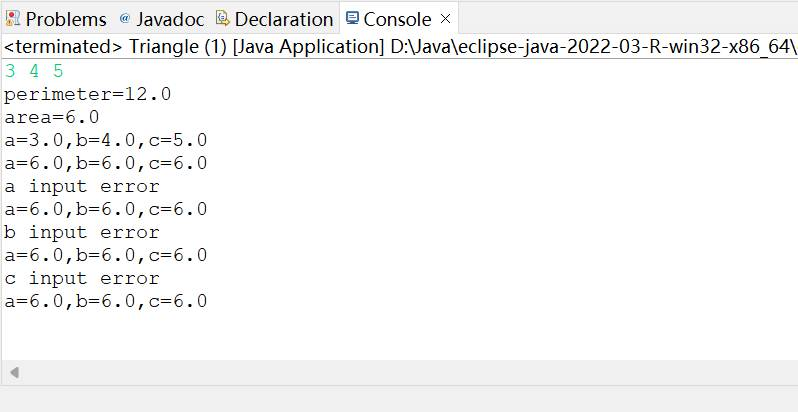
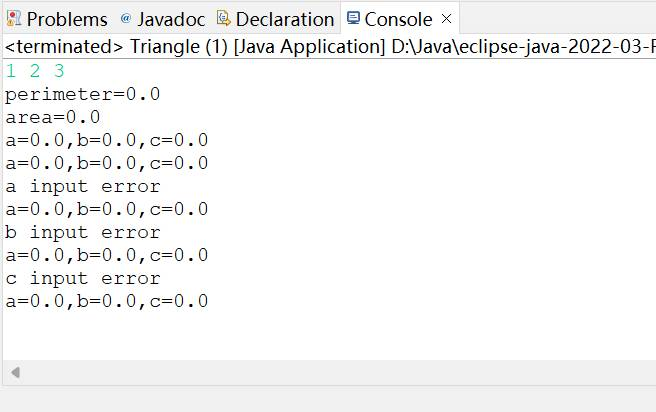
**题目①**

**一、源程序调试过程**  
1、实现了一个构造方法Triangle(double a, double b, double c)，使用setSide(a, b, c)初始化三角形的三边。这个方法确保了对象创建时即对边长的合法性进行验证。  
2、在setA(double a)、setB(double b)和setC(double c)中，我添加了对各边长度设置的同时进行合法性检查的功能。如果新的变成不能形成三角形，将输出错误信息。

3、isValidTriangle(double a, double b, double c)方法用于验证给定的边长是否可以构成三角形。这是通过检查三角形的两边之和大于第三边来实现的。

4、getPerimeter()和getArea()方法计算三角形的周长和面积，使用了海伦公式。  
**二、实验实习结果分析**  
1、通过输入有效和无效的三边长度，验证了三角形对象的响应。对于有效输入“3、4、5”正确地输出了周长和面积。对于无效输入“1、2、3”输出了错误信息。



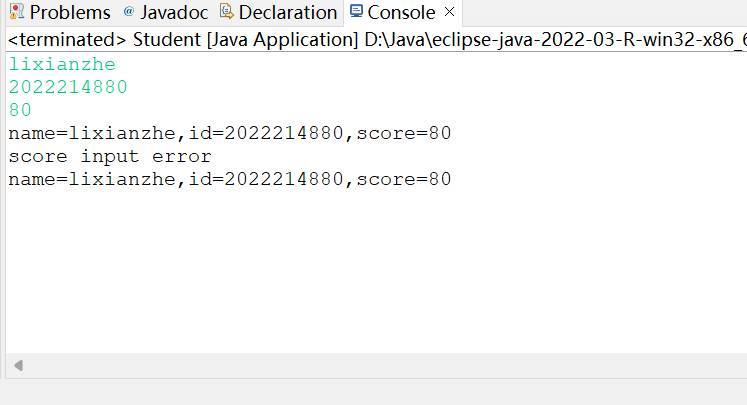


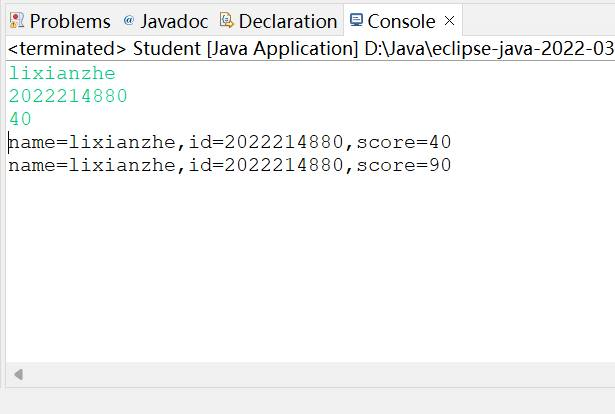
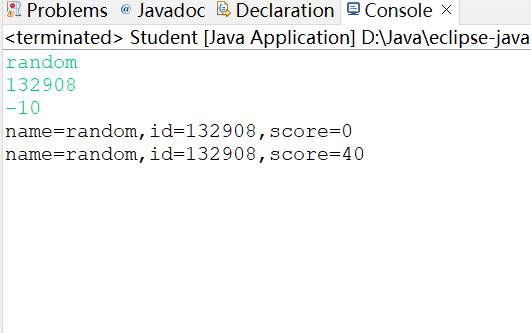
2、错误消息的引入“a input error"有效的指明了设置失败的原因，增强了交互体验。

**三、心得体会**  
此次实验让我深入了解了如何在类中封装数据和方法，以及如何通过方法保证数据的有效性和类的健壮性。通过实现和测试三角形类的方法，我加深了面向对象程序设计的实践理解，尤其是在处理用户错误信息输入验证方面。

**题目②**

**一、源程序调试过程**  
1、在Student(String name, String id, int score)方法中，我确保了即使输入的分数超出正常范围，对象也能被正确创建。如果分数不在0到100之间，则将其设置为0。  
2、在setScore(int score)方法中，添加了对分数合法性的检查。如果分数不在0到100之间，则输出错误信息，并保持分数不变。

3.在toString()方法中，我确保了学生信息的输出格式符合要求，重写了toString方法。  
  
  
**二、实验实习结果分析**  
1、对分数的有效性检查确保了数据的准确性和合法性。即使在输入错误的情况下，程序也能给出明确的反馈，并保持数据状态的正确性。  
2、通过多个测试输入，验证了学生信息输出的一致性和准确性，包括在输入分数非法时的处理情况。  


**三、心得体会**  
通过这次实验，我学到了如何设计和实现一个具有数据验证功能的类，这不仅增强了我的Java编程能力，也让我对如何处理用户输入和保护类内数据的完整性有了更深的理解。通过实际编码测试，我提高了编写可维护和可拓展代码的能力。