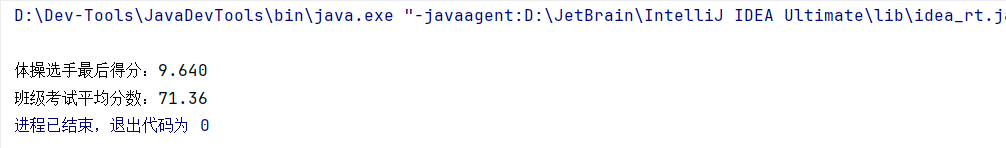
**第七次作业**

1. **教材7.11上机实践**
2. **评价成绩**

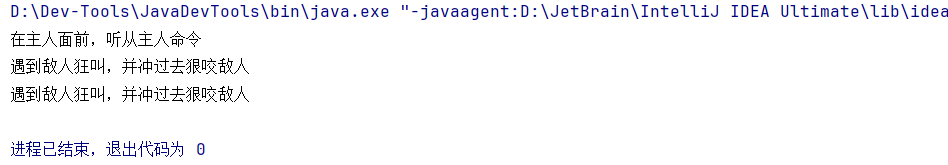
interface ComputerAverage {  
 public double average(double x[]);  
}  
  
class Gymnastics implements ComputerAverage {  
 public double average(double x[]) {  
 int count = x.length;  
 double aver = 0, temp = 0;  
 for (int i = 0; i < count; i++) {  
 for (int j = 0; j < count; j++) {  
 if (x[j] < x[i]) {  
 temp = x[j];  
 x[j] = x[i];  
 x[j] = temp;  
 }  
 }  
 }  
 for (int i = 1; i < count - 1; i++) {  
 aver = aver + x[i];  
 }  
 if (count > 2) {  
 aver = aver / (count - 2);  
 } else {  
 aver = 0;  
 }  
 return aver;  
 }  
}  
  
class School implements ComputerAverage {  
 public double average(double x[]) {  
 int count = x.length;  
 double aver = 0;  
 for (int i = 0; i < count; i++) {  
 aver = aver + x[i];  
 }  
 aver = aver / count;  
 return aver;  
 }  
}

public class Estimator {  
 public static void main(String[] args) {  
 double a[] = {9.88, 9.88, 9.86, 9.12, 9.69, 9.65, 9.97};  
 double b[] = {87, 56, 79, 85, 96, 76, 56, 45, 36, 79, 90};  
 ComputerAverage computer;  
 computer = new Gymnastics();  
 double result = computer.average(a);  
 System.*out*.printf("%n");  
 System.*out*.printf("体操选手最后得分：%5.3f\n",result);  
 computer = new School();  
 result = computer.average(b);  
 System.*out*.printf("班级考试平均分数：%-5.2f",result);  
 }  
}

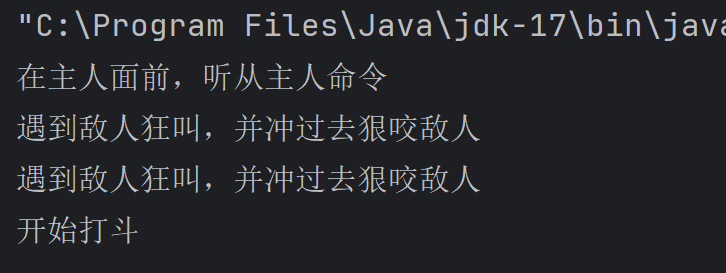


1. **小狗的状态**
2. interface DogState{  
    public void showState();  
   }  
     
   class SoftlyState implements DogState{  
    public void showState(){  
    System.*out*.println("在主人面前，听从主人命令");  
    }  
   }  
     
   class MeetEnemyState implements DogState{  
    public void showState(){  
    System.*out*.println("遇到敌人狂叫，并冲过去狠咬敌人");  
    }  
   }  
     
   class MeetFriend implements DogState{  
    public void showState(){  
    System.*out*.println("遇到朋友晃动尾巴，表示友好");  
    }  
   }  
     
   class Dog{  
    DogState state;  
    public void cry(){  
    state.showState();  
    }  
    public void setState(DogState s){  
    state = s;  
    }  
   }

public class CheckDogState {  
 public static void main(String[] args) {  
 Dog yellowDog = new Dog();  
 yellowDog.setState(new SoftlyState());  
 yellowDog.cry();  
 yellowDog.setState(new MeetEnemyState());  
 yellowDog.cry();;  
 yellowDog.setState(new MeetEnemyState());  
 yellowDog.cry();  
 }  
}



实验后的练习：

1. 
2. **教材p144-145习题**
3. 接口中能声明变量吗？

在 Java 中，接口（Interface）中不能声明实例变量（即类的成员变量），但可以声明常量。接口中的变量默认是 public static final 的，因此它们是常量，而不是实例变量。这些常量必须在接口中进行初始化，因为它们是 final 的，一旦声明就不能再修改。

1. 接口中能定义非抽象方法吗？

在 Java 中，接口中不能定义非抽象方法。接口中的方法默认都是抽象方法，也就是说，只有方法的声明而没有方法的实现。在 Java 8 之后，可以使用默认方法（default methods）和静态方法（static methods）为接口提供具体的实现，但这些方法仍然是抽象方法的一种扩展，只不过具有默认的实现。默认方法和静态方法都必须提供实现，但它们不能被重写。

1. 什么叫接口的回调？

接口的回调是一种常见的编程模式，通常用于事件处理、异步任务处理等场景。在这种模式中，一个对象（通过接口暴露某种功能，然后其他对象可以注册到该接口上，以便在特定事件发生时得到通知或回调。

1. 15

8

1. 18

15