**实验五 继承、多态与接口**

学生姓名： 胡权富 学号： 8002119097 专业班级： 计算机193

实验类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期： 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() {

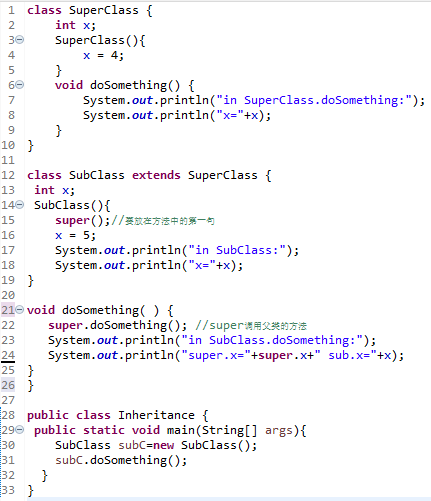
weight=70;

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**写出各个习题的算法（设计思路）、程序、结果并能对结果的正确性及产生的问题进行适当分析。

实验报告应包括以下几个部分：

1、程序流程图或UML类图；

2、程序的数据结构设计；

3、程序的源代码及相关注释

4、程序运行结果的分析。

**题一：**

[源代码]

People.java

1. **package** T1;
3. **public** **class** People {
4. **protected** **double** height;
5. **protected** **double** weight;
6. **public** **void** speakHello(){System.out.println("您好");}
7. **public** **void** averageHeight(){
8. height=13;
9. System.out.println("average height:"+height);
10. }
11. **public** **void** averageWeight(){
12. weight=70;
13. System.out.println("average weight:"+weight);}
14. }

ChinaPeople.java

1. **package** T1;
3. **public** **class** ChinaPeople **extends** People {
4. **public** **void** ChinaGongFu(){;}
5. **public** **void** speakHello(){
6. System.out.println("您好");
7. }
8. **public** **void** averageHeight(){
9. height=168.8;
10. System.out.println("中国人的平均身高(cm):"+height);
11. }
12. **public** **void** averageWeight(){
13. weight=65.0;
14. System.out.println("中国人的平均体重(kg):"+weight);
15. }
16. }

AmericanPeople.java

1. **package** T1;
3. **public** **class** AmericalPeople **extends** People{
4. **public** **void** americanBoxing(){;}
5. **public** **void** speakHello(){
6. System.out.println("how do you do ");
7. }
8. **public** **void** averageHeight(){
9. height=176.0;
10. System.out.println("american's average height(cm):"+height);
11. }
12. **public** **void** averageWeight(){
13. weight=75.0;
14. System.out.println("american's average weight(kg):"+weight);
15. }
16. }

BeijingPeople.java

1. **package** T1;
3. **public** **class** BeiJingPeople **extends** ChinaPeople{
4. **public** **void** BeiJingOpera(){
5. System.out.println("坐如钟,站如松,睡如松\n" +
6. "直拳、勾拳、组合拳\n" +
7. "花脸,青衣,花旦和老生" +
8. "坐如钟,站如松,睡如松");}
9. **public** **void** speakHello(){
10. ;
11. }
12. **public** **void** averageHeight(){
13. height=172.5;
14. System.out.println("北京人的平均身高为(cm):"+height);
15. }
16. **public** **void** averageWeight(){
17. weight=70.0;
18. System.out.println("北京人平均体重为(kg):"+weight);
19. }
21. }

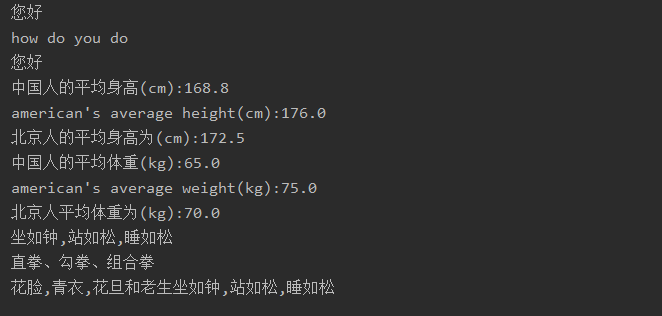
Main.java

1. **package** T1;
3. **public** **class** Main {
4. **public**  **static** **void** main(String[] args){
5. People people=**new** People();
6. ChinaPeople chinaPeople=**new** ChinaPeople();
7. AmericalPeople americalPeople=**new** AmericalPeople();
8. BeiJingPeople beiJingPeople=**new** BeiJingPeople();
10. people.speakHello();
11. americalPeople.speakHello();
12. chinaPeople.speakHello();
13. chinaPeople.averageHeight();
14. americalPeople.averageHeight();
15. beiJingPeople.averageHeight();
16. chinaPeople.averageWeight();
17. americalPeople.averageWeight();
18. beiJingPeople.averageWeight();
19. beiJingPeople.BeiJingOpera();
20. }
22. }

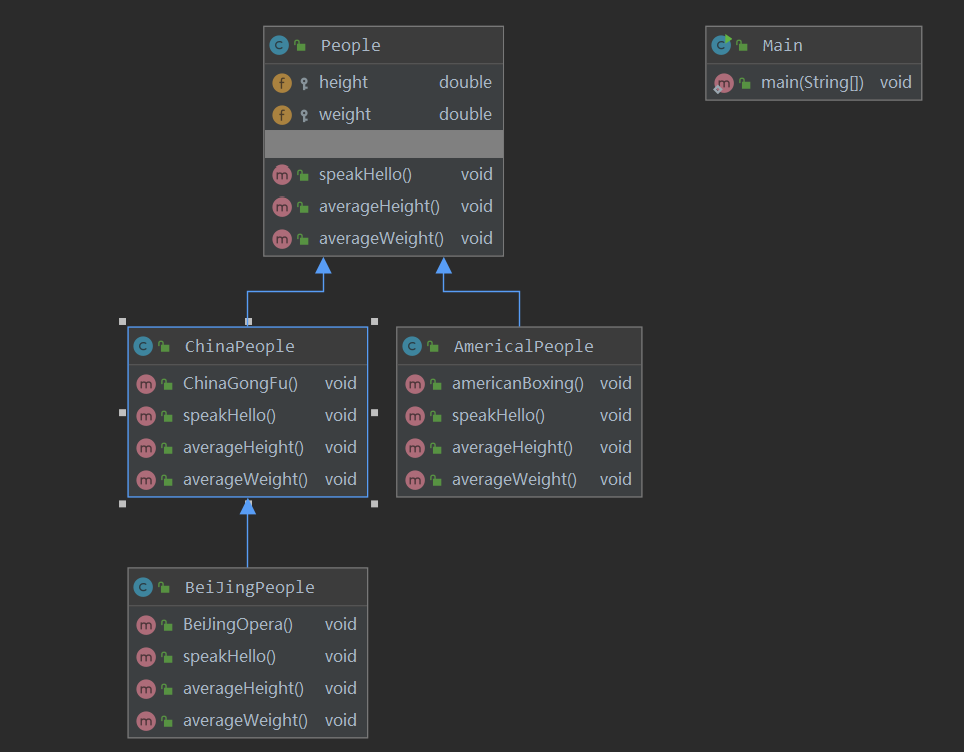
[思路及算法]

答:这题是对类的继承性质的运用和理解，实现好每个类，正确的进行继承是这道题的基础。因此这道题没有什么算法是一道基础题。

[运行结果]



[UML图]



[分析]

这道题难度不大，实现好每个类后，根据输出结果，在主函数中创建不同类的对象进行相应的函数调用即可。实现和思路都很简单。

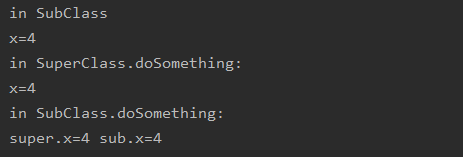
**题二:**

本题是一道程序阅读题。问题及回答如下:

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？

答:删除该两行代码代码，在SubClass子类中只能调用其父类的x,所以程序输出在subclass与superclass中的x是相同的，均为4.

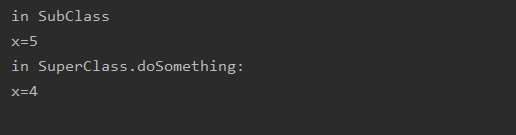
修改后的输入如下：



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。

答:注释后，subC.dosomething()调用的是其从父类继承下来的dosomething函数，所以输出的是x=4;

注释后输出结果:



问题3：将第30行、31行分别改为

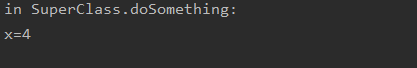
SuperClass superA = new SubClass(); //第30行

superA.doSomething(); //第31行

程序的输出结果有变化吗？为什么？

答:有变化。修改后，创建的是父类的实例对象，所以调用的dosomething不能同时调用其子类的dosomething函数。所以只能输出x=4;

输出结果为:



**题三：**

[源代码]

Animal.java

1. **package** T3;
3. **public** **class** Animal {
4. **private** **int** id;
5. **protected** String race;
6. **public** Animal(){};
7. **public** Animal(**int** id){
8. **this**.id=id;
9. }
10. **public** **void** showMessage(){
11. System.out.print("id="+id+" race="+race+" ");
12. }
14. }
16. **class** birds **extends** Animal **implements** Talent{
17. **private** String talent="fly";
18. @Override
19. **public** **void** showTalent() {
20. System.out.println("talent="+talent);
21. }
22. **public** birds(){};
23. **public** birds(**int** id){
24. **super**(id);
25. race="birds";
26. }
27. }
29. **class** insect **extends** Animal **implements** Talent{
30. **private** String talent="small";
31. **public** **void** showTalent(){
32. System.out.println("talent="+talent);
33. }
34. **public** insect(){};
35. **public** insect(**int** id){
36. **super**(id);
37. race="insect";
38. }
39. }
40. **class** fish **extends** Animal **implements**  Talent{
41. **private** String talent="swim";
42. **public** **void** showTalent(){
43. System.out.println("talent="+talent);
44. }
45. **public** fish(){};
46. **public** fish(**int** id){
47. **super**(id);
48. race="fish";
49. }
50. }

Talent.java

1. **package** T3;
3. **public** **interface** Talent {
4. **void** showTalent();
5. }

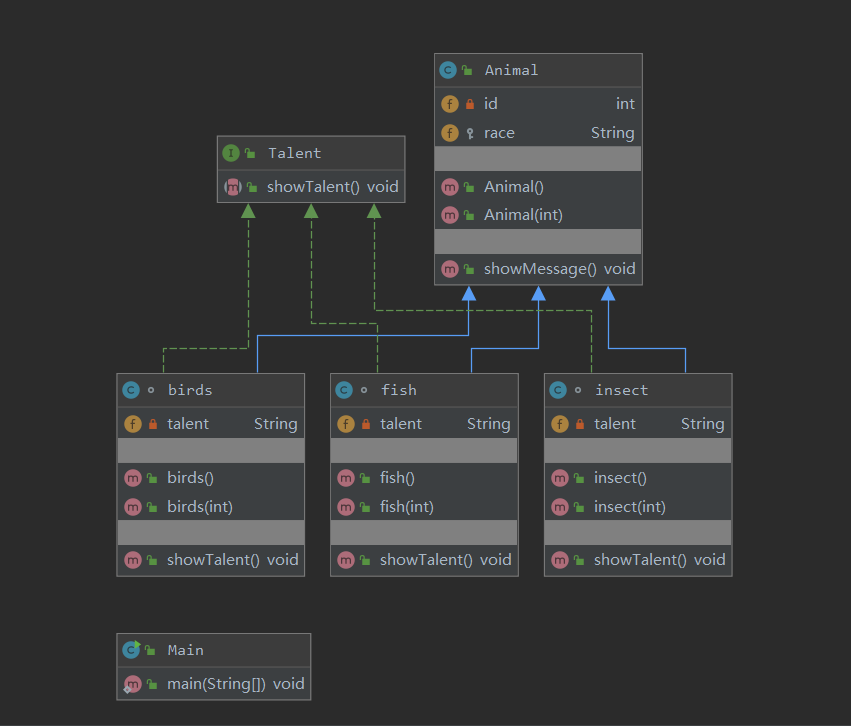
Main.java

1. **package** T3;
3. **import** java.util.Random;
5. **public** **class** Main {
6. **public** **static** **void** main(String[] args){
7. Animal[] animals=**new** Animal[100];
8. **for** (**int** i=0;i<100;i++){
9. Random random=**new** Random();
10. **int** temp=random.nextInt(1000);
11. **if** (temp%3==0) animals[i]=**new** birds(temp);
12. **else** **if**(temp%3==1) animals[i]=**new** insect(temp);
13. **else** animals[i]=**new** fish(temp);
14. }
15. **for** (**int** i=0;i<100;i++){
16. animals[i].showMessage();
17. **if** (animals[i].race.equals("birds")) **new** birds().showTalent();
18. **else** **if**(animals[i].race.equals("insect")) **new** insect().showTalent();
19. **else** **new** fish().showTalent();
20. }
21. }
22. }

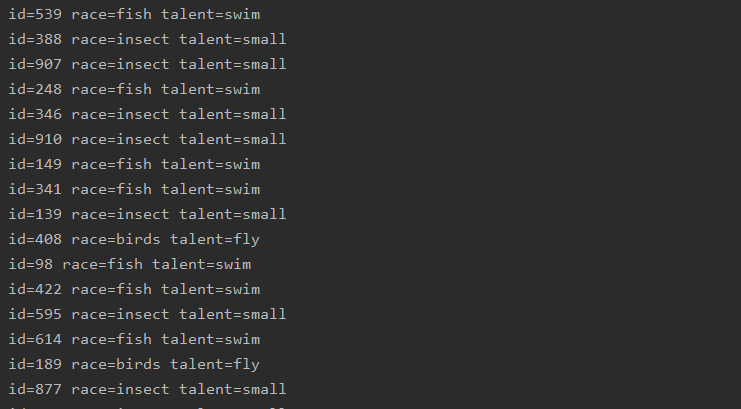
[思路及算法]

定义一各Animal类作为父类，在定义一个接口Talnet,所有动物都继承Animal类并且都要实现这个接口，按照这个思路，就可以一步步的实现代码。在随机生成100个对象的时候，我们可以设置一个Animal数组，再根据随机产生的id号对3进行取余来选择哪种动物即可。思路和实现都较为简单。

[UML图]



[运行结果及截图]



[分析]

这题在交代了思路的情况下实现并不难，一步步的去实现即可得到实验的结果，但前提是对接口的使用和设计要有了解。因此在实现这题的时候并没有遇到什么困难，比较顺利的完成，但对接口的进一步理解和使用还需要练习。

**题四:**

[源代码]

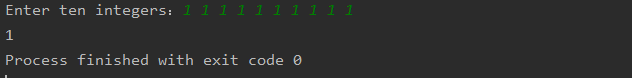
1. **import** java.util.ArrayList;
2. **import** java.util.Scanner;
4. **public** **class** T4 {
5. **public** **static** **void** removeDuplicate(ArrayList<Integer> list){
6. **for** (**int** i=0;i<list.size();i++){
7. **for** (**int** j=i+1;j<list.size();j++){
8. **if** (list.get(i)==list.get(j)) {
9. list.remove(j);
10. j--; //删除后，arraylist里面得数会自动往前移一个位置，所以j不能动。
11. }
12. }
13. }
14. **for** (**int** i=0;i<list.size();i++) System.out.print(list.get(i)+" ");
15. }
16. **public** **static** **void** main(String[] args){
17. ArrayList<Integer> arrayList=**new** ArrayList<Integer>();
18. Scanner input=**new** Scanner(System.in);
19. System.out.print("Enter ten integers：");
20. **for** (**int** i=1;i<=10;i++) arrayList.add(input.nextInt());
21. removeDuplicate(arrayList);
22. }
23. }

[思路及算法设计]

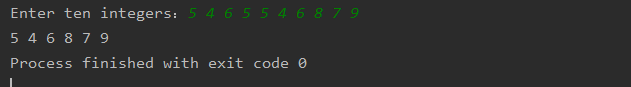
答:这题运用了arrayList数据结构，首先我们创建一个ArrayList对象arraylist，将10个数据通过add函数加入到arraylist中，到这里就实现了数据输入的部分。在去重和输出功能，我设计了一个removeDuplicate函数，通过使用双重循环，将每一个数和其后面所有数进行比较，若相同，则将后面的数通过arraylist.remove()函数删除即可，实现去重的效果。

[运行结果及截图]

全为1的情况：



其他另外情况:



[分析]

答:这题的主要难点就是如何去重。我所使用的是暴力法去重，即通过双重循环来比较了是否有相同的数。这一有一个需要注意的地方: 删除后，arraylist里面的数据会自动往前移一个位置，改变了ayyaylist的长度，所以要特别关注第二个下标，此时第二个下标所表示的数其实就是需要比较的下一个数，不需要在进行加一操作的。只要这里处理好就可以这题就迎刃而解。

因为是简单的顺序结果，且没有判断，所以这里不给出流程图.

**题五：**

[源代码]

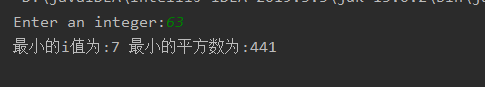
1. **import** java.util.Scanner;
3. **public** **class** T5 {
4. **private** **static** **int**[] array=**new** **int**[10000];
5. **public** **static** **void** whichi(**int** n){
6. **int** m=n;
7. **for** (**int** i=2;i<Math.sqrt(m);i++){
8. **while** (n%i==0){
9. array[i]++;
10. n/=i;
11. }
12. }
13. **int** ans=1;
14. **for** (**int** i=0;i<m;i++){
15. **if** (array[i]%2==1) ans\*=i;
16. }
17. System.out.println("最小的i值为:"+ans+" 最小的平方数为:"+ans\*m);
18. }
19. **public** **static** **void** main(String[] args){
20. Scanner input=**new** Scanner(System.in);
21. System.out.print("Enter an integer:");
22. **int** m=input.nextInt();
23. whichi(m);
24. }
25. }

[思路计算法设计]

答:这题的思路是将输入的数n进行因式分解，记录每个因式的个数，并将个数存入数组。然后遍历数据，检查这个数的每个因式是否为奇数。找出所有个数为奇数的因式，他们相乘的乘积i即为能够使n成为完全平方的最小数。i\*n即为最小的完全平方数。

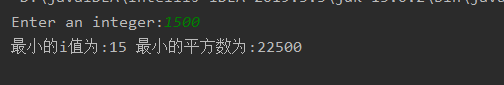
[运行结果及截图]

当输入的数为63.正确结果应为 i=7 i\*n=441 .运行结果如下:



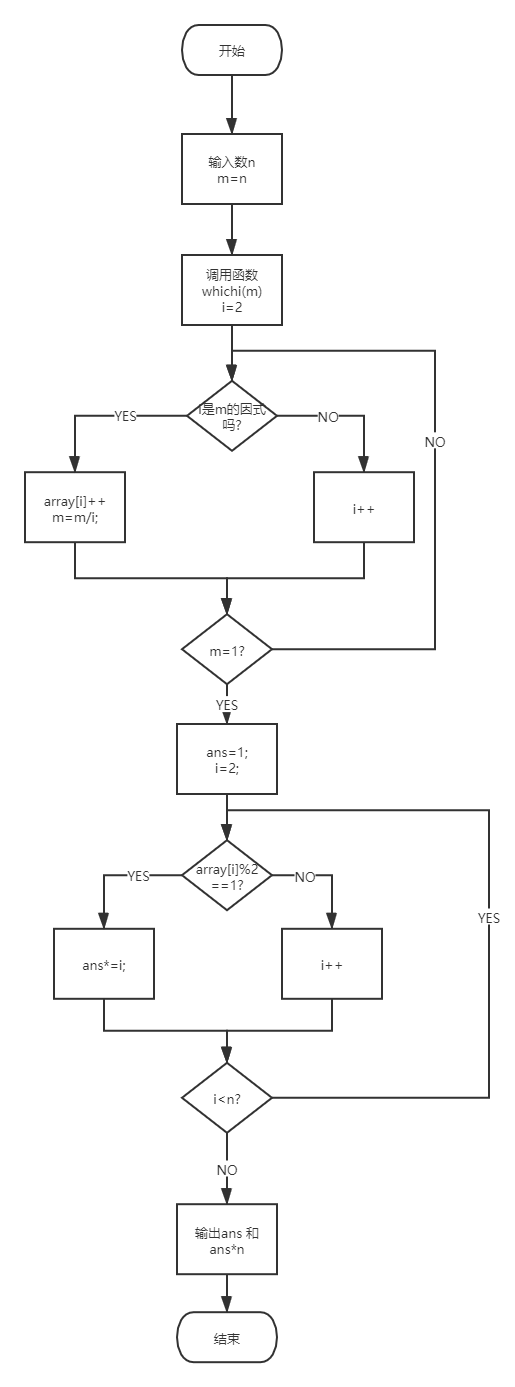
运行正确

当输入的数为1500.正确的结果应为 i=15 i\*n=22500.运行结果如下:



运行结果正确

[流程图]



[分析]

这道题若能想到将数进行因式分解，通过每个因式的个数是奇数还是偶数来判断最小需要乘的数是哪个的话其实还挺容易实现的。虽然在刚写的时候我出现零逻辑错误，通过debug，发现是在求因式的while语句中把传入的形参n给改变了，导致在遍历存入因式个数的数组时出现错误。为了解决这个为题，其实只需要将形参n再赋值给一个数m保留下来即可。总体上，这道题还是挺简单的。

**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。

Super是代表这个类的父类，可以用来调用父类中的普通方法和构造函数。this关键字是对调用对象的引用，可以用来调用引用对象的实例成员。

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:抽象类和接口的继承都需要去重写实现当中的抽象方法。

异:类的继承只能是单继承，和接口可以实现多个接口。

**八、实验总结与体会**

在这次实验中，熟悉了继承，多态和接口的使用，这也是第一次使用到接口的实验。因为几乎每个实验题目都给出了实验步骤，比如先创建什么类，要实现什么功能的，这些都能提供一个清晰地思路，所以这次实验的题目的难度都不大，都可以顺利的完成，在基础知识，概念理解上对我们有很大的帮助。