****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验四

**姓名：任宇**

**学号：33920212204567**

**学院：信息学院**

**专业：软件工程**

**完成时间：2024-03-23**

1. **实验目的及要求**

* 开始熟悉图形界面
* 熟悉枚举
* 熟悉继承

1. **实验题目及实现过程**
2. 题目一：

设计程序，具备以下功能（要求用到继承）：

* 1. 学生分本科生（学号、姓名、班级）和研究生（学号、姓名、班级、导师）两种；
  2. 课程（编号、课程名、学分）分必修和选修两种；
  3. 创建4个学生信息（2个本科生，2个研究生）
  4. 创建4门课程信息（2门必修，2门选修）
  5. 自动选课部分：为每个学生自动选修所有必修课；
  6. 秘书手动选课部分：为每个同学选修1-2门选修课；
  7. 打印出每个学生的选课信息。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

为了实现题目要求，我设计了Student（学生），Undergraduate（本科生），Postgraduate（研究生），Course（课程），RequiredCourse（必修课），和ElectiveCourse（选修课）这几个类。此外，我还设计了一个CourseSelectionSystem类来模拟选课系统，它包括自动选课和教秘手动选课的方法。具体设计如下：

1. Course（课程基类）

用途：表示一个通用的课程，提供课程的基本信息。

主要成员变量：

* courseId（String）：课程的唯一标识符。
* courseName（String）：课程名称。
* credits（int）：课程学分。
* courseType（CourseType，枚举）：课程类型（必修或选修）。

主要方法：

* 构造方法：初始化课程信息。
* toString()：返回课程的字符串表示，用于打印课程信息。

1. RequiredCourse 和 ElectiveCourse（Course的子类）

用途：分别表示必修课和选修课，继承自Course类。

主要成员变量：继承自Course类，没有额外成员。

主要方法：使用基类的构造方法，并在构造时指定courseType为相应的枚举值。

1. Student（学生基类）

用途：表示一个通用的学生，提供学生的基本信息和选课功能。

主要成员变量：

* + studentId（String）：学生的唯一标识符。
  + name（String）：学生姓名。
  + className（String）：学生所在班级。
  + courses（List<Course>）：学生所选的课程列表。

主要方法：

* + 构造方法：初始化学生信息。
  + addCourse(Course course)：向学生的课程列表中添加一个课程。
  + printCourses()：打印学生的选课信息。

1. Undergraduate 和 Postgraduate（Student的子类）

用途：分别表示本科生和研究生，继承自Student类。

主要成员变量：Postgraduate类包含额外的supervisor（String）成员，表示研究生的导师。

主要方法：使用基类的方法，Postgraduate类的构造方法还需初始化导师。

1. CourseSelectionSystem（模拟选课系统）

用途：管理学生、课程及选课流程的系统。

主要成员变量：

* students（List<Student>）：系统中的学生列表。
* courses（List<Course>）：系统中的课程列表。

主要方法：

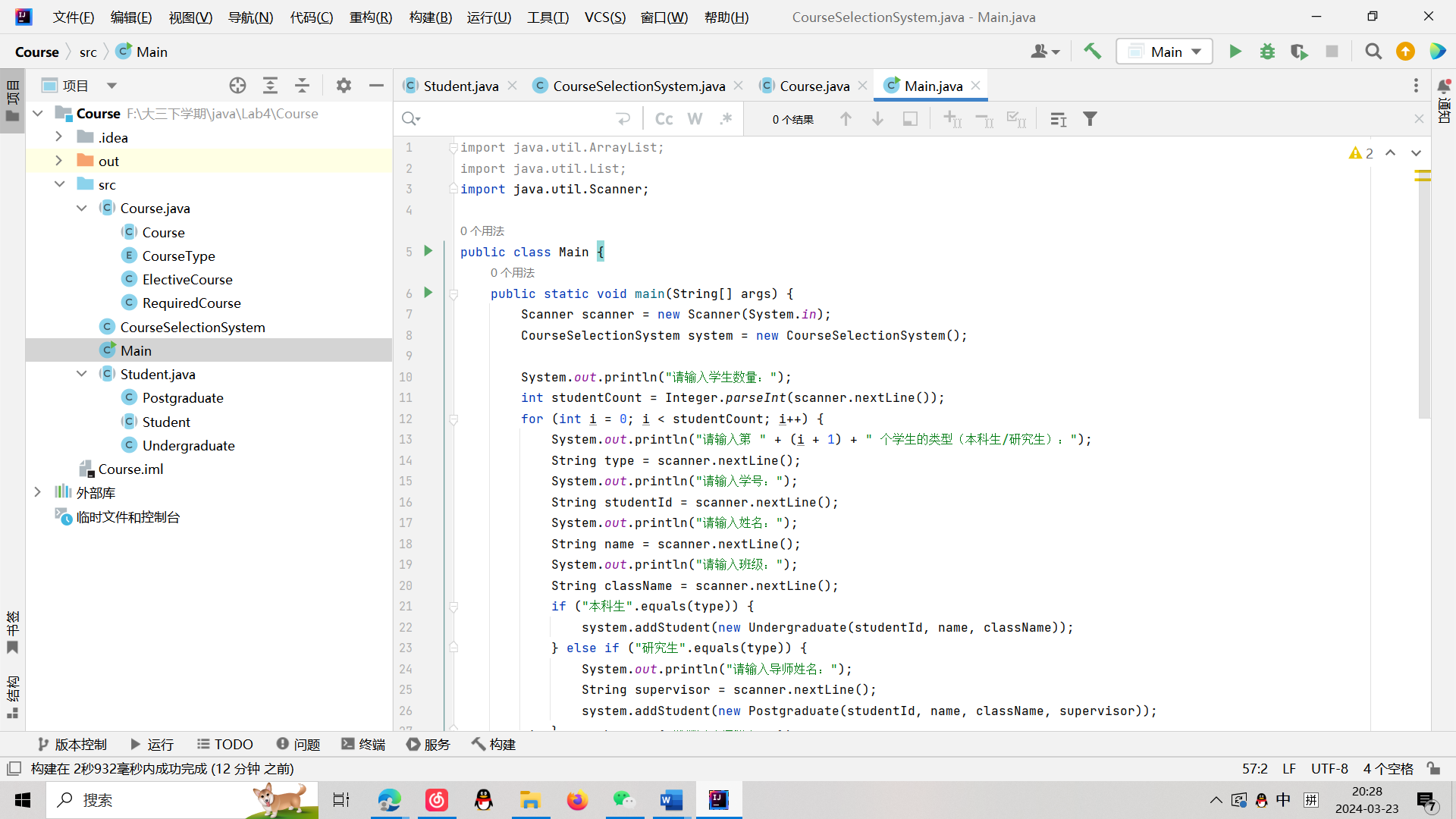
* addStudent(Student student)：向系统中添加一个学生。
* addCourse(Course course)：向系统中添加一个课程。
* autoSelectRequiredCourses()：为所有学生自动选修所有必修课。
* SelectElectiveCourses()：教秘为学生手动选修1-2门选修课。
* printAllStudentCourses()：打印所有学生的选课信息。

1. Main（主类）

调用关系：CourseSelectionSystem是整个程序的核心，负责创建和管理Student对象和Course对象的集合。当程序运行时，首先通过CourseSelectionSystem创建并添加课程（RequiredCourse和ElectiveCourse）和学生（Undergraduate和Postgraduate）。接着自动选课，autoSelectRequiredCourses方法确保所有学生自动选修所有必修课。该过程遍历所有学生和课程，自动为学生添加必修课程到他们的课程列表中。自动选择必修课后，程序通过SelectElectiveCourses方法，根据用户从控制台输入的信息，为学生手动添加选修课程。这一步骤允许秘书为每位学生选择1至2门选修课。最后，系统可以通过printAllStudentCourses方法在控制台上打印每个学生的所有选课信息，展示每位学生所选的课程详情。

1. 过程截图

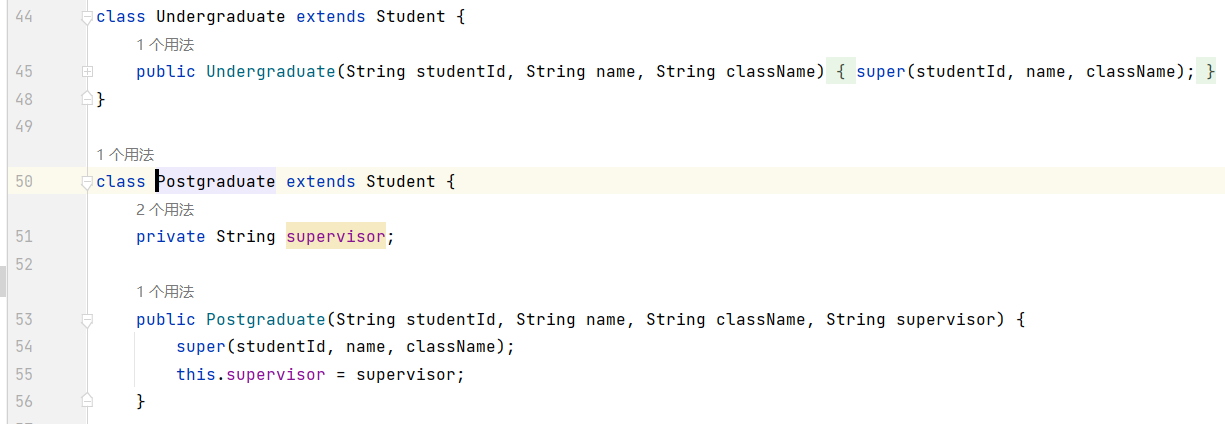
全屏截图：



Student类主要代码：



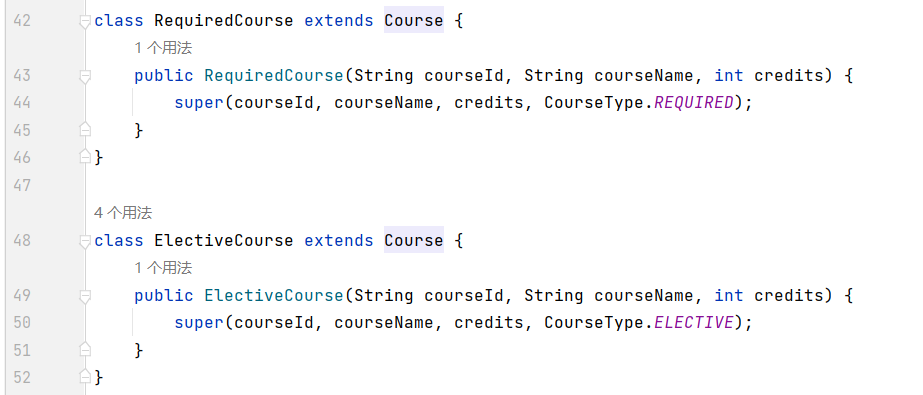
Undergraduate和Postgraduate类实现代码：



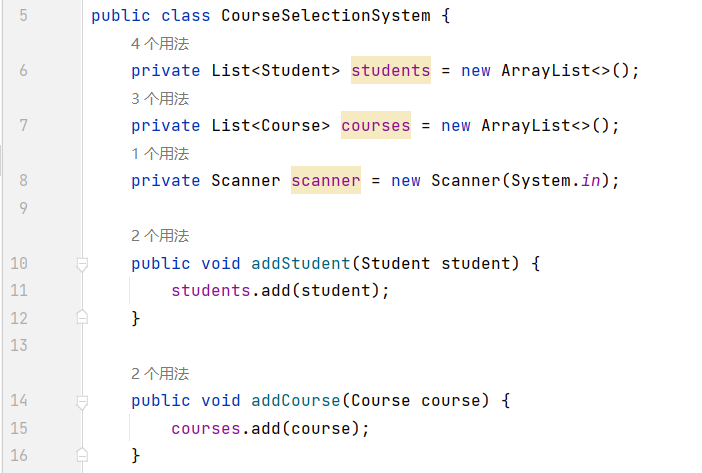
Course类主要代码：

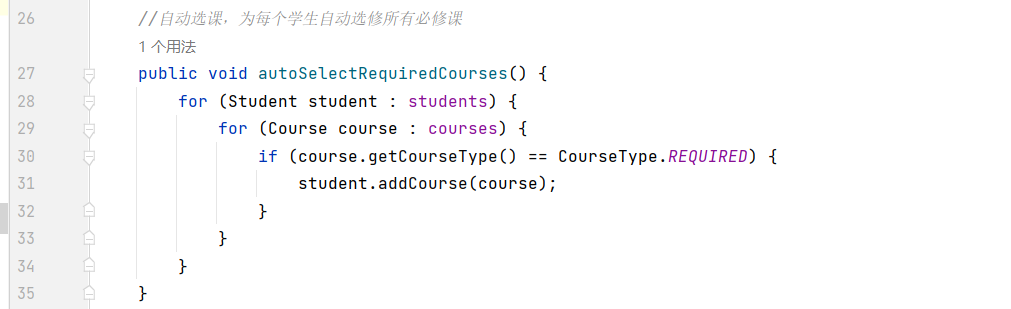


RequiredCourse 和 ElectiveCourse主要代码：

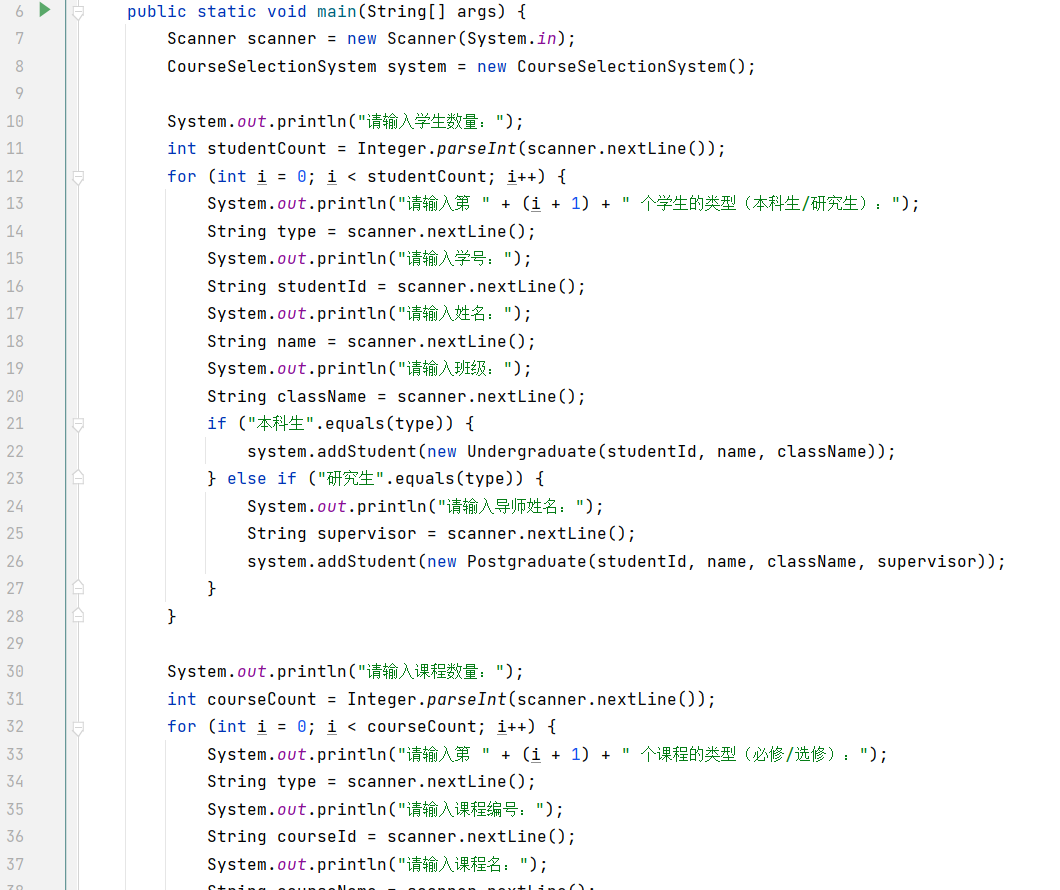


CourseSelectionSystem主要代码：



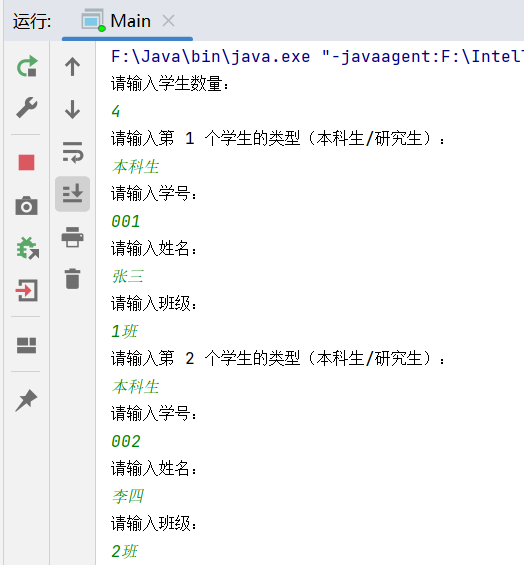


Main类主要代码：

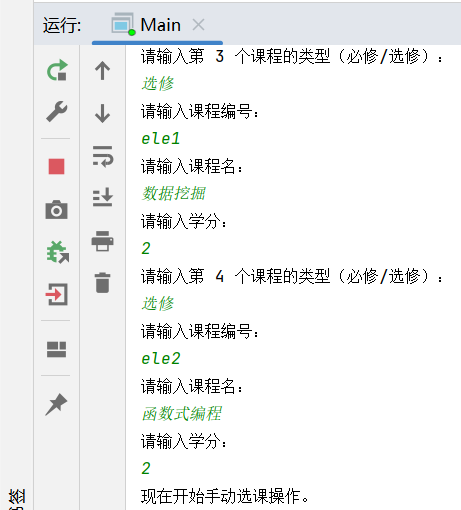


Main类运行情况：

* 创建4个学生信息（2个本科生，2个研究生）

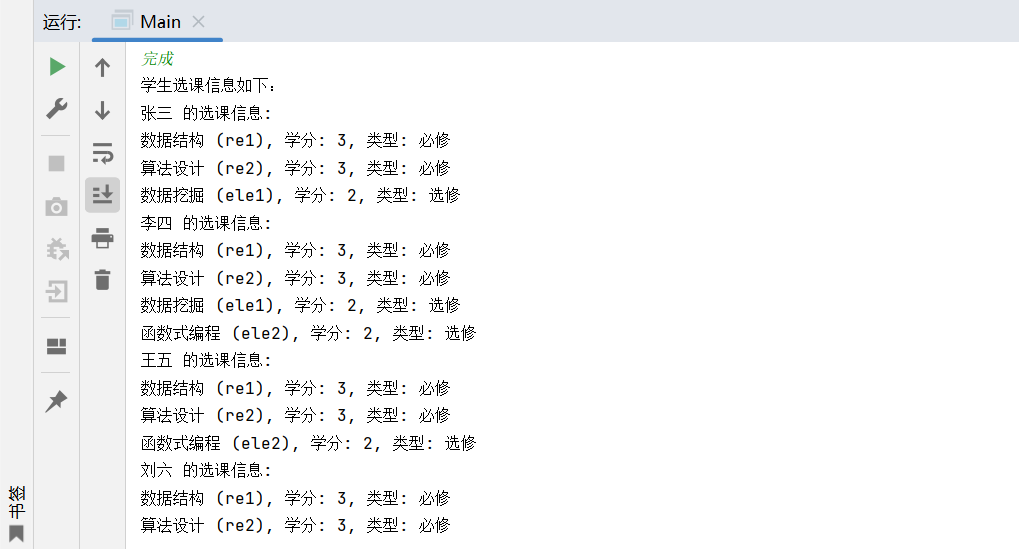
* 创建4门课程信息（2门必修，2门选修）

* 秘书手动选课：为每个同学选修1-2门选修课

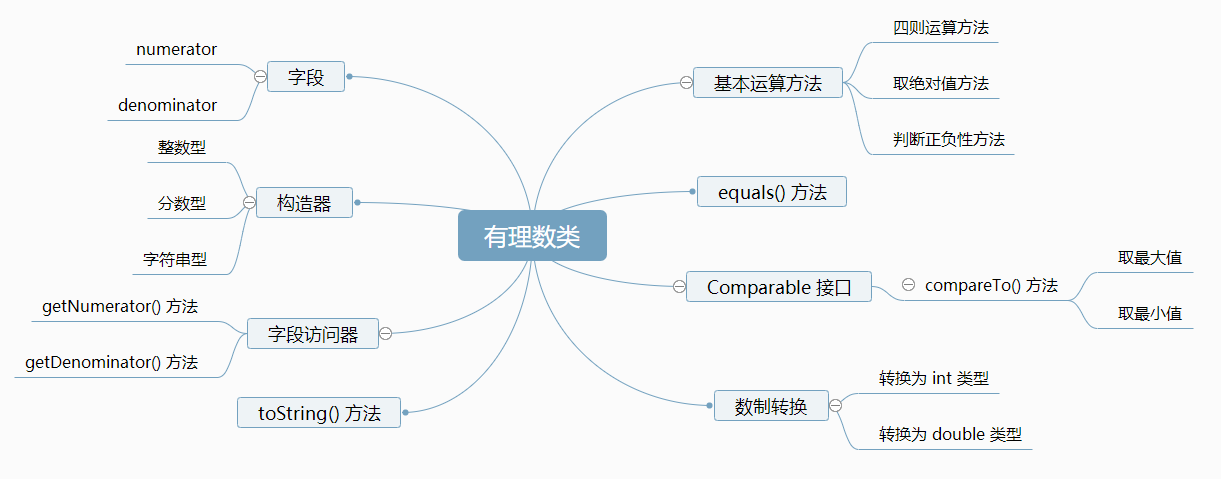
 

* 打印出每个学生的选课信息



1. 题目二：

写一个有理数类（Rational）相关属性和方法要求如下图：



写一个测试类，创建两个有理数对象，输出两个有理数的加、减、乘、除结果。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

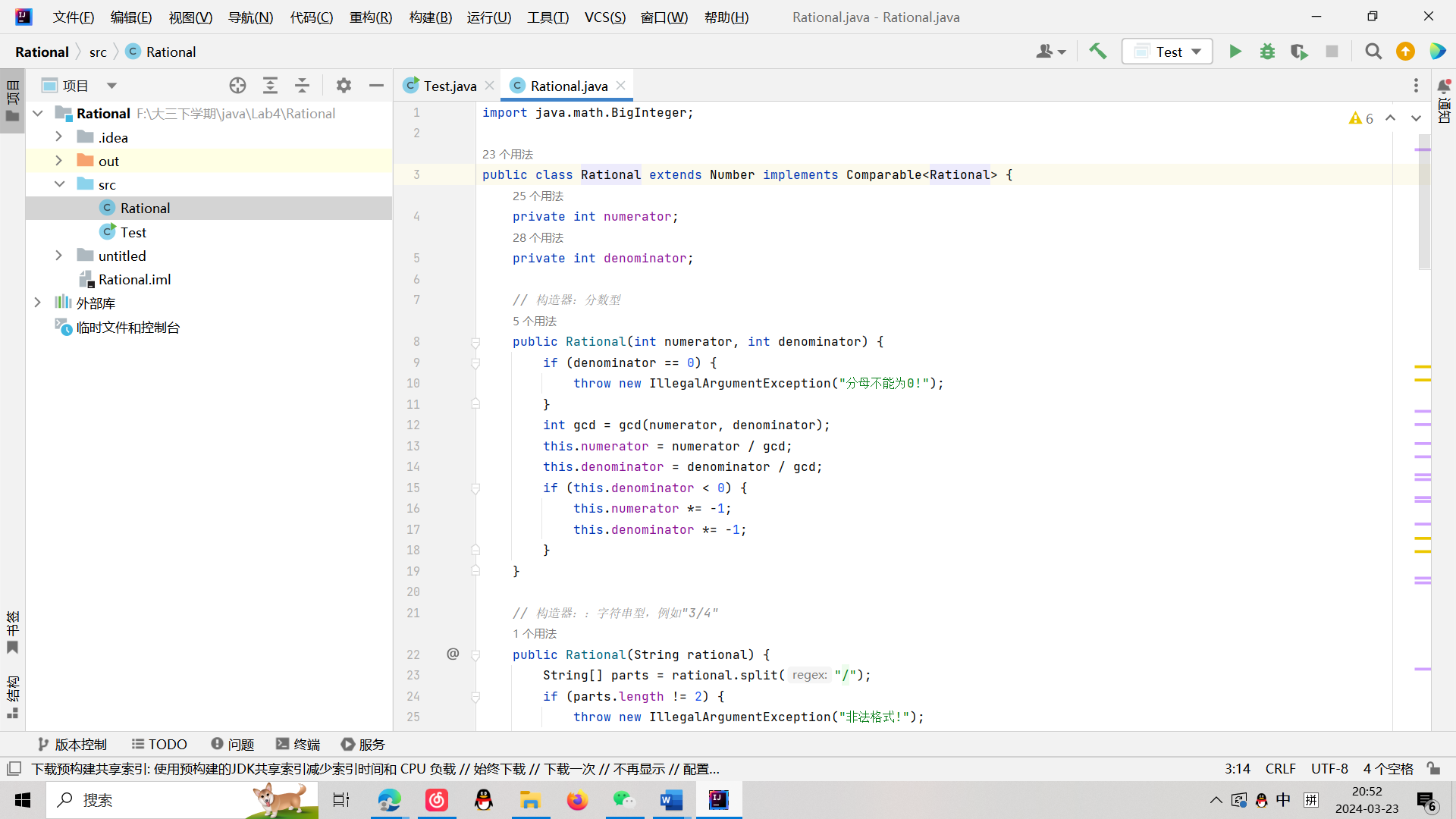
实验中实现了Rational类和Test类，其设计用于表示和操作有理数，有理数是由两个整数的比（分子和分母）所构成的数，重点分析Rational类：

* 主要成员变量
* numerator（分子）：表示有理数的分子部分，为整数类型。
* denominator（分母）：表示有理数的分母部分，为整数类型。分母不能为零。
* 构造方法
* 接受两个整数参数：分别代表分子和分母，用于创建有理数实例。
* 接受一个字符串参数：字符串格式为"分子/分母"，用于从字符串表示创建有理数实例。
* 接受一个浮点数参数：将浮点数转换为其有理数表示形式。
* 主要方法
* getNumerator()和getDenominator()：获取有理数的分子和分母。
* toString()：返回有理数的字符串表示，形式为"分子/分母"。
* 基本运算方法：包括加（add）、减（subtract）、乘（multiply）、除（divide）等操作。
* abs()：取绝对值。
* isPositive()和isNegative()：判断有理数的正负性。
* equals(Object obj)：判断两个有理数是否相等。重写自Object类。
* 实现Comparable<Rational>接口：允许有理数之间进行比较，主要用于排序。
* 数制转换方法：包括转换为整型（toInt）和双精度浮点型（toDouble）。
* 调用关系

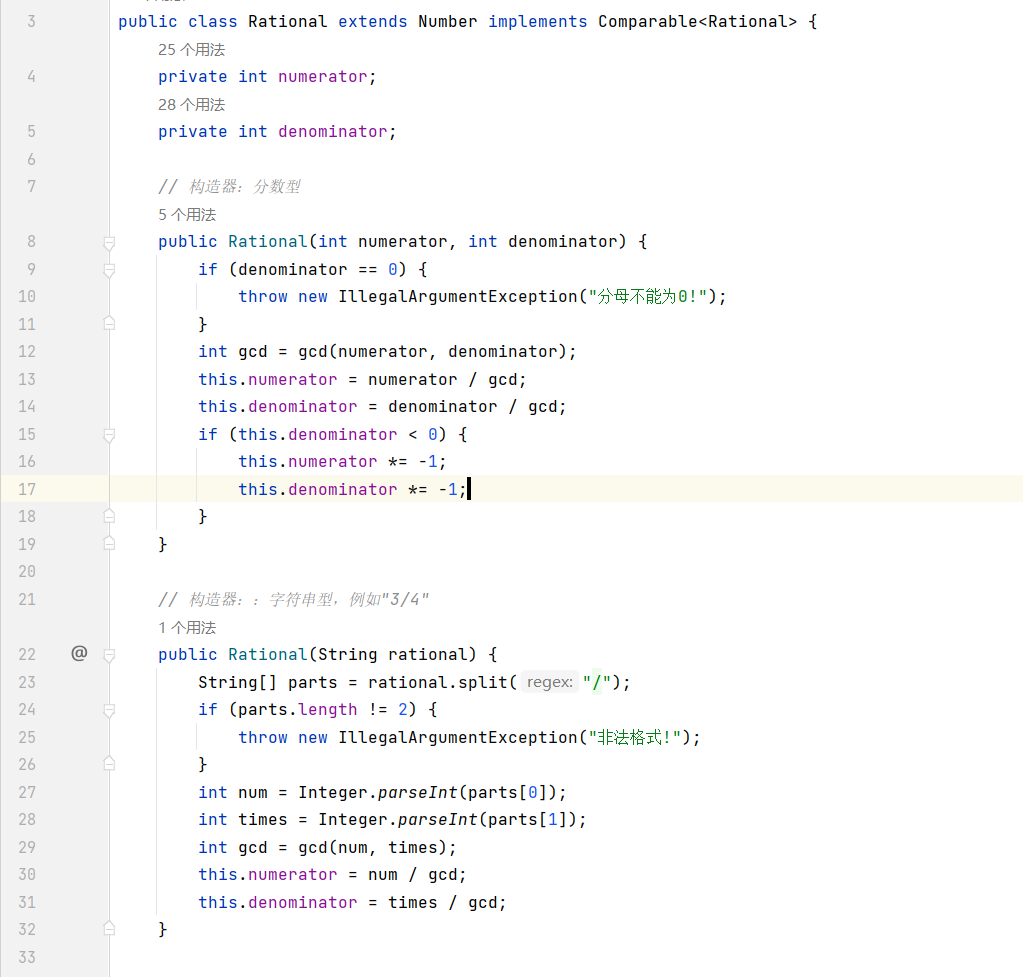
Rational类的对象可以独立存在，其方法主要涉及到与其他Rational对象的交互。运算方法创建并返回新的Rational实例，这些实例是对原有理数进行运算后的结果。equals方法和compareTo方法用于比较两个Rational对象，支持有理数的等值判断和排序。

1. 过程截图

全屏截图：



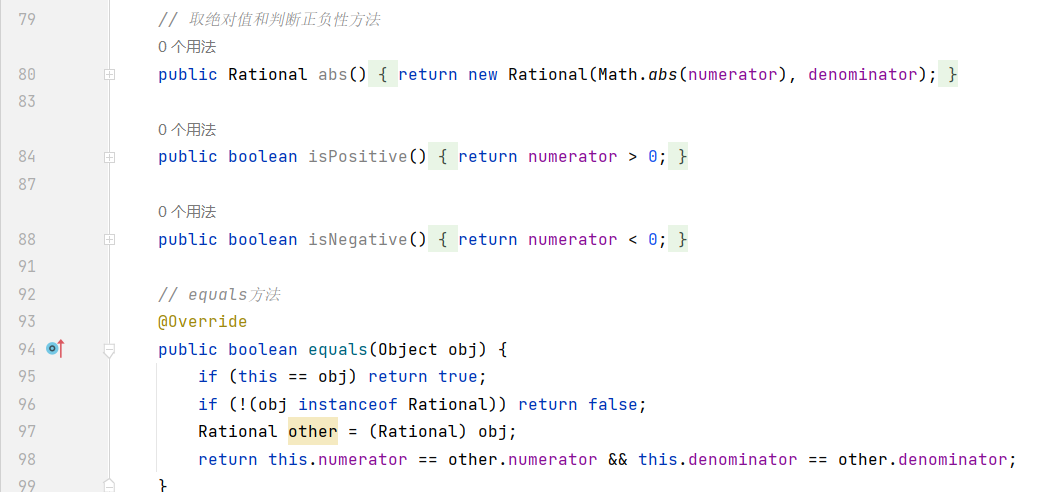
Rational类主要代码：



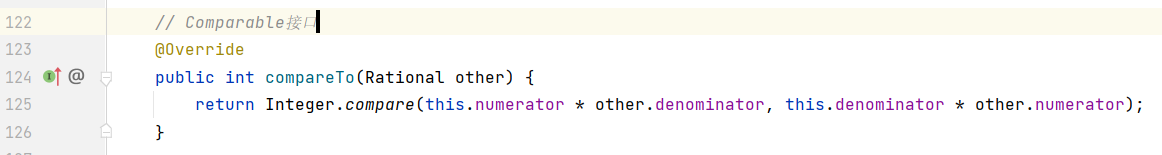
四则运算：



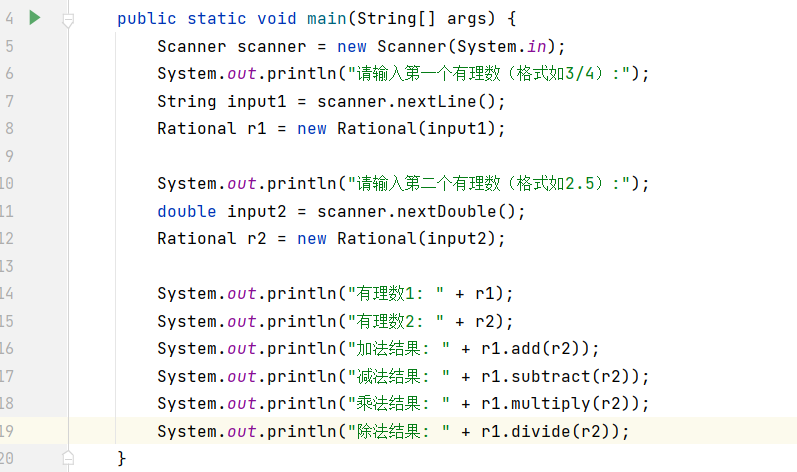
判断正负性和相等：



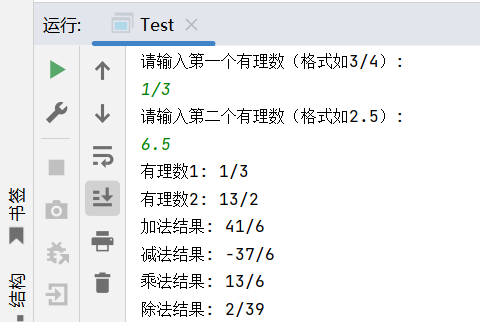
实现Number类的Comparable接口



Test类主要代码：



Test类运行情况：



1. 题目三：

实现一个基础图形类Graph，然后实现三角形类Triangle和矩形类Rectangle，继承自Graph。根据输入的边数实现不同的对象，并计算面积。

输入格式：

一行，一个整数n，表示图形个数。

n行，每行是用空格隔开的整数。

输出格式：

n行，每行是一个图形的面积。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

1. 实现过程

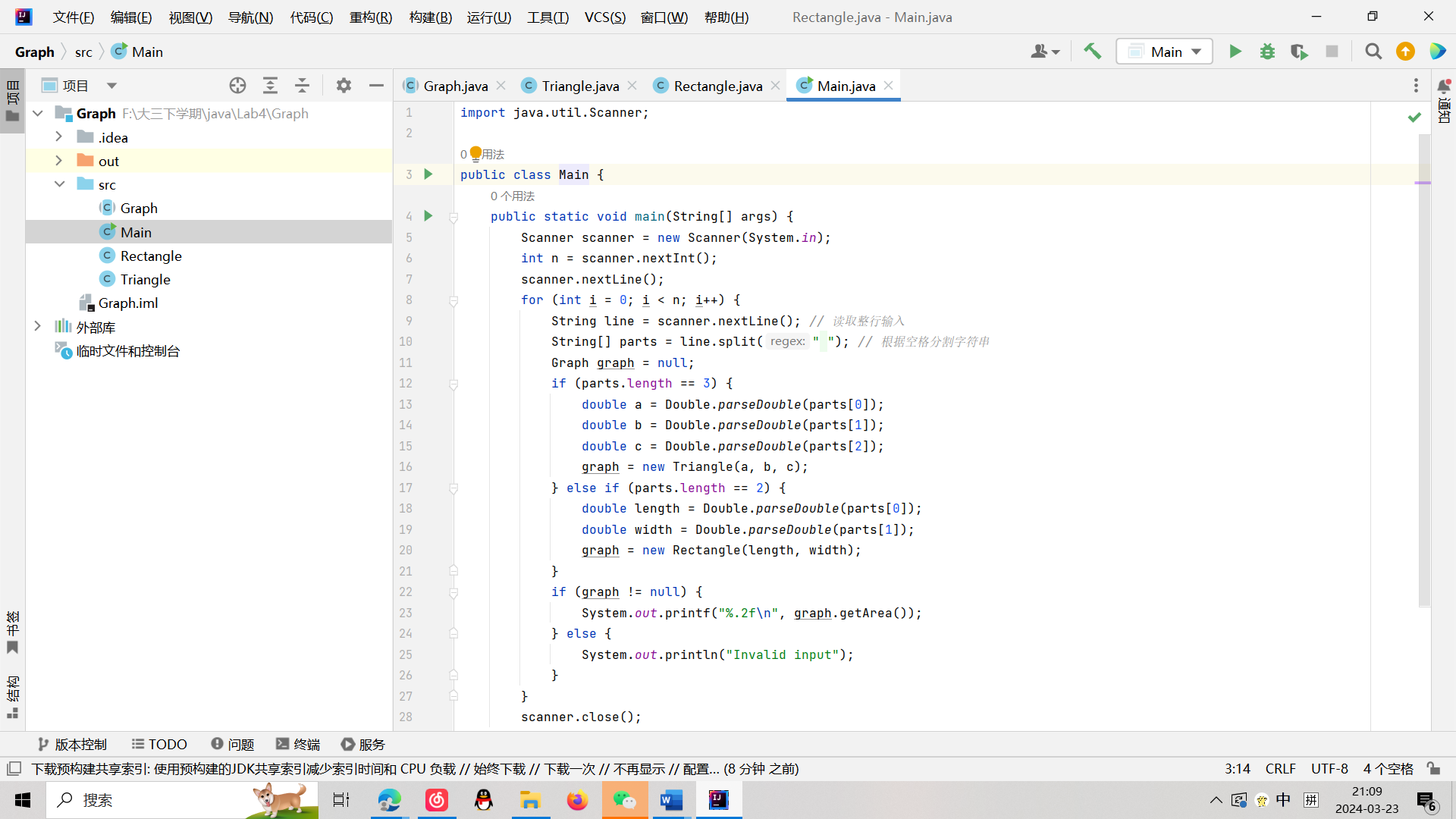
本题具体设计如下：

* Graph（抽象基类）
* 用途：提供一个抽象方法getArea()，用于计算图形的面积。
* getArea()：抽象方法，要求继承Graph类的子类必须实现这个方法以返回图形的面积。
* Rectangle（Graph的子类）
* 用途：表示矩形图形，继承自Graph类。
* 主要成员：length（double）表示矩形的长度以及width（double）表示矩形的宽度。
* 构造方法：接受长度和宽度作为参数并初始化。
* getArea()：重写Graph类的方法，计算并返回矩形的面积。
* Triangle（Graph的子类）
* 用途：表示三角形图形，继承自Graph类。
* 主要成员：a、b、c（double）表示三角形的三边长。
* 构造方法：接受三边长作为参数并初始化。
* getArea()：重写Graph类的方法，使用海伦公式计算并返回三角形的面积。
* Main类
* main(String[] args)：主方法，负责处理输入，创建对应的图形对象，并计算打印这些图形的面积。
* 运行逻辑和调用关系

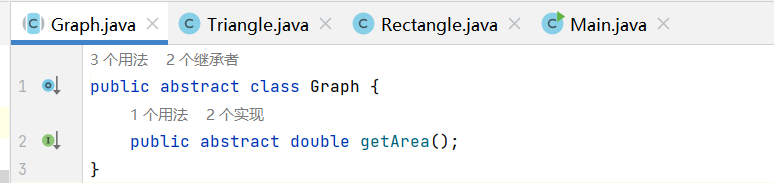
程序启动后，Main类的main方法首先读取输入，根据输入的参数数量决定创建Triangle或Rectangle对象。对于每个输入行，根据参数数量判断图形类型。如果数量为2，认为是矩形；如果数量为3，认为是三角形。对每个图形对象调用getArea()方法计算其面积，并打印出来。

1. 过程截图

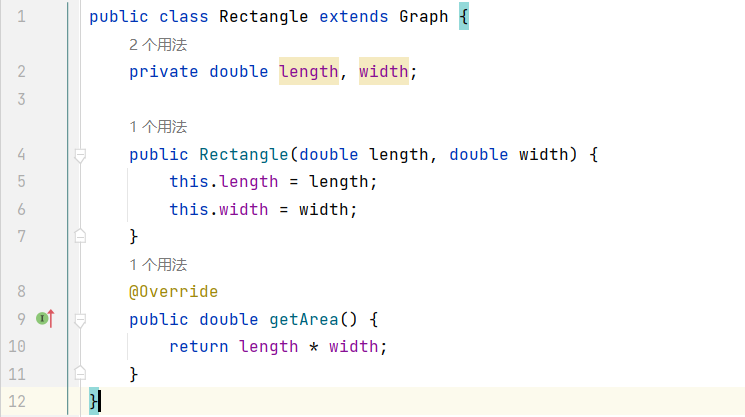
全屏截图：



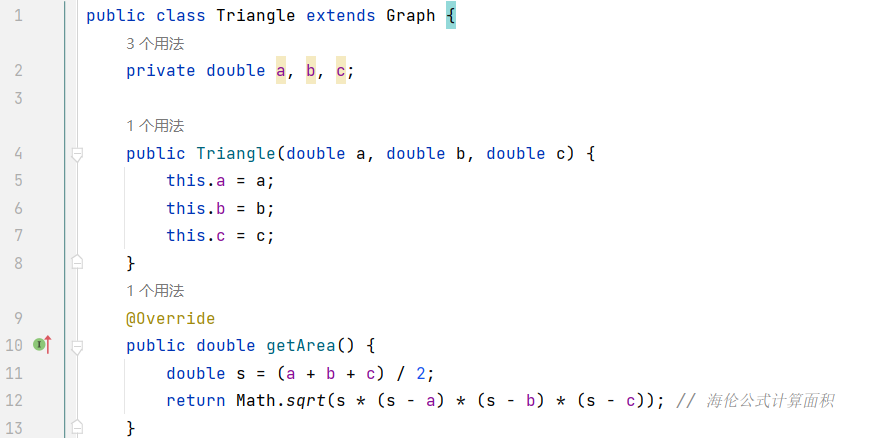
Graph类主要代码：



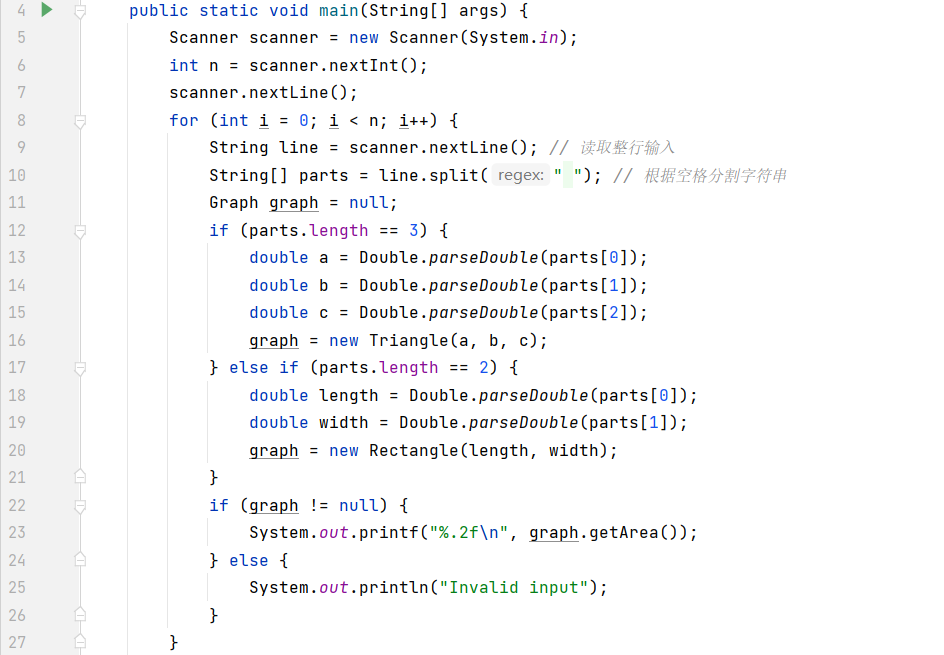
Rectangle主要代码：



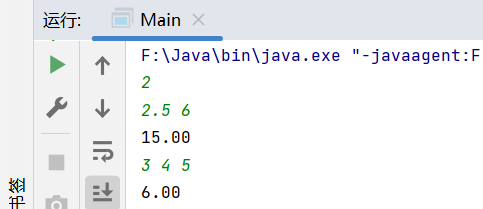
Triangle主要代码：



Main主要代码：



Main执行情况：



1. 题目四：

创建一个简单的JavaFX绘图程序，要求如下：

* 随机产生一个随机数（0,1,2），三个随机数分别对应直线、矩形和椭圆三种图形。根据随机数对应图形，提示用户输入图形所需初始化参数，提示信息应包括参数的范围，用户输入后进行范围检查，若合法，则根据用户输入的信息在界面上绘制出相应的图形。
* 绘制20个图形后，不再创建新的图形。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17，JavaFx22

1. 实现过程

本题具体设计如下：

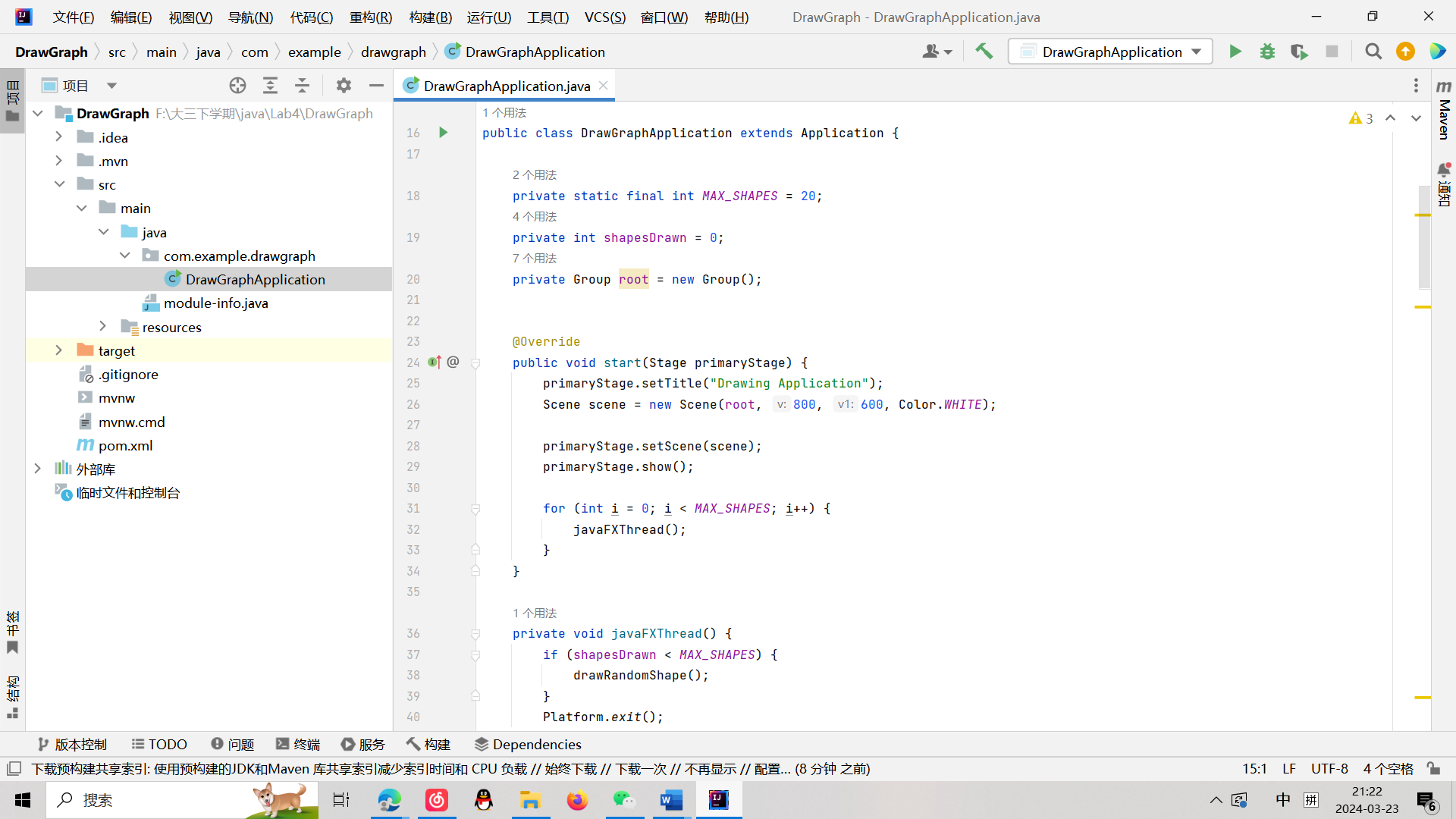
* DrawGraphApplication类

继承：继承自Application类，是JavaFX程序的入口点。

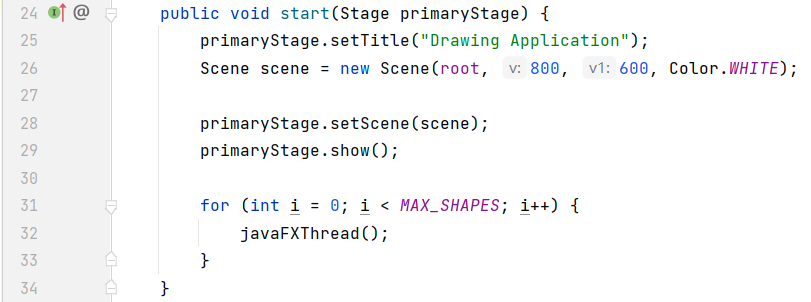
* 主要成员：
* MAX\_SHAPES：定义最大绘制图形数量。
* shapesDrawn：追踪已绘制图形的数量。
* root（Group对象）：用作绘制图形的容器。
* 主要方法：
* start(Stage primaryStage)：覆盖自Application，设置舞台（Stage）和场景（Scene），并开始绘制图形流程。
* javaFXThread()：控制图形绘制的流程，确保不超过最大图形数量。
* drawRandomShape()：随机选择图形类型并调用相应的绘制方法。
* drawLine(), drawRectangle(), drawEllipse()：具体绘制线条、矩形和椭圆的方法，使用TextInputDialog接收用户输入，并将绘制的图形添加到root容器。
* showAlert(String title, String message)：显示错误提示信息的辅助方法。
* JavaFX组件和控件
* Scene和Group：Scene表示程序的主画面，Group作为根节点（root）容纳其他图形节点。所有绘制的图形都会添加到这个Group中。
* Line、Rectangle、Ellipse：JavaFX提供的图形类，分别用于绘制线条、矩形和椭圆。
* TextInputDialog：用于从用户那里接收图形参数的输入对话框。
* Alert：用于显示错误信息的对话框。
* 调用关系
* start方法是程序的起点，它初始化舞台和场景，并开始绘制图形。
* javaFXThread方法控制图形绘制的主循环，直至达到最大图形数量。
* drawRandomShape方法根据随机数选择图形类型，并调用对应的绘制方法（drawLine, drawRectangle, drawEllipse）。
* 绘制方法通过显示TextInputDialog来接收用户输入，根据输入创建并添加图形到场景中。

1. 过程截图

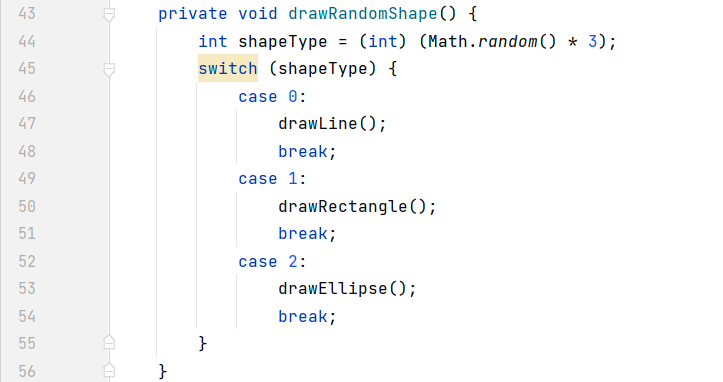
全屏截图：



主要方法实现：



随机绘制图形：



Drawline方法，其他方法类似：



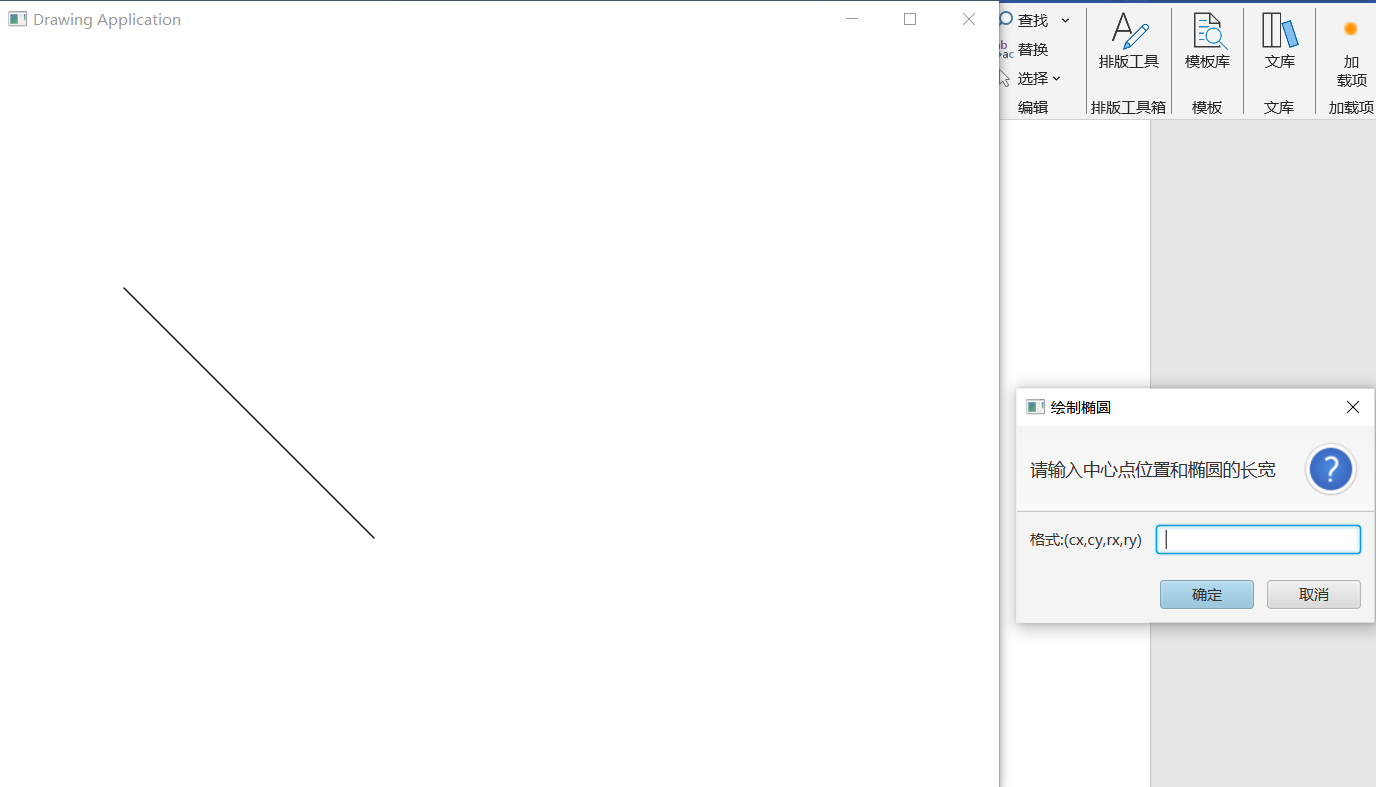
运行截图：

绘制椭圆：



绘制线条：

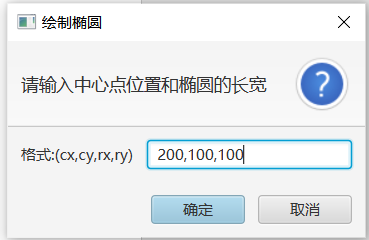


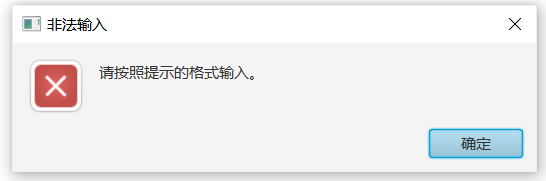


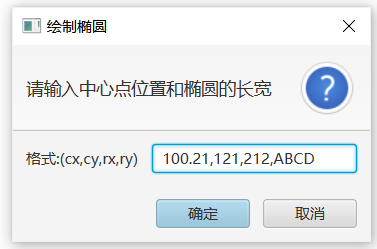
绘制矩形：

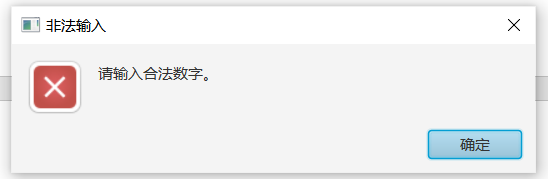


错误情况：









绘制20个图形后程序自动退出。

1. 题目五：

写一个交通信号灯枚举类TrafficLight，并在Test类中输出每种灯颜色的RGB值。

1. 实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK版本：JDK17

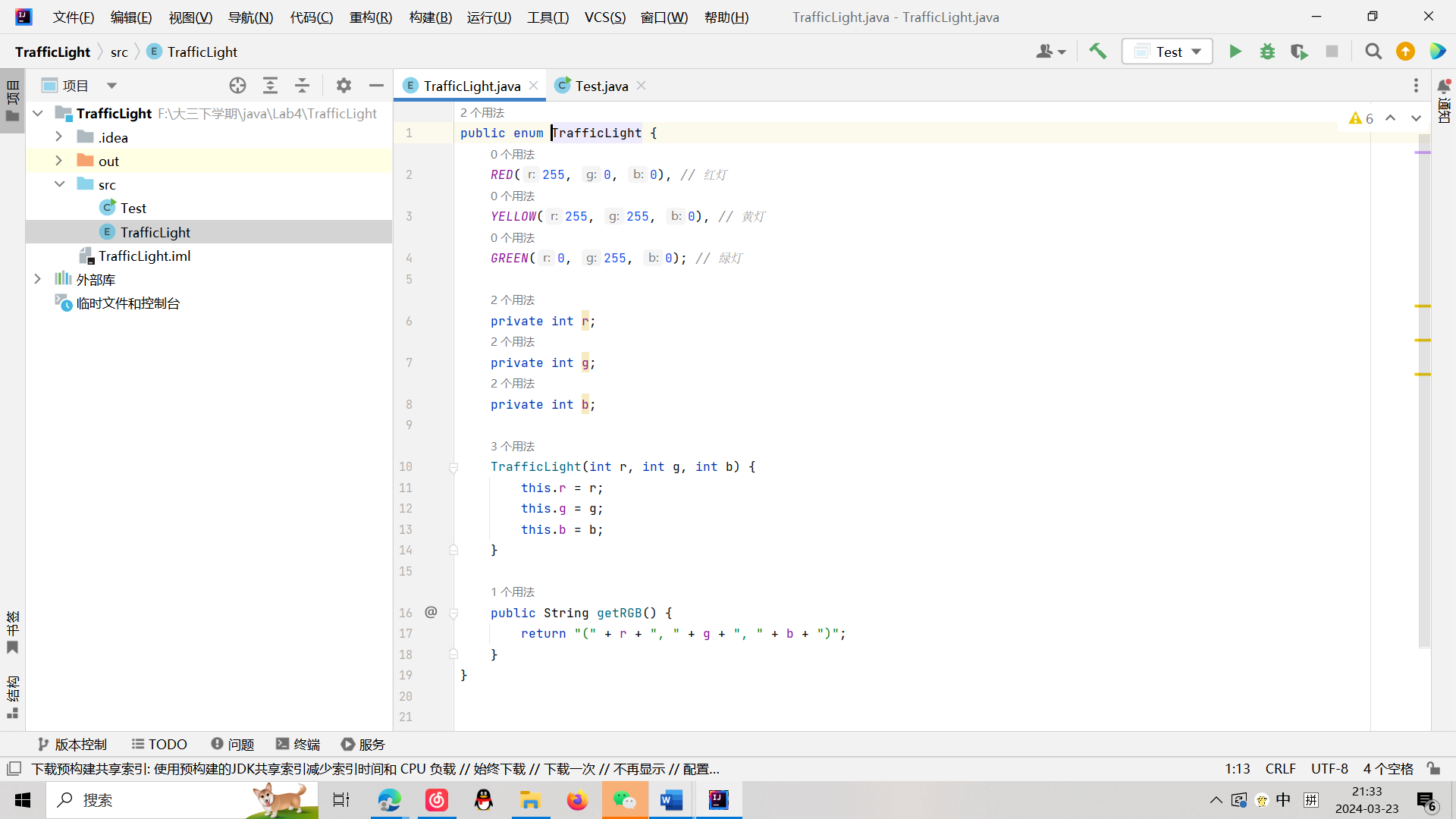
1. 实现过程

程序包含两个主要部分：TrafficLight枚举和Test类，具体如下：

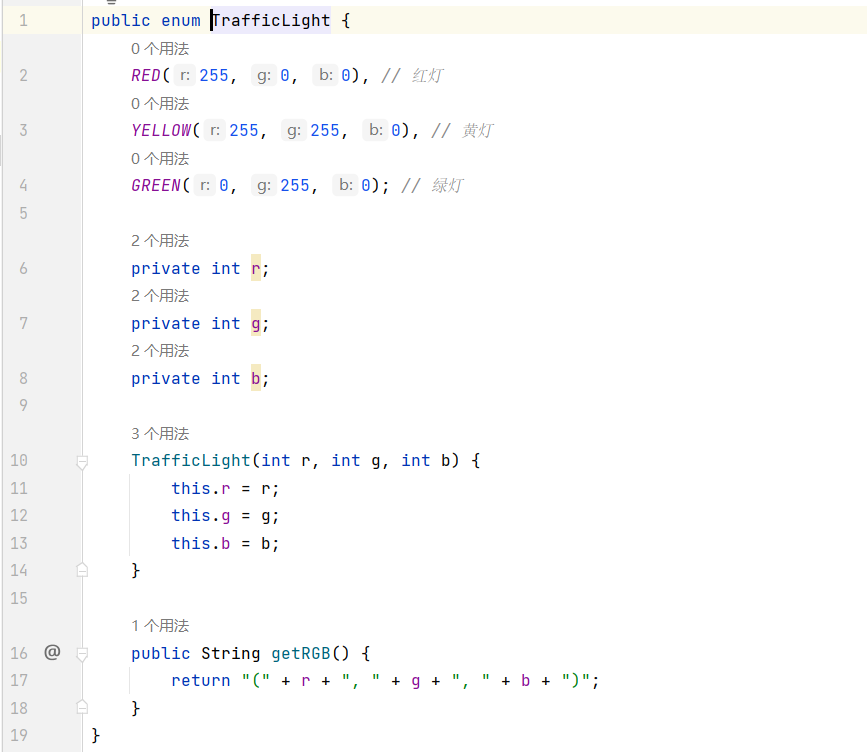
* TrafficLight枚举
* 用途：定义了交通信号灯的三种颜色（红、黄、绿）以及它们对应的RGB颜色值。
* r、g、b：代表颜色的红（R）、绿（G）、蓝（B）分量。
* TrafficLight(int r, int g, int b)：用于初始化每个枚举实例的RGB值。
* getRGB()：返回表示颜色RGB值的字符串，格式为"(r, g, b)"。
* Test类
* 用途：包含了main方法，用于遍历TrafficLight枚举的实例，并打印出每个实例的名称和对应的RGB值。
* 在main方法中，使用for循环遍历TrafficLight枚举的所有实例。对于每个实例，将实例的名称（通过toString()方法隐式获取）和通过调用getRGB()方法获取的RGB值打印到控制台。

1. 过程截图

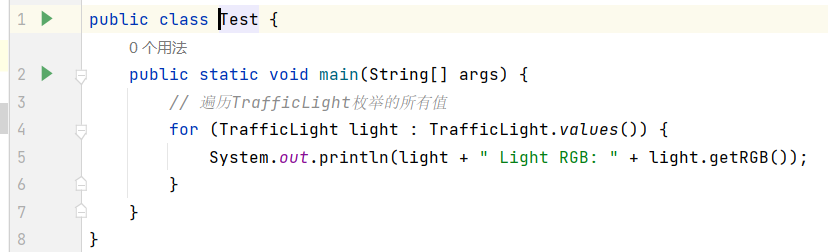
全屏截图：



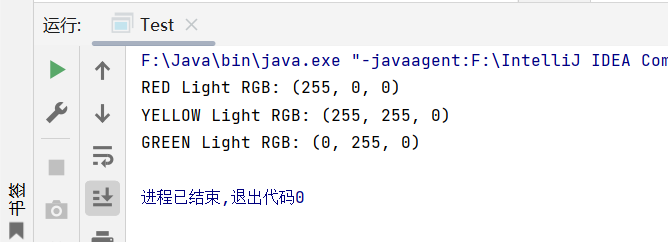
TrafficLight类主要代码：



Test类主要代码：



Test类运行情况：



**三、实验总结与心得记录**

通过完成本次实验，我深入理解和掌握了Java编程中的几个重要概念：图形界面的基本操作、枚举的定义与应用、以及继承机制的实际运用。在实验过程中，我遇到了一些挑战，特别是在实现图形界面和处理用户输入时。通过不断地查阅资料和反复试验，我逐步解决了这些问题。这个过程让我更加熟悉了JavaFX库的使用。同时，通过枚举和继承的实践，我对Java语言的面向对象编程有了更深入的理解。

此外，我也认识到了代码设计的重要性。在实验中，良好的代码结构和清晰的逻辑关系大大提高了代码的可读性和可维护性。例如，通过继承和多态性，我能够用更简洁的代码实现功能更为丰富的程序。