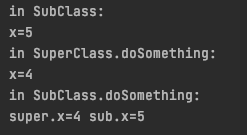
[数据结构设计]

省略

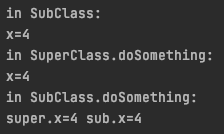
[程序源代码]



[程序运行结果]

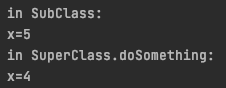


如果删除13行和16行



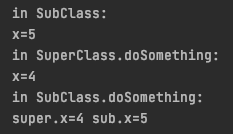
如果删除这两行，在SubClass中的x的值会继承SuperClass里的值，所以全部显示为4。

如果将21-25行注释掉，依然可以运行



当注释掉SubClass中的doSomething方法后，subC.doSomething(); 指令就不会执行，但是其他的指令仍然会正常运行，所以最后也能成功编译。

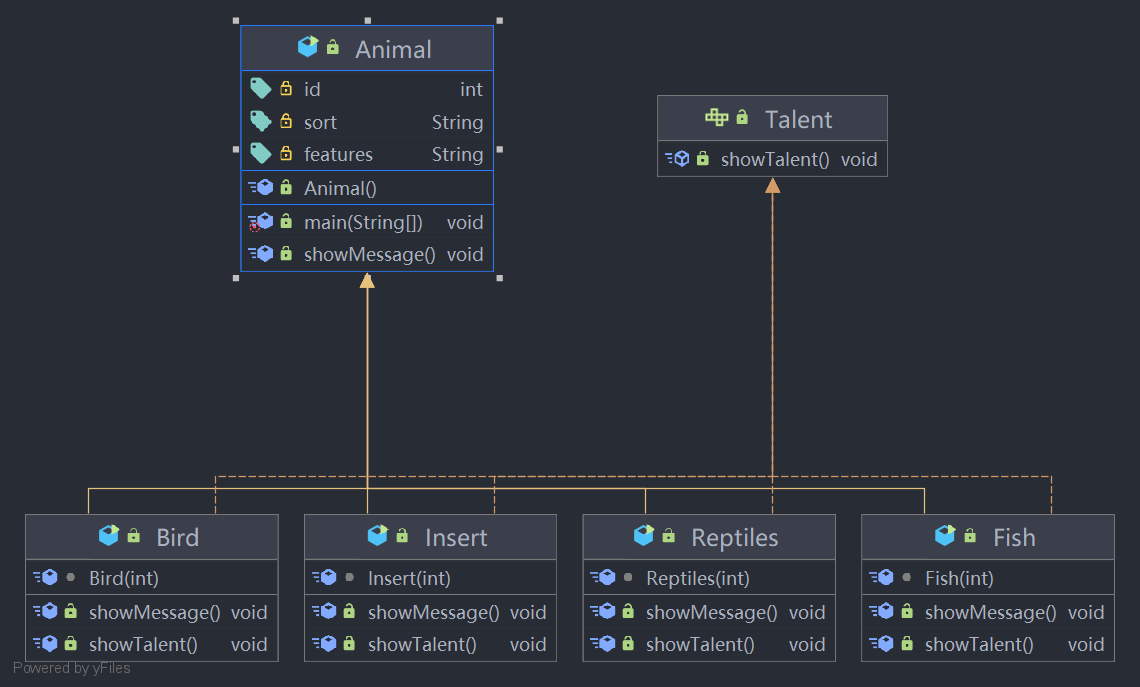
当修改了30-31行的内容之后，输出结果没有发生改变



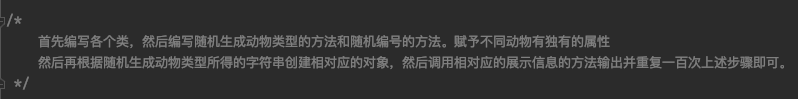
经过观察我们知道，实际上修改的内容只是将subClass修改为了superClass，但实际上，此处是用于声明对象类型，而他们的对象类型是一样的，所以最后的输出结果不会受到影响。

**习题三：**

[程序流程图]

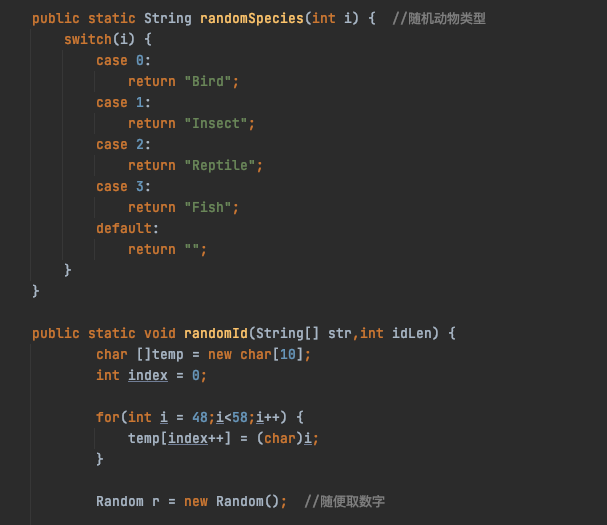


[数据结构设计]

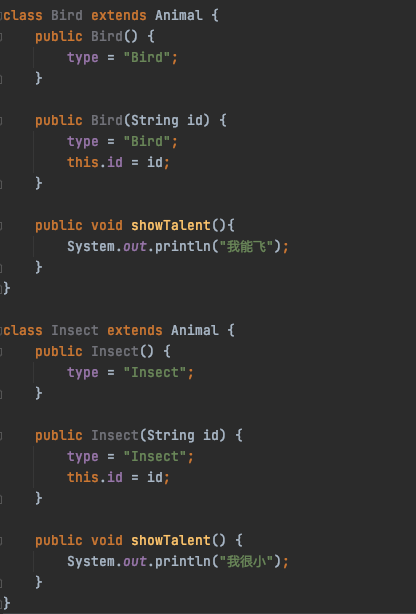


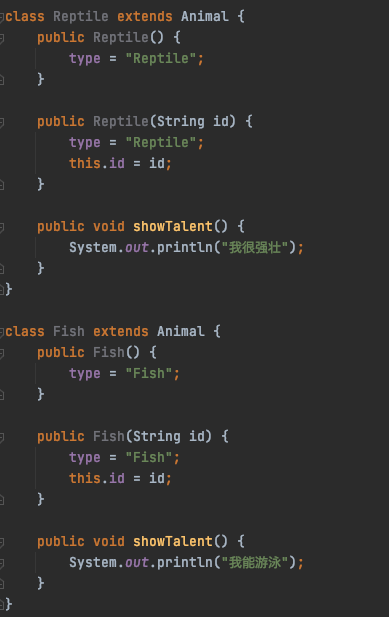
[程序源代码]



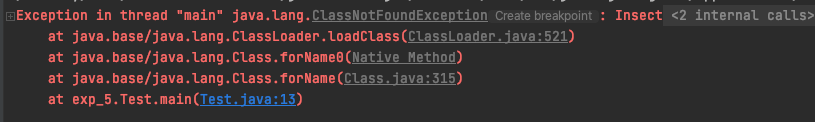








[程序运行结果]

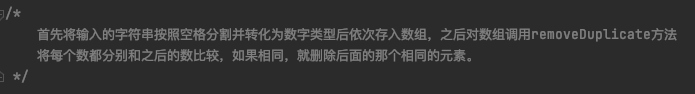


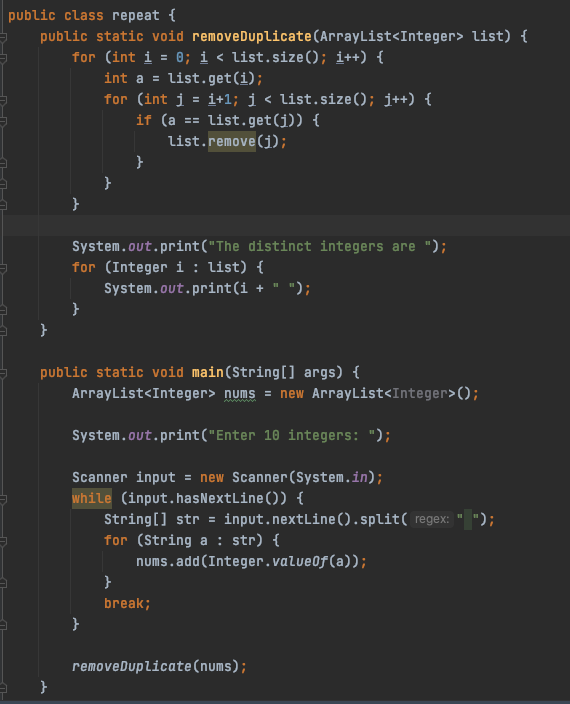
这里应该是步骤中的“根据字符串内容生成对应的对象”这一步的代码错误，一开始查询发现使用forName和newInstance方法可以达到目标，但是实际上手后发现newInstance方法已经被弃用，于是改用getDeclaredConstructor().newInstance(),运行后仍然报错，多次查询无果。

**习题四：**

[程序流程图]

省略

[数据结构设计]  [程序源代码]



[程序运行结果]

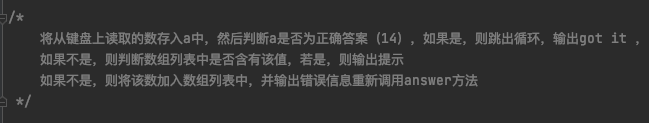


**习题五：**

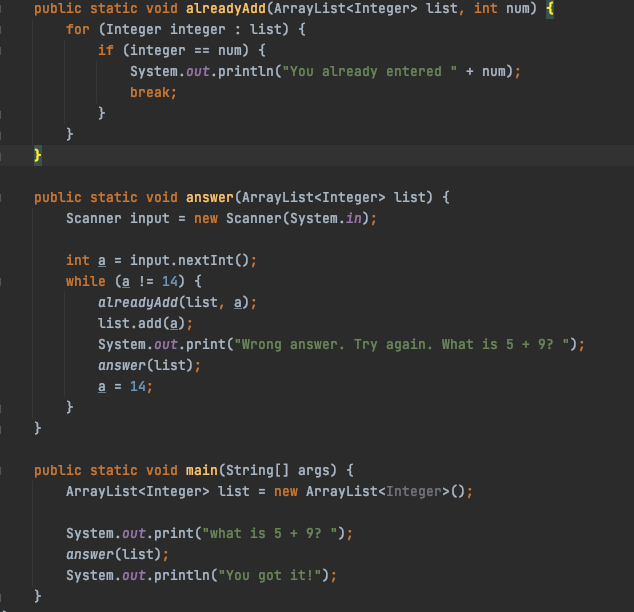
[程序流程图]

省略

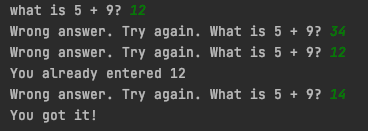
[数据结构设计]



[程序源代码]



[程序运行结果]



**七、思考题**

1．试说明super关键字和this关键字的含义及主要用途。

在Java中，this通常指当前对象，super则指父类。

this可以用于获取当前对象的方法或某个成员，而super则用于引用父类的某种东西。

2. 试描述Java程序使用抽象类和接口实现类间继承机制的异同。

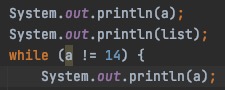
同：接口和抽象类都不能被实例化，只能被其他类实现和继承

异：接口允许多继承

**八、实验总结与体会**

经过这次实验，学习了继承和多态的概念与实现方法，掌握了抽象类和接口的相关知识，尝试了ArrayList类的使用。

在实验过程中，第五个实验调试时，发现了问题：



在经过反复测试之后，发现是由于a的值在循环中没有更新，导致即使下一次循环从键盘接收的值为14，也照样会进行循环。解决办法是在while循环的最后加上 a = 14。调试后正常运行。

经过本次错误，我学到了之后的实验过程中，需要花费更多的心思，细细推敲错误的发生，并且能做到在问题出现后沉着冷静的发现问题并对应解决。