****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验4

**姓名：宋泽涛**

**学号：25120222201292**

**学院：信息学院**

**专业：软件工程**

**完成时间：2024/3/21**

**一、实验目的及要求**

## 实验目的：

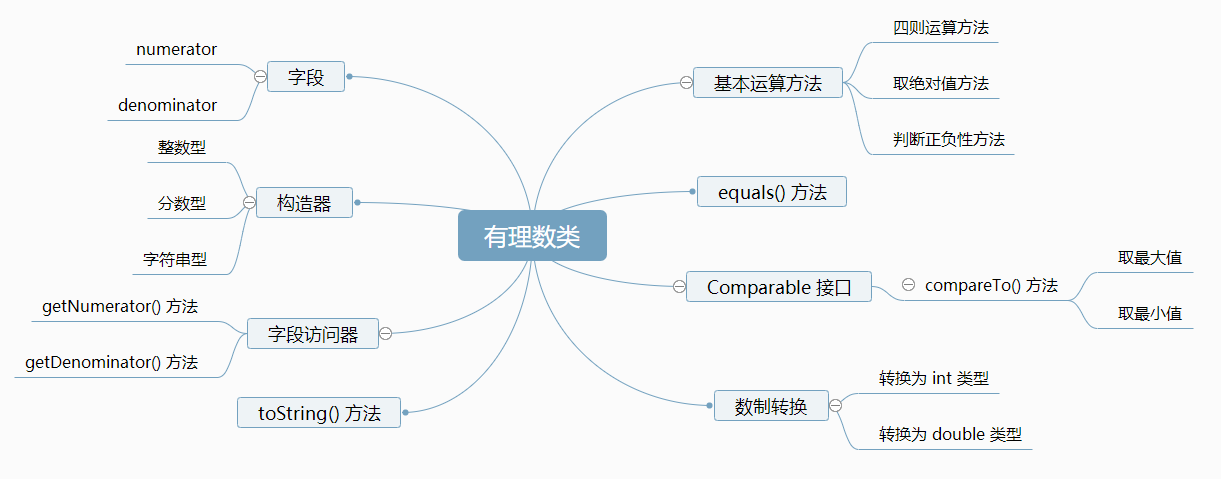
* 开始熟悉图形界面
* 熟悉枚举
* 熟悉继承

## 实验要求：

* 按照题目要求写代码和实验报告，并上传到FTP

**二、实验题目及实现过程**

1. 基本题目：
2. 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
3. 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。

提示：

* + 有理数是有分子、分母以形式a/b表示的数，其中a是分子，b是分母。例如，1/3，3/4，10/4。
  + 有理数的分母不能为0，分子却可以为0。每个整数a等价于有理数a/1。有理数用于分数的精确计算中。例如1/3=0.0000…，它不能使用数据类型double或float的浮点格式精确表示出来，为了得到准确结果，必须使用有理数。
  + 由于有理数与整数、浮点数有许多共同特征，并且Number类是数字包装的根类，因此，把有理数类Rational定义为Number类的一个子类是比较合适的。由于有理数是可比较的，那么Rational类也应该实现Comparable接口。

1. 请你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。

输入样例：

2

5 5

6 6 6

输出样例：

25

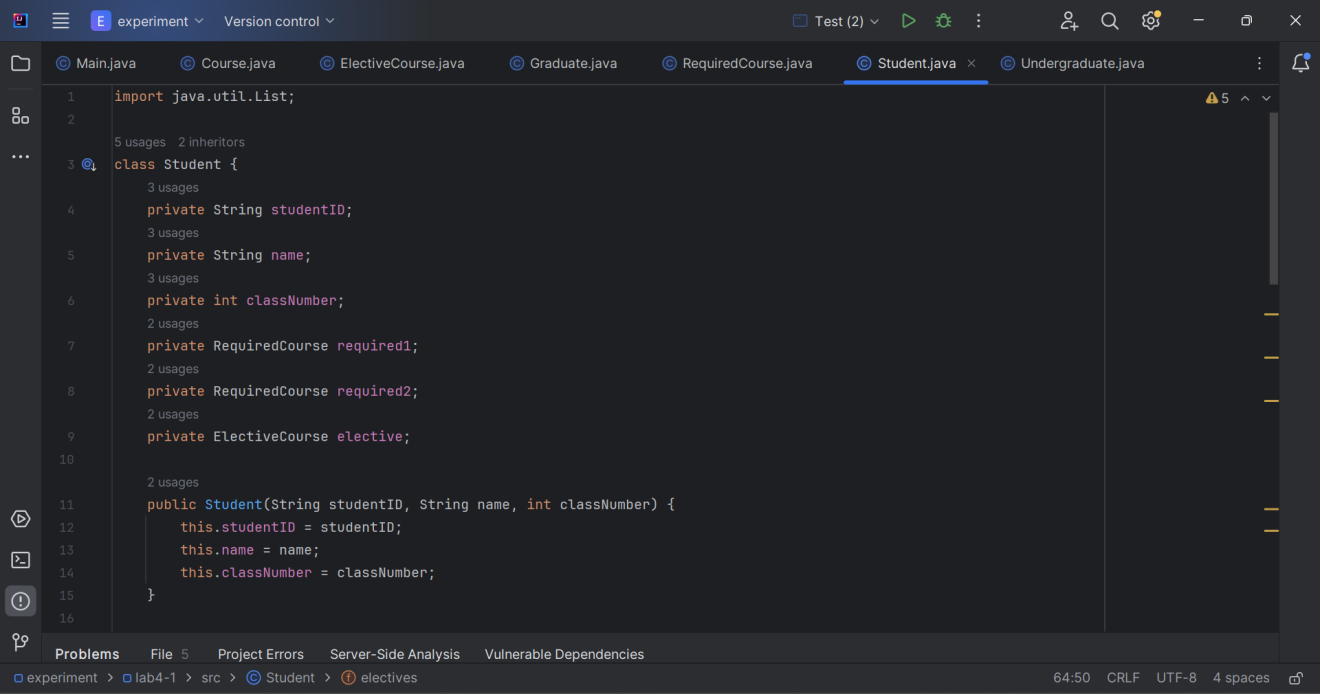
15

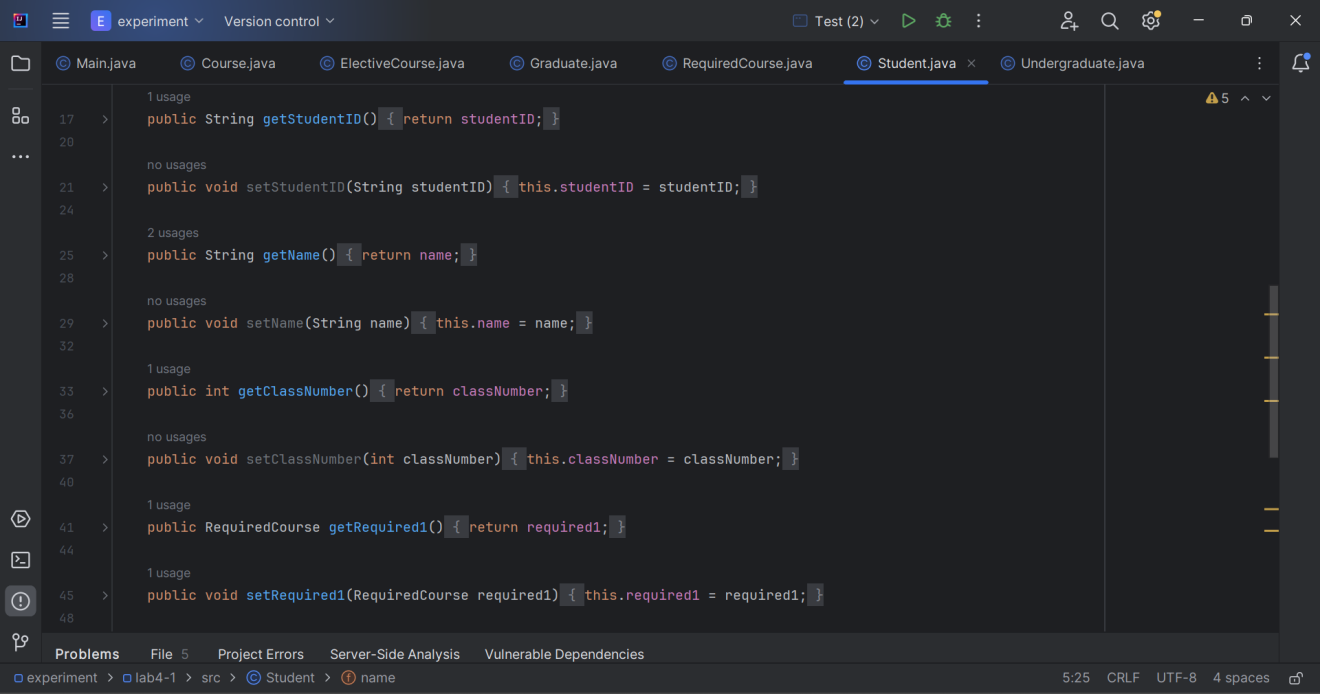
1. 创建一个简单的JavaFX绘图程序，要求如下：
2. 随机产生一个随机数（0,1,2），三个随机数分别对应直线、矩形和椭圆三种图形。根据随机数对应图形，提示用户输入图形所需初始化参数，提示信息应包括参数的范围，用户输入后进行范围检查，若合法，则根据用户输入的信息在界面上绘制出相应的图形。
3. 绘制20个图形后，不再创建新的图形。
4. 写一个交通信号灯枚举类TrafficLight，并在Test类中输出每种灯颜色的RGB值。

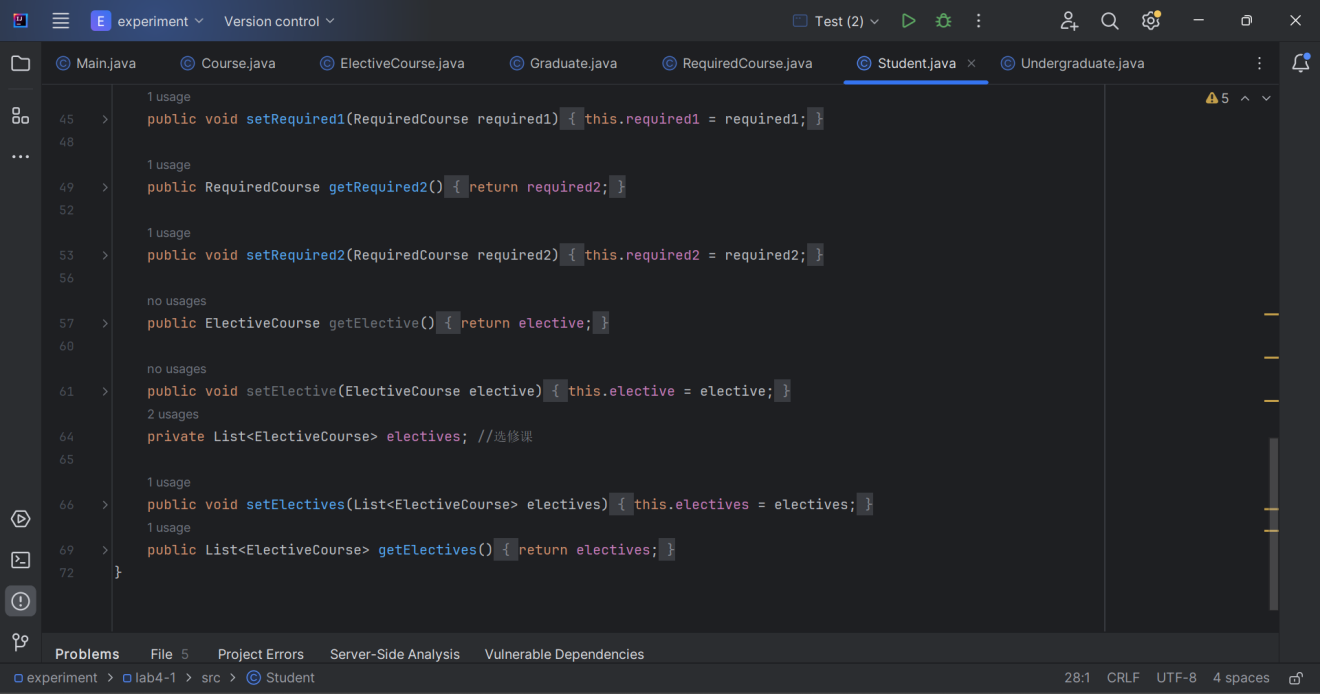
**实现过程**

1.分析题目需求，他需要我们创建4个学生信息与4门课程信息，并且设计一个自动选课部分和教学秘书手动选课部分。对于4个学生信息，还分为2个本科生和2个研究生，对于课程信息也是如此，分为2门必修和2门选修。我们可以定义两个父类：学生类和课程类。之后再定义本科生类与研究生类去继承学生类，必修课类和选修课类去继承课程类。完成这些类的创建后就可以创建实例对象进行过程实现。具体代码如下

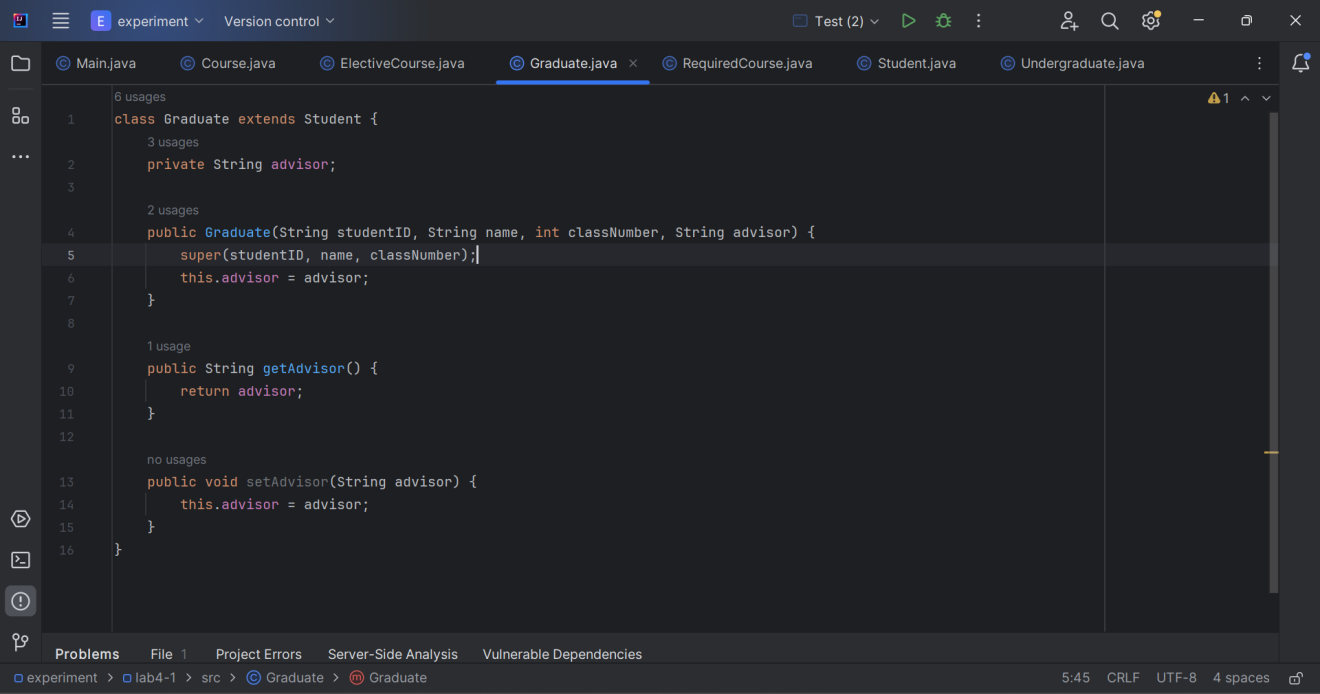
**学生类**

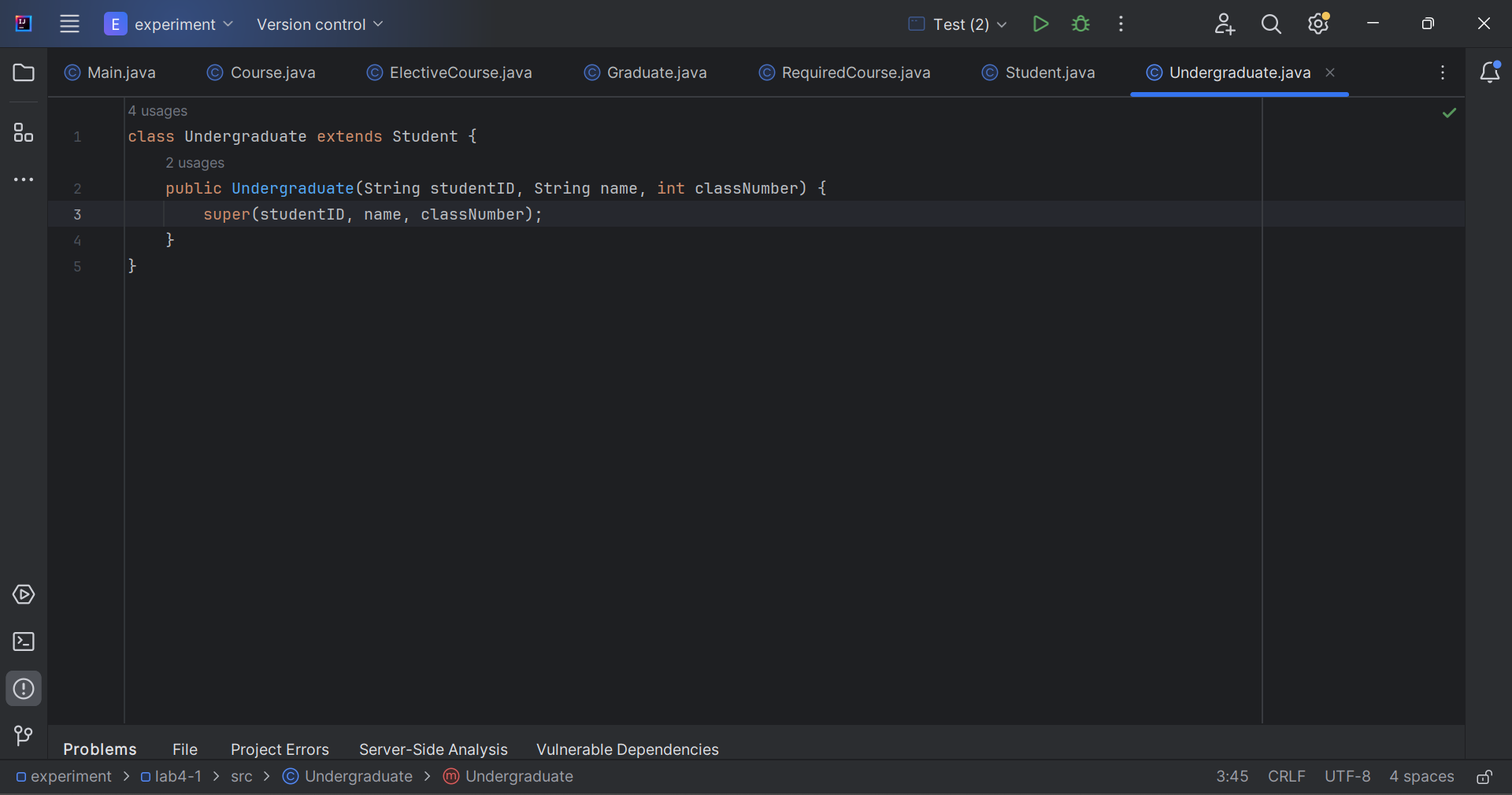




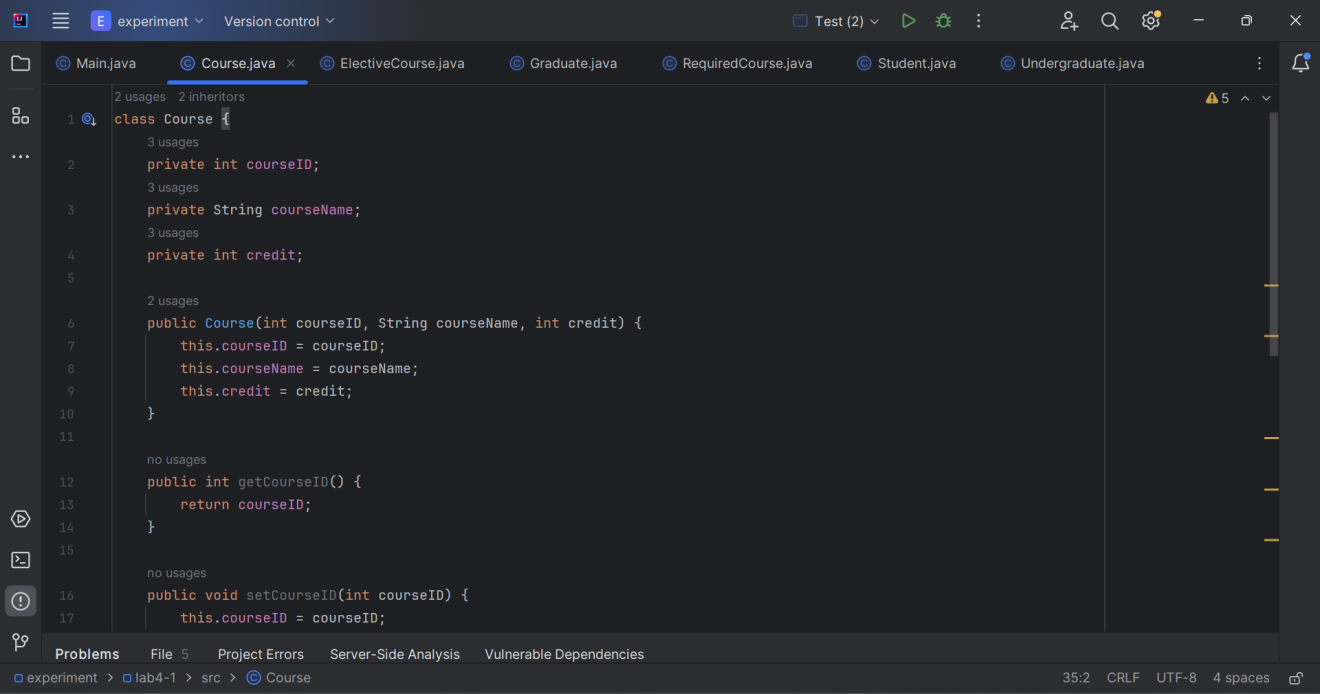


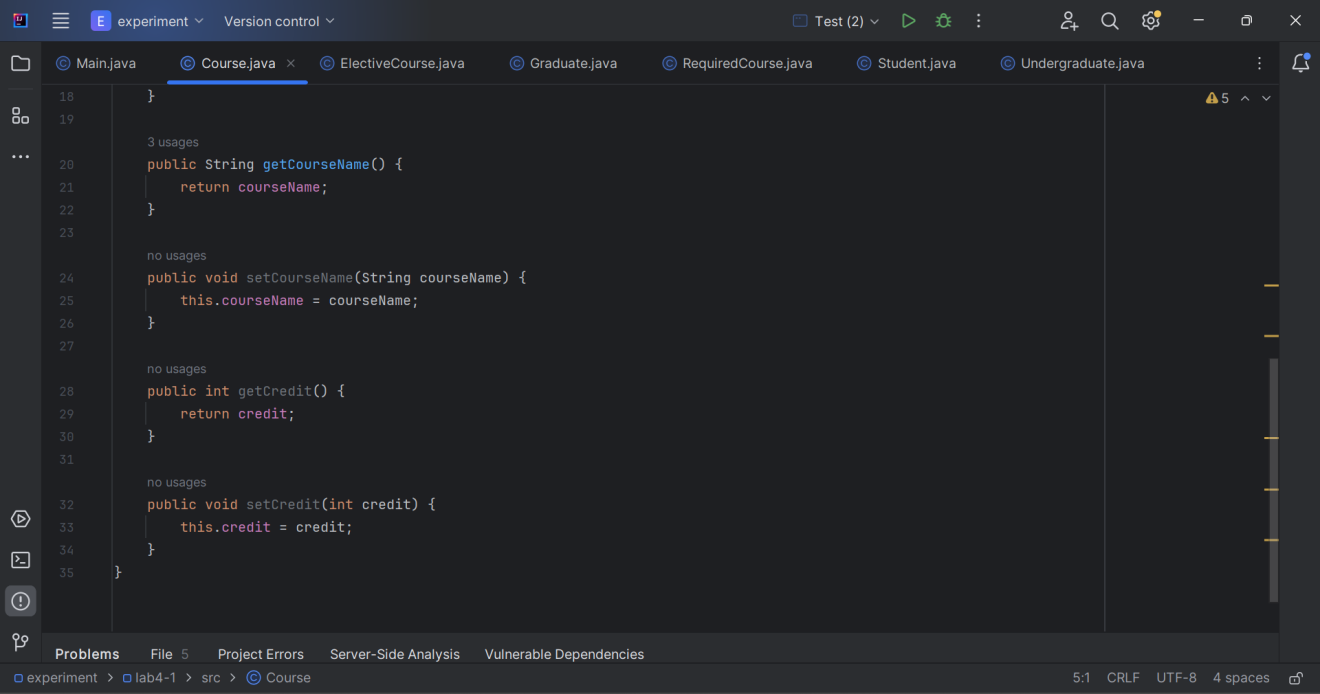
**学生类的子类**



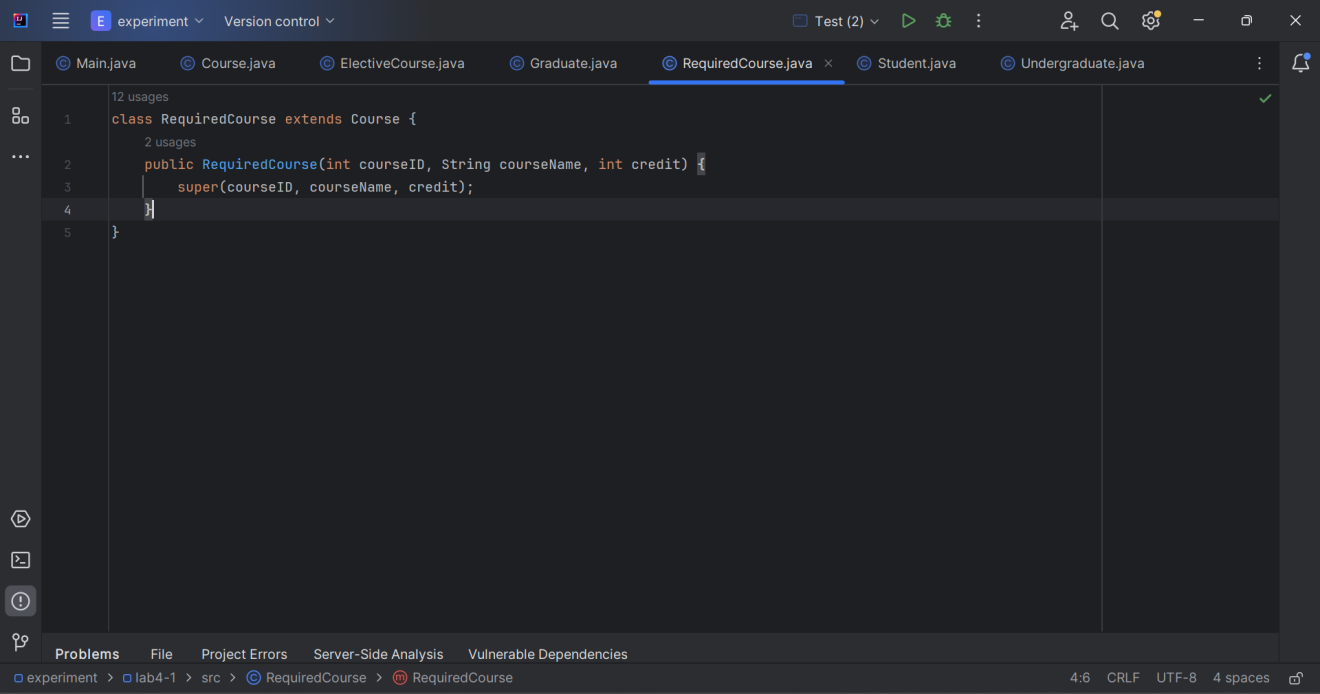


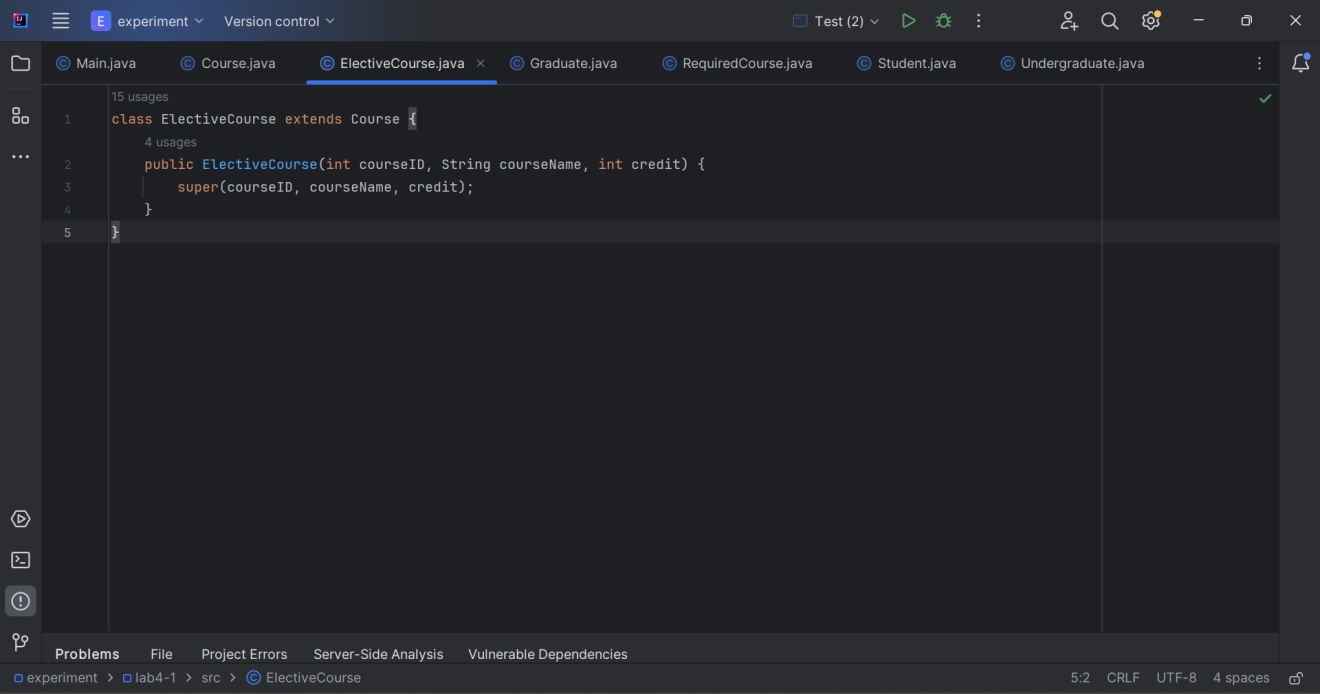
**课程类**





**课程类的子类**





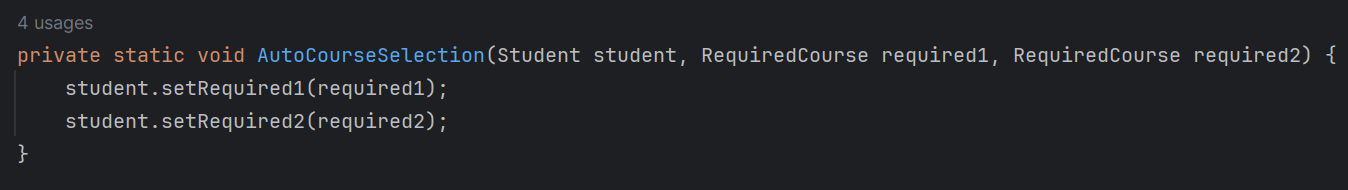
从以上类的定义可以看出，子类可以通过super关键字对父类定义的变量或构造方法进行调用，这使得在程序设计的过程中代码更加易读。接下来看具体实现过程

(1).在主函数中，我们先实例化4个学生对象，4个课程对象，并调用构造器对其进行初始化操作

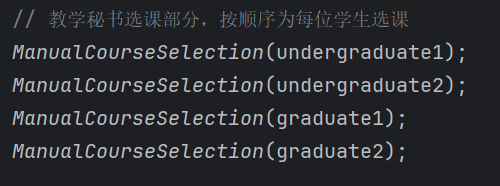


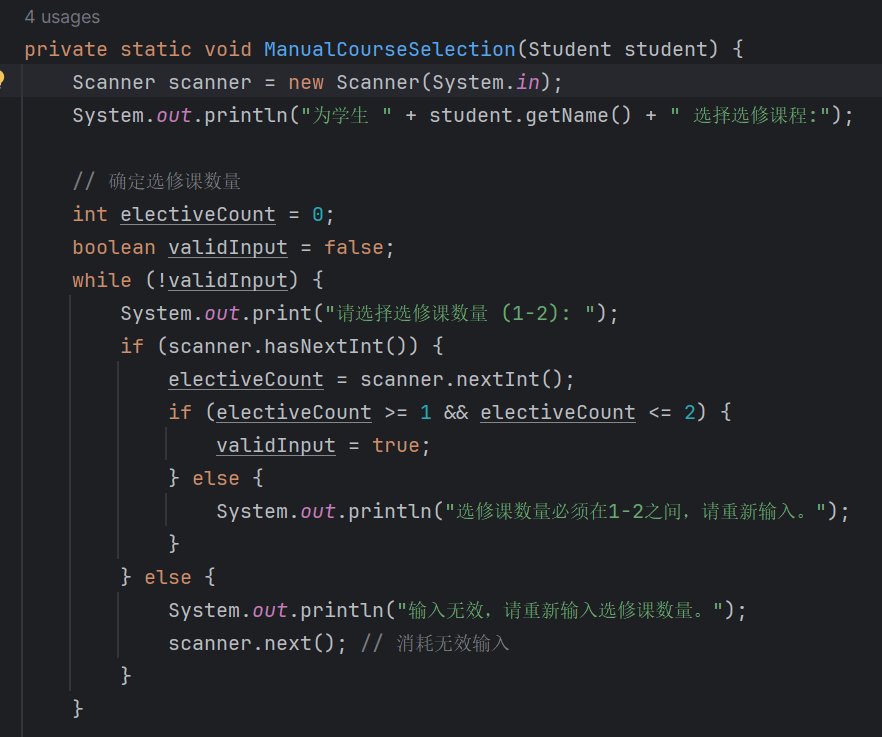
(2).接下来是自动选课部分，我们调用AutoCourseSelection方法，在该方法中调用传入实例对象的set方法进行初始化

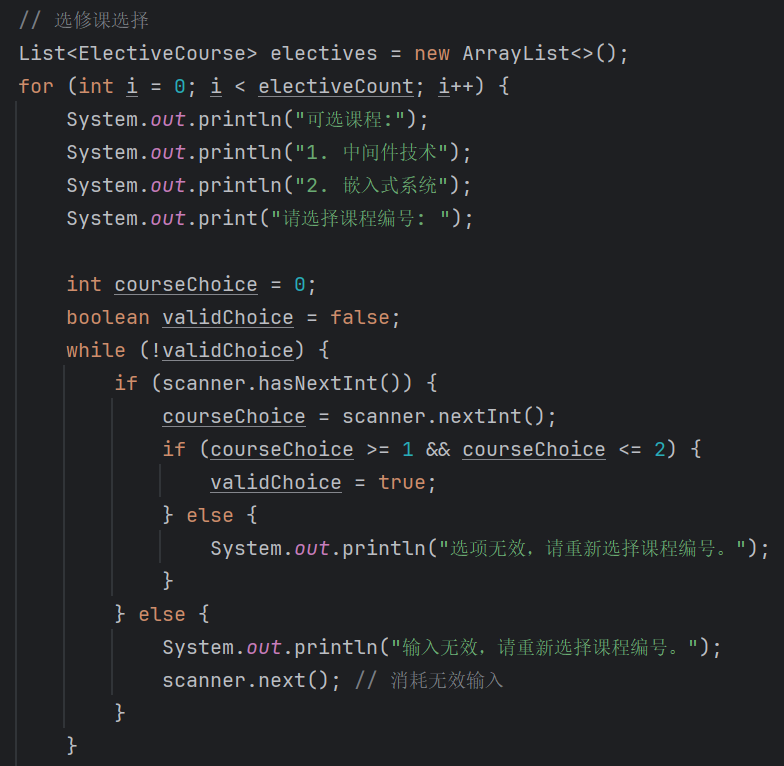


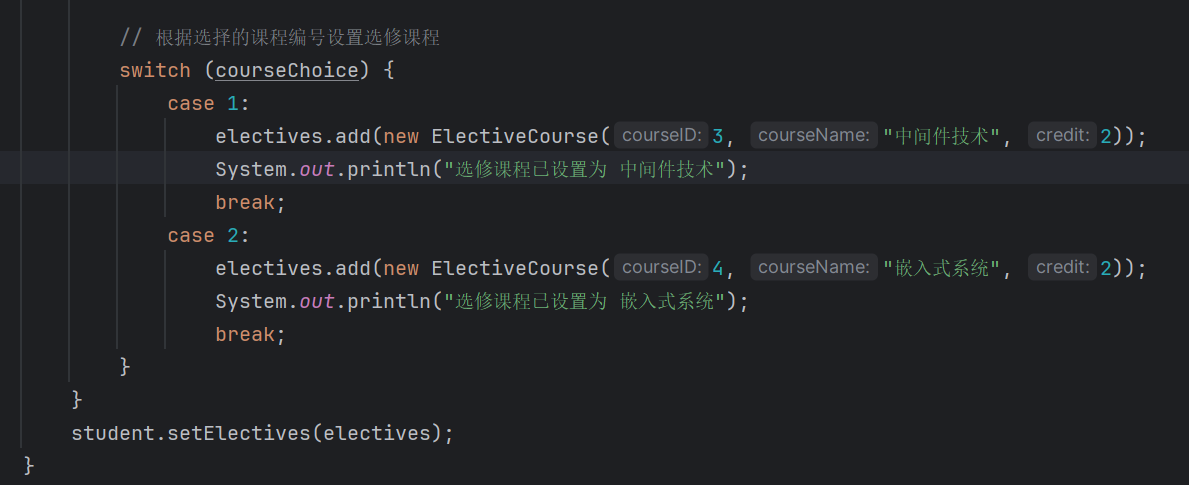


(3).自动选课操作完成后，调用手动选课方法。在该方法中，传入了一个学生对象，根据需求，教秘需要知道学生要选几门选修课，因此首先读入选修课的数量。确定之后再创建一个ArrayList存放每位学生选修的课程，便于之后的输出。教秘选择完成后，给予相应的反馈，让用户知道选修了那些具体的课程。

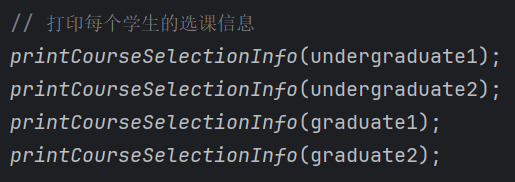


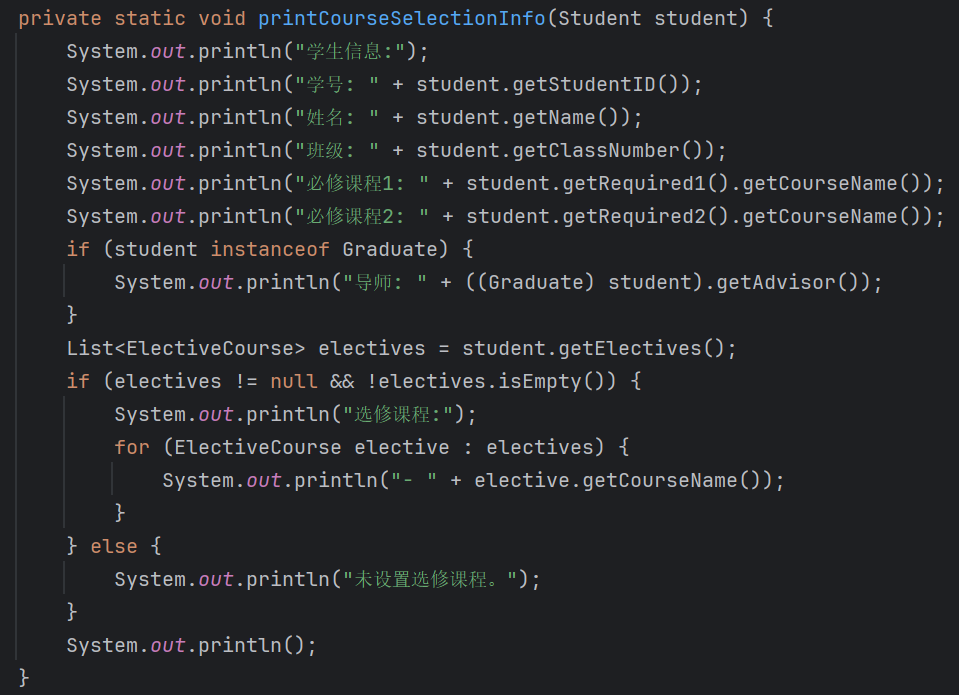




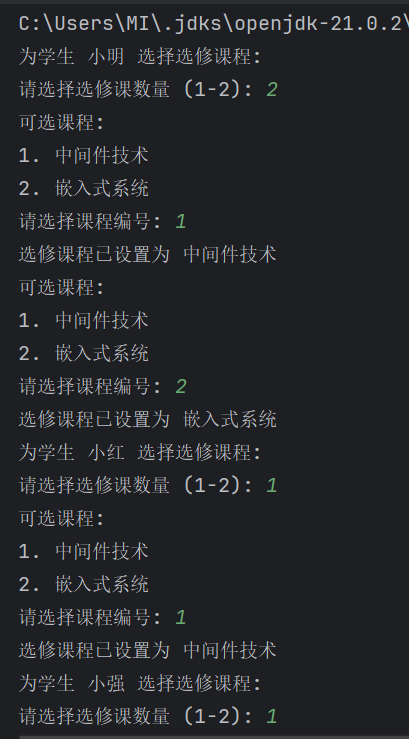


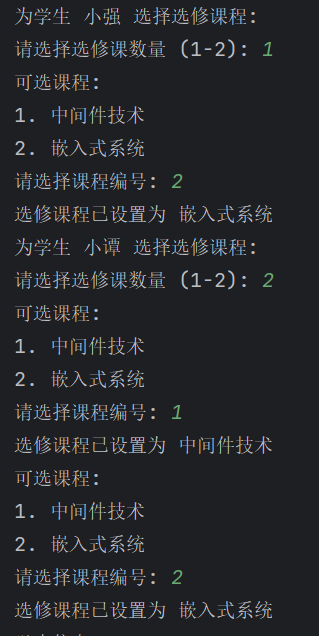
(4).最后打印每个学生的选课信息即可

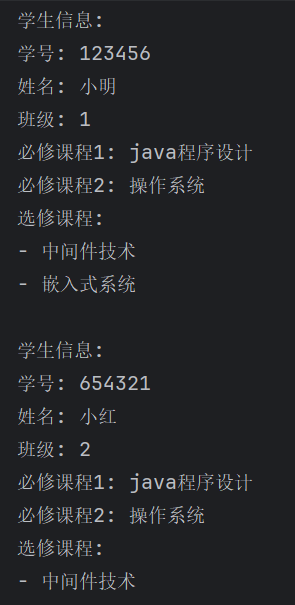


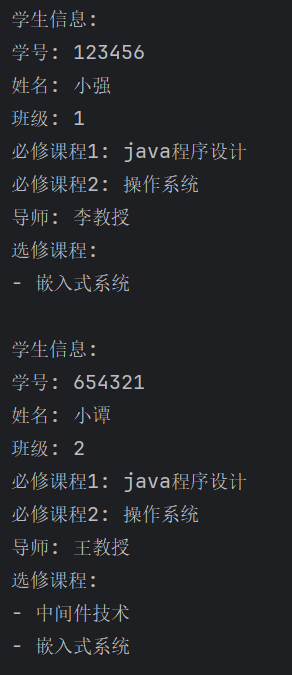


运行结果如下



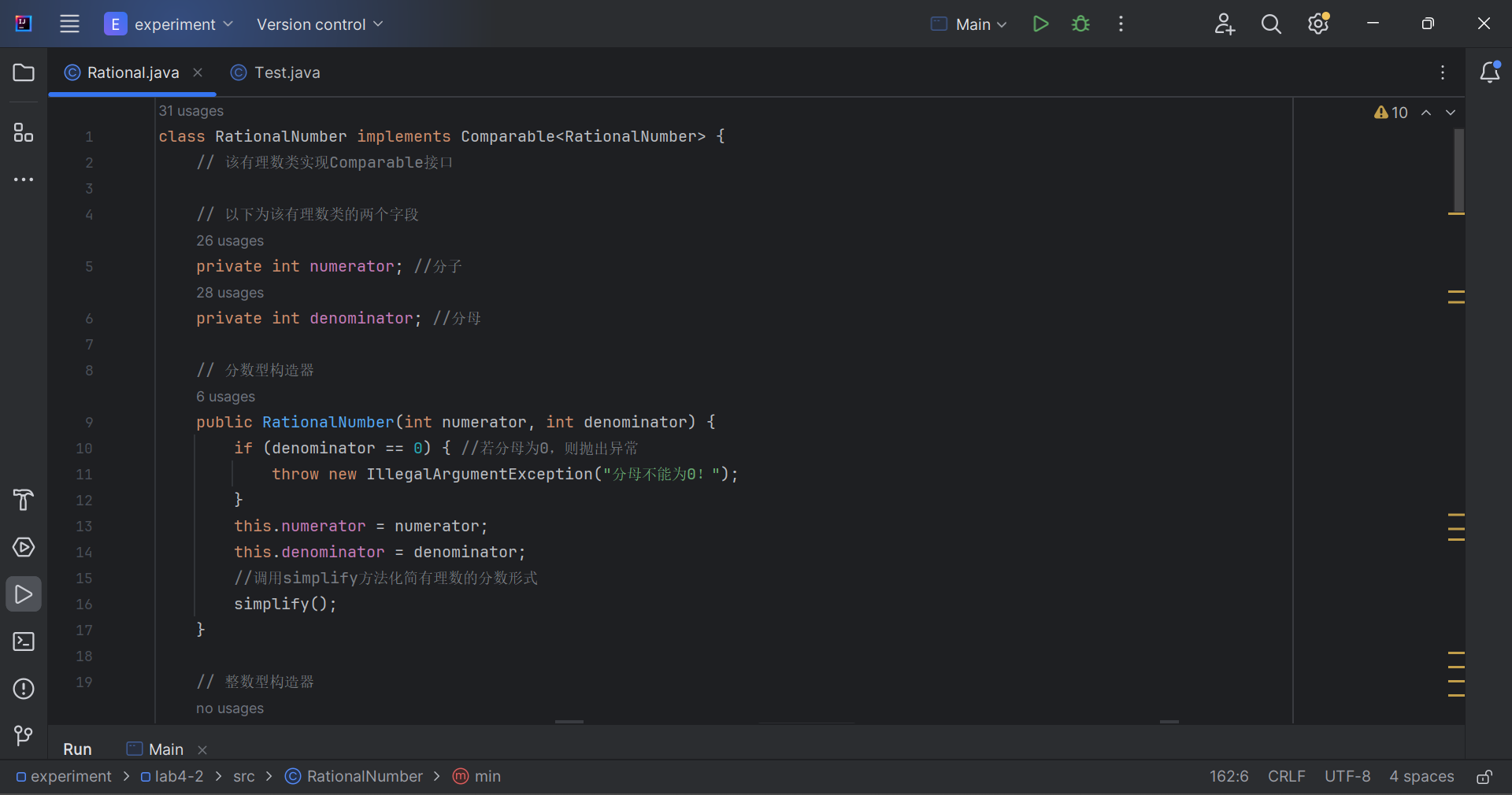




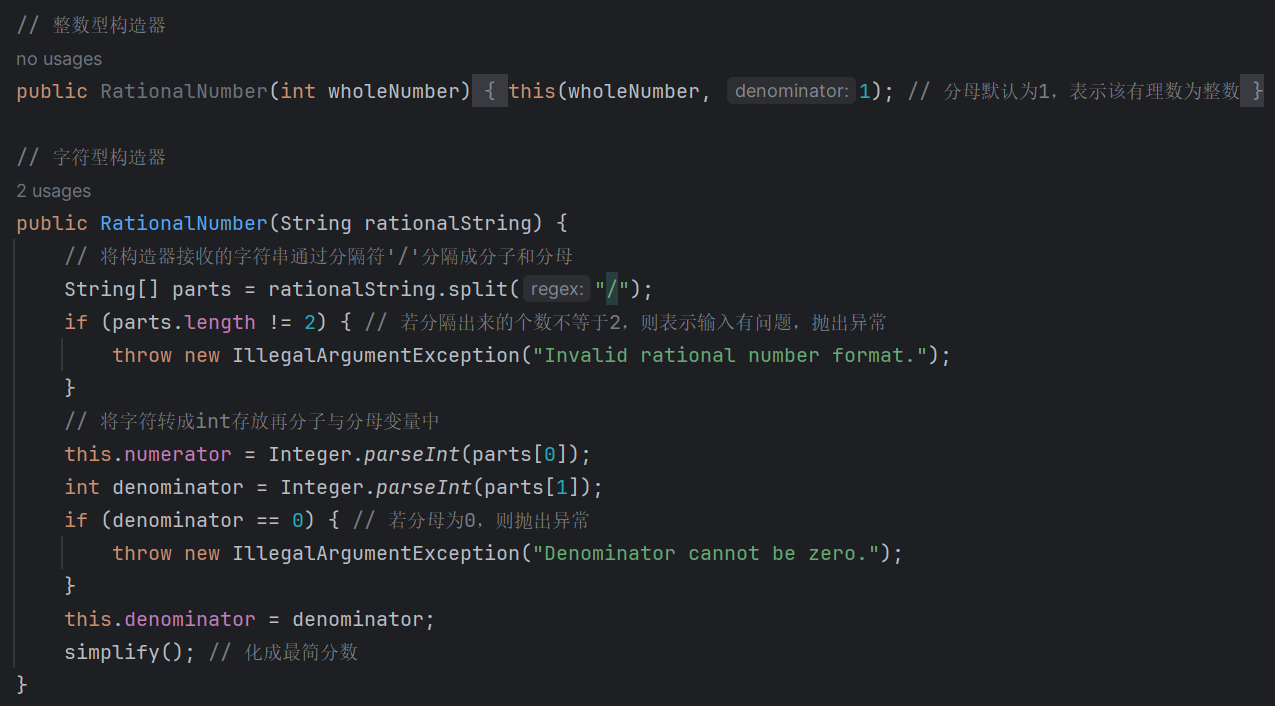


1. 根据需求，需要编写一个有理数类，在该类中提供各种类型的构造器、字段访问器、toString方法、基本运算方法、equals方法、数制转换，还需要实现Comparable接口。同时，需求中给定了该有理数类的字段：numerator和denominator，分子和分母。根据需求创建以下类

(1).在有理数类中定义两个int类型变量，用来存放分子和分母，然后重写构造方法，实现分数型构造器。在分数型构造器中，先判断分母是否为0，若为0，则不合法，抛出异常，否则就对当前类的变量进行赋值，最后调用的simplify方法会在后面进行解释。

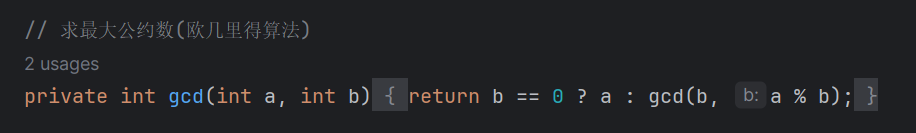


(2).定义整数型构造器与字符型构造器。整数型构造器较简单，只需要将分母置为1，再接收分子输入即可。对于字符型构造器，需求要求我们接收形如”1/2”的字符，因此我们可以调用String类的split方法，根据’/’对字符进行分割，将分子分母存放在String数组中。之后先判断String数组的长度是否为2，若不为2则表示输入有问题，抛出异常，否则进入下一步。在下一步中，先将String变量转成int类型再存在类中的两个变量中，剩余操作同(1)

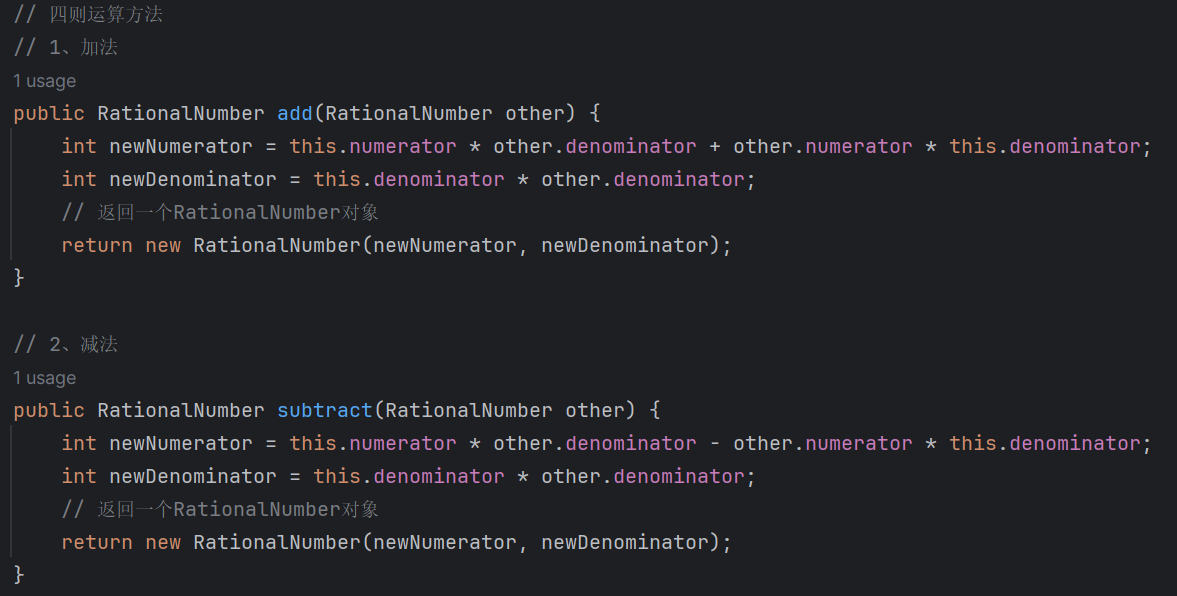


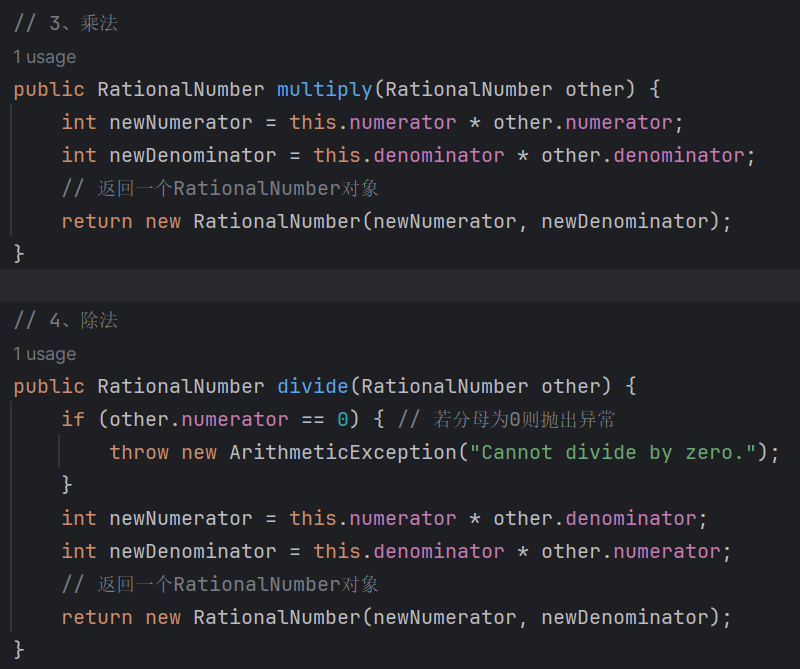
(3).创建字段访问器，编写之前的分数化简方法。字段访问器用类的get方法即可，simplify方法需要实现一个简单的gcd算法，具体可以看如下代码



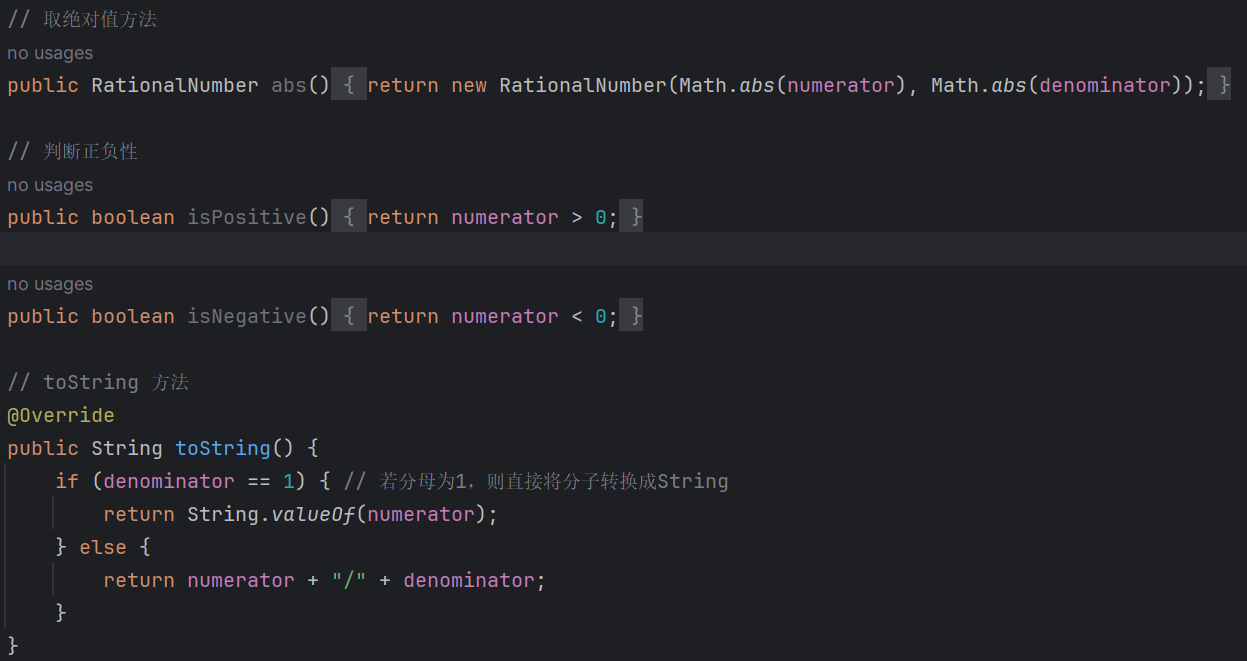


(4).实现四则运算方法，需要在草稿纸上模拟一遍分数加减乘除，然后编写对应方法即可。需要注意的是，除法需要判断分母是否为0，具体看如下代码

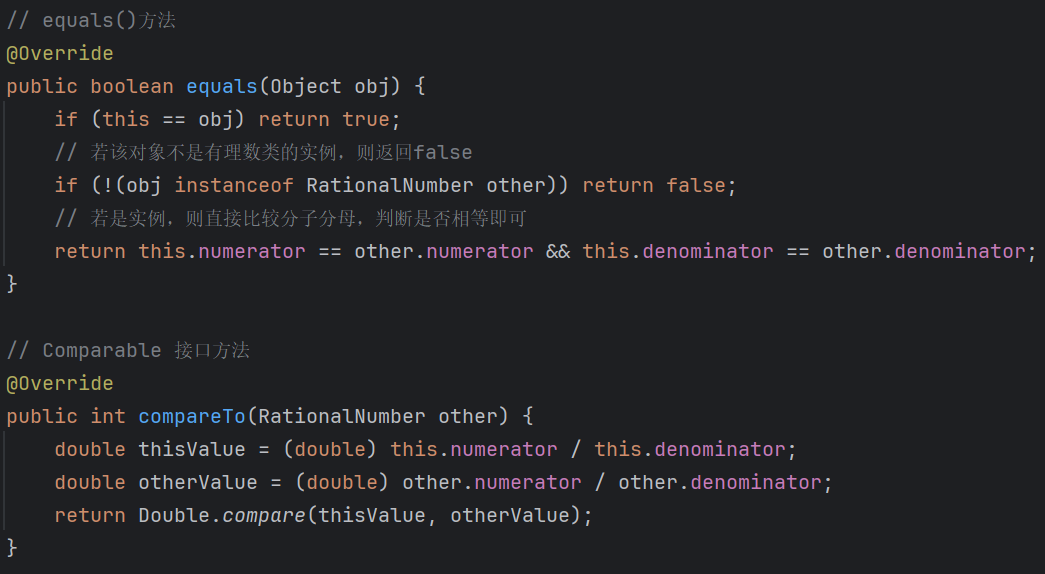




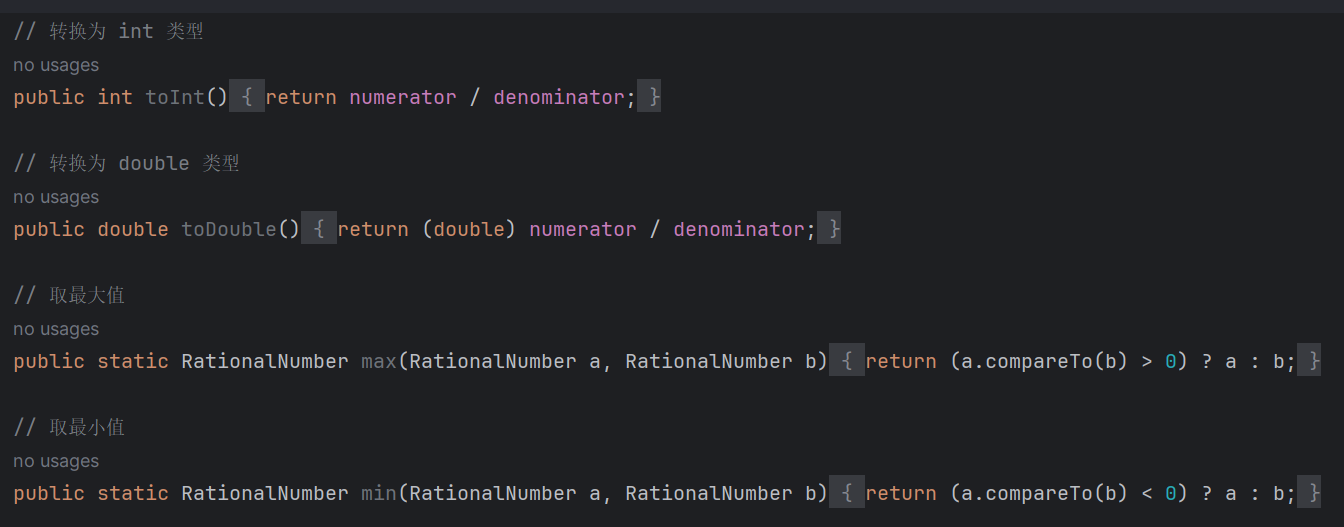
(5).取绝对值方法调用一下Math包中的abs方法即可，判断正负性方法直接返回numerator > 0或numerator < 0的值即可。对于toString方法，我们需要对其进行重写，对分母是否为1进行分支编写，若为1则表示正数，否则为分数，最终返回一个String即可。具体看如下代码



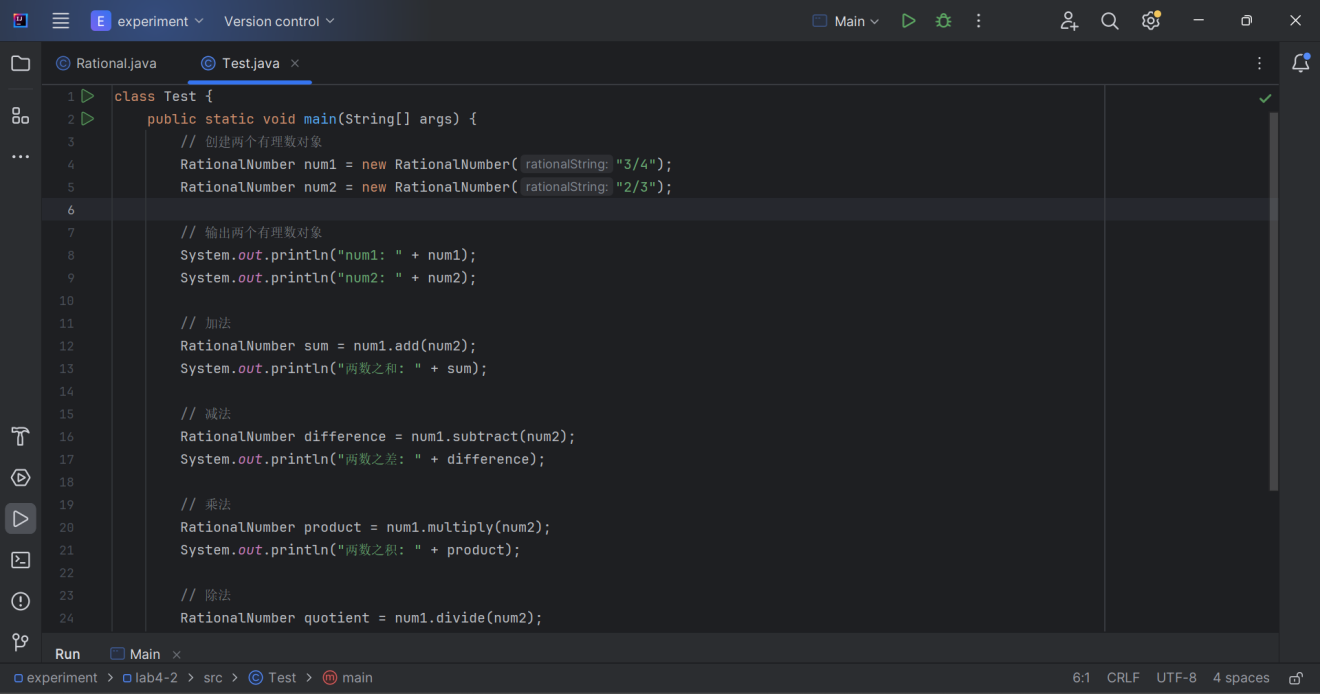
(6).重写equals方法与comparable方法

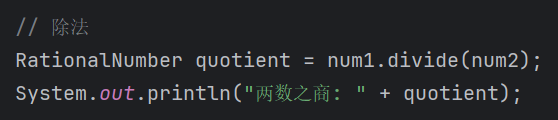


(7).类型转换与取最大最小值

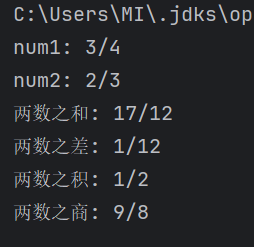


Test类如下





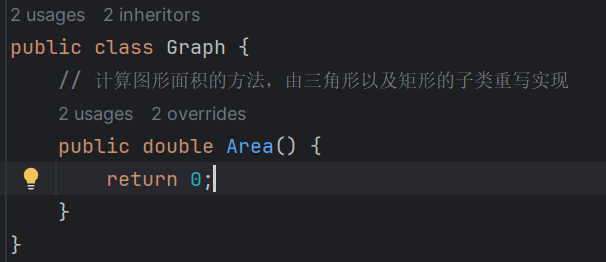
测试结果如下



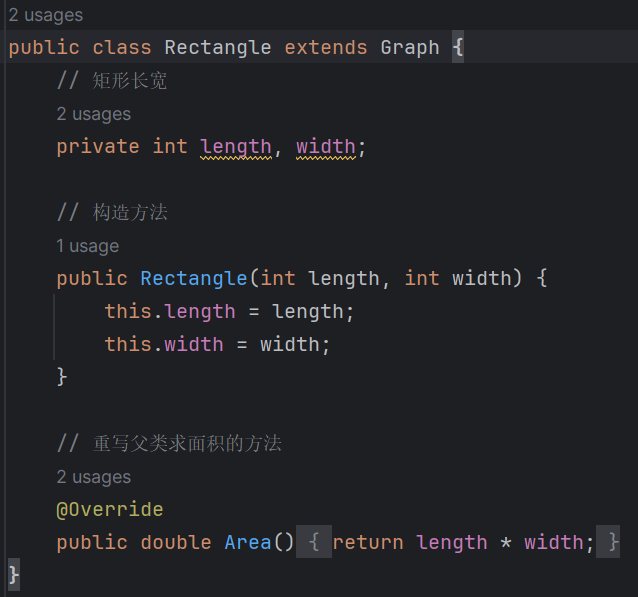
3.根据需求，我们要实现一个基础图形类，然后实现三角形类和矩形类，并在相应类中实现输出图形面积。

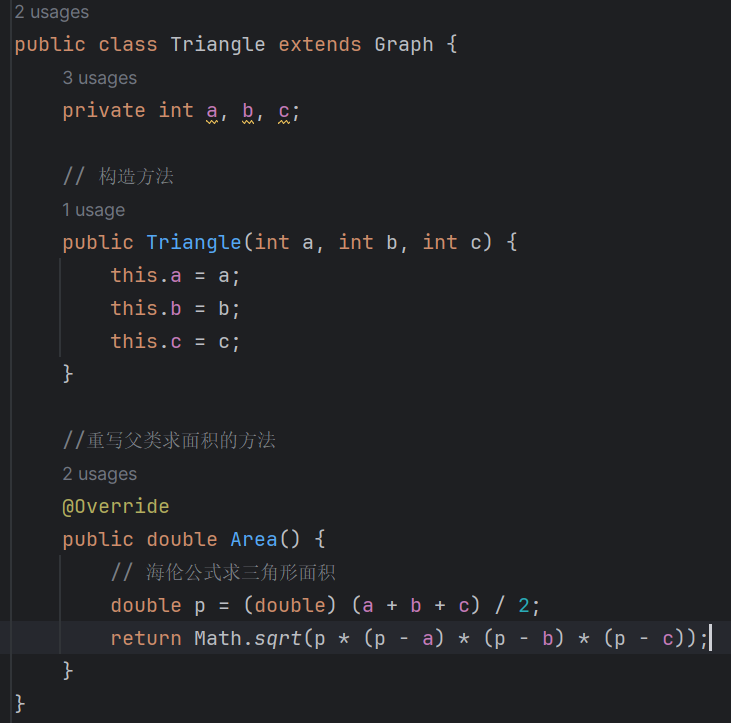
(1).由于基础类Graph的子类只需要实现计算图形面积，所以在父类中也不需要定义其他变量，只需要定义一个父类方法，用于计算图形面积，然后将该父类方法交给子类重写即可

父类Graph

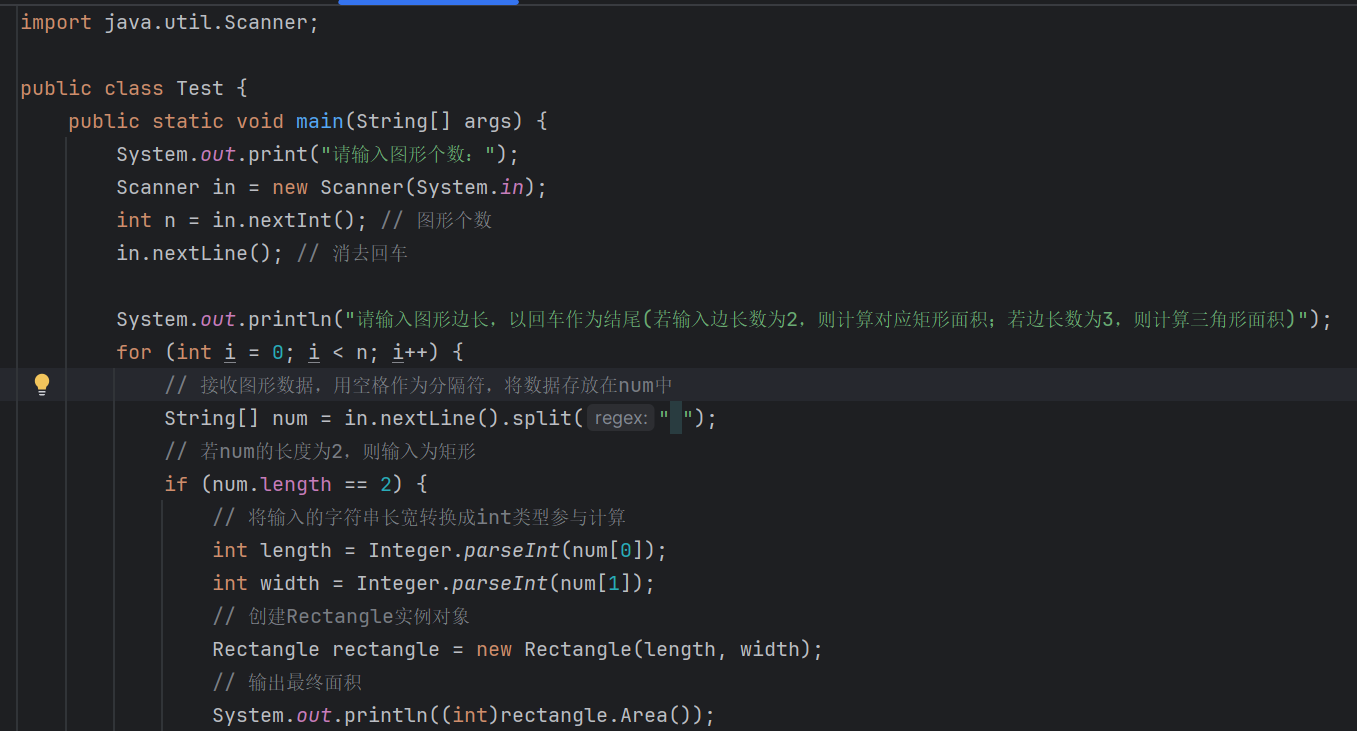


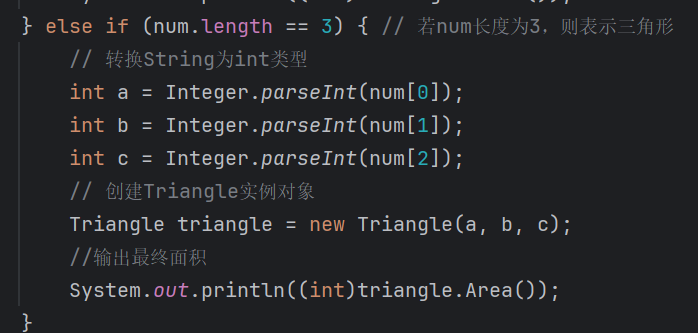
子类Rectangle和Triangle



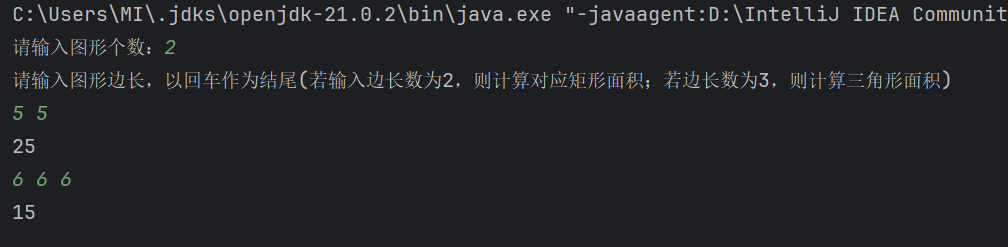


(2).测试代码如下



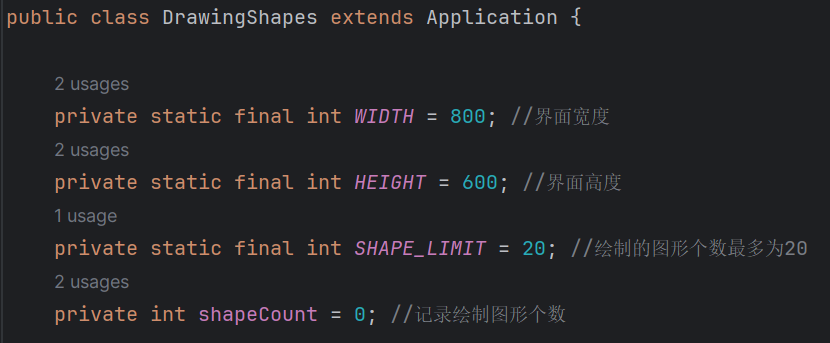


测试结果如下



4.根据需求，我们可以创建一张画布，每次产生随机数过后，进行相应的初始化参数询问，然后在画布上画出图形，每次画完图形后进行画布展示，展示出上次操作的结果。总共进行20次操作，完成操作后直接将最终画布展示出来。具体实现过程如下

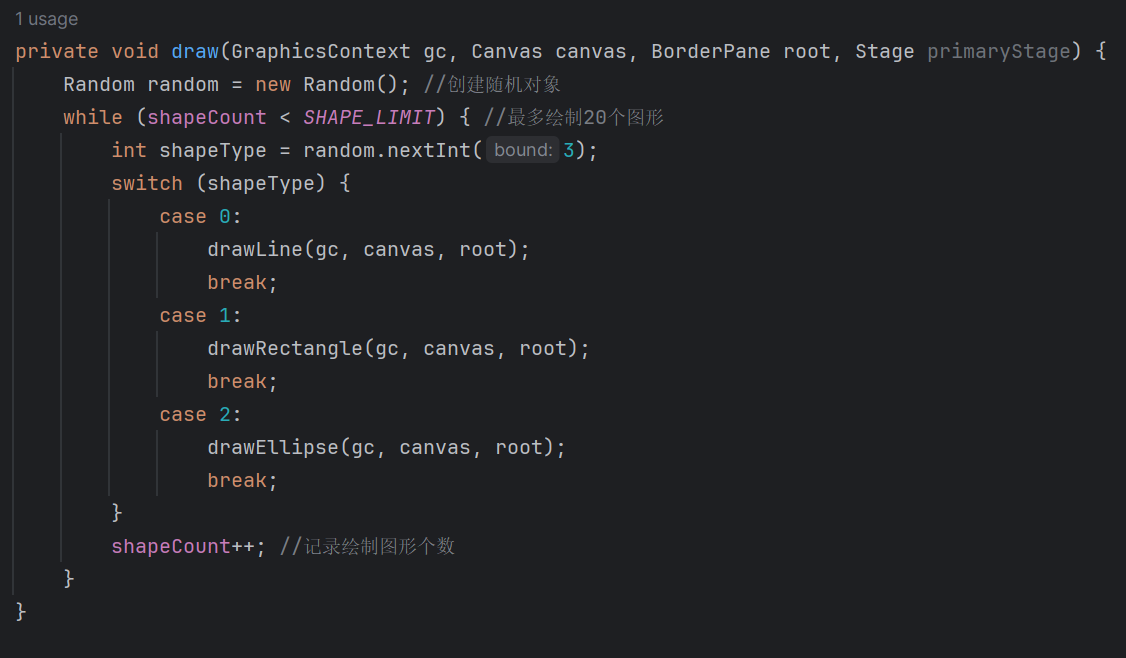
(1).定义画布大小和图形个数，同时还要定义一个变量记录当前已绘制的图形个数



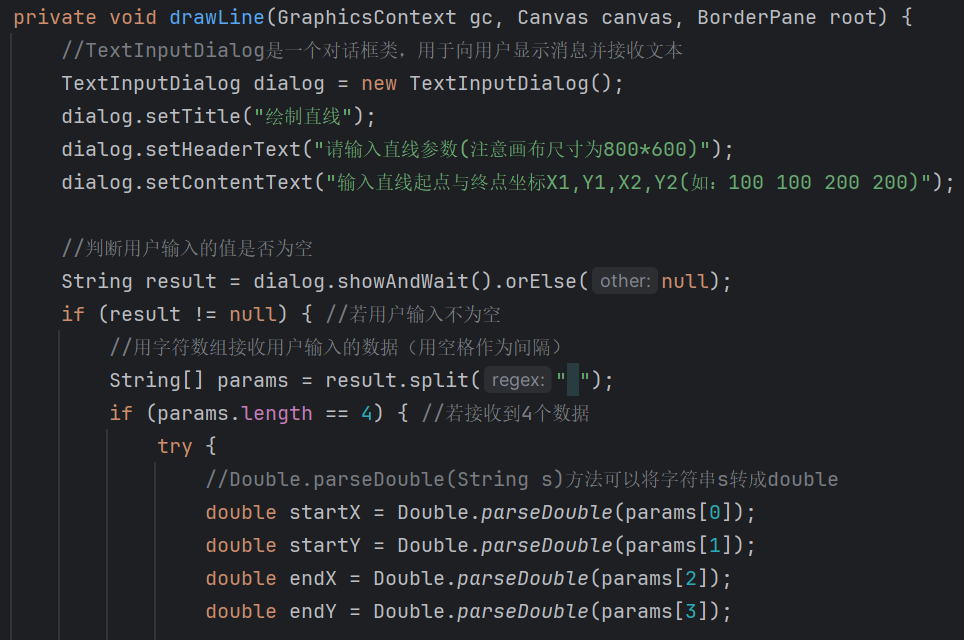
(2).在重写的start方法中，创建布局对象，用于调整与用户交互的UI界面案件布局；再创建画布对象，在上面进行作画；然后再创建GraphicsContext对象，用于在画布上进行绘制。然后调用绘制方法，完成绘制后创建Scene和Stage，将最终结果展现在上面，最后展示Stage即可。

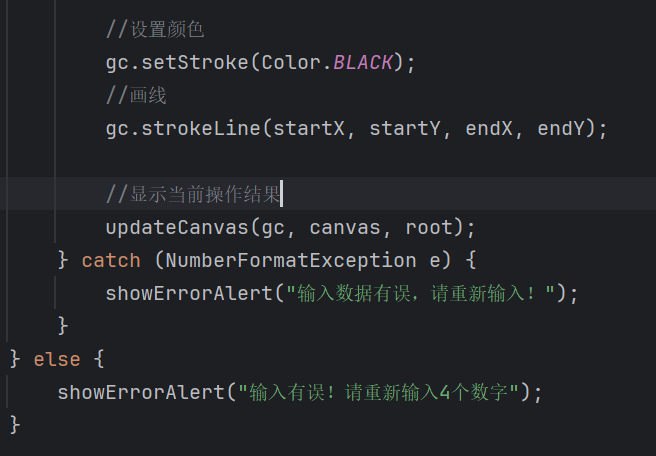


(3)绘制方法draw

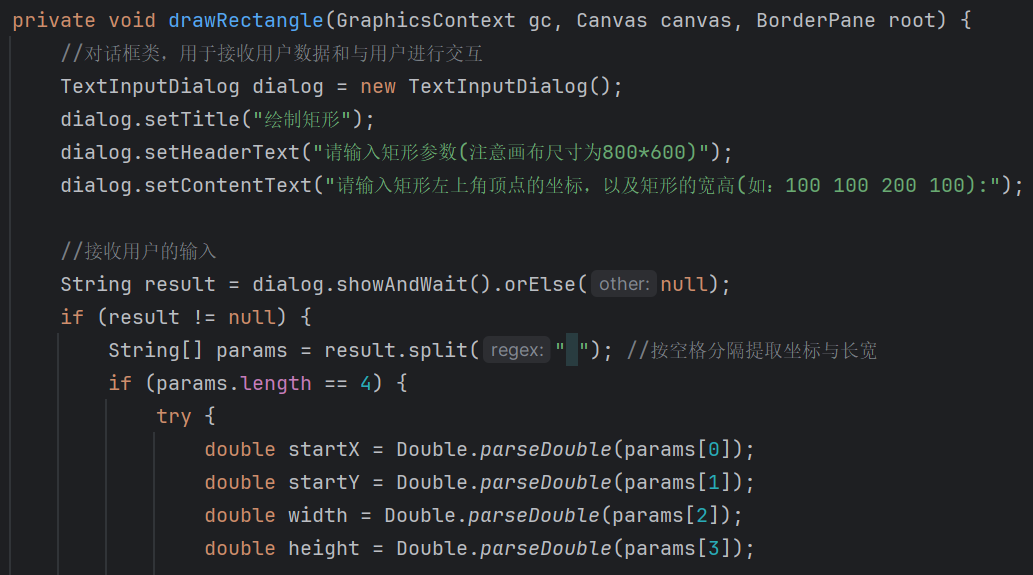


(4).绘制直线方法





(5).绘制矩形方法



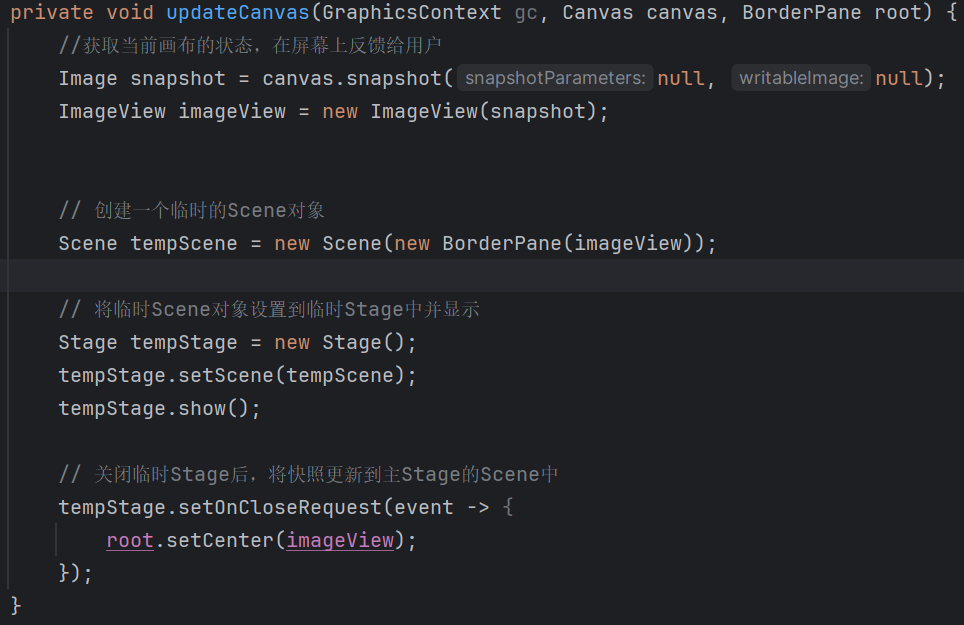


(6).绘制椭圆方法

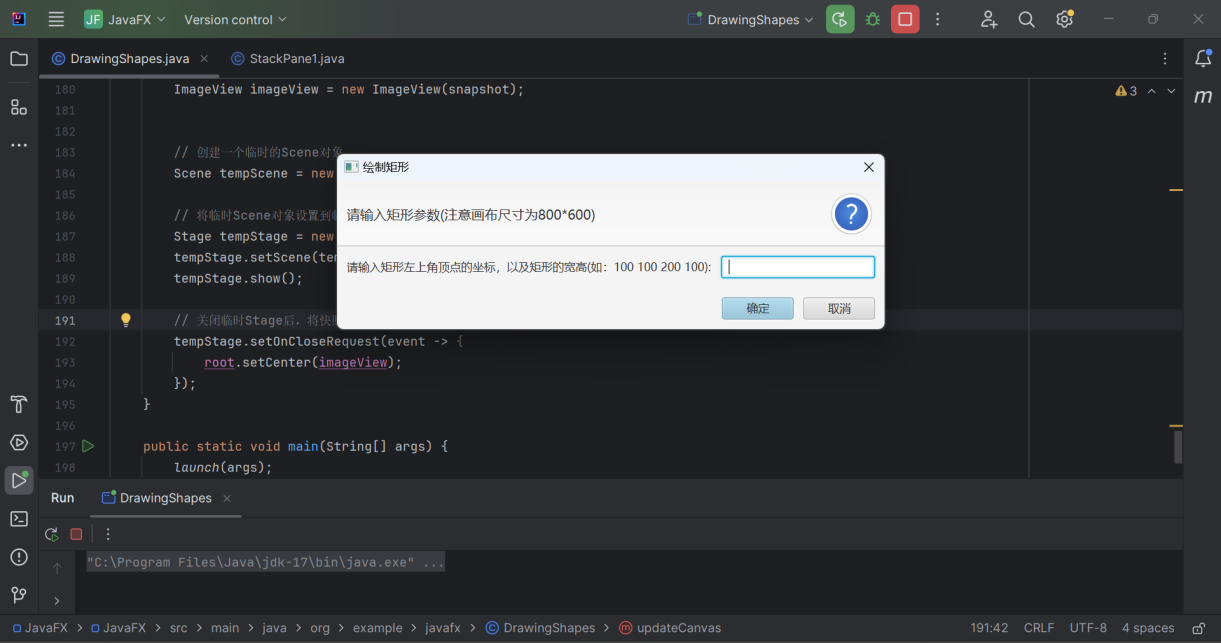




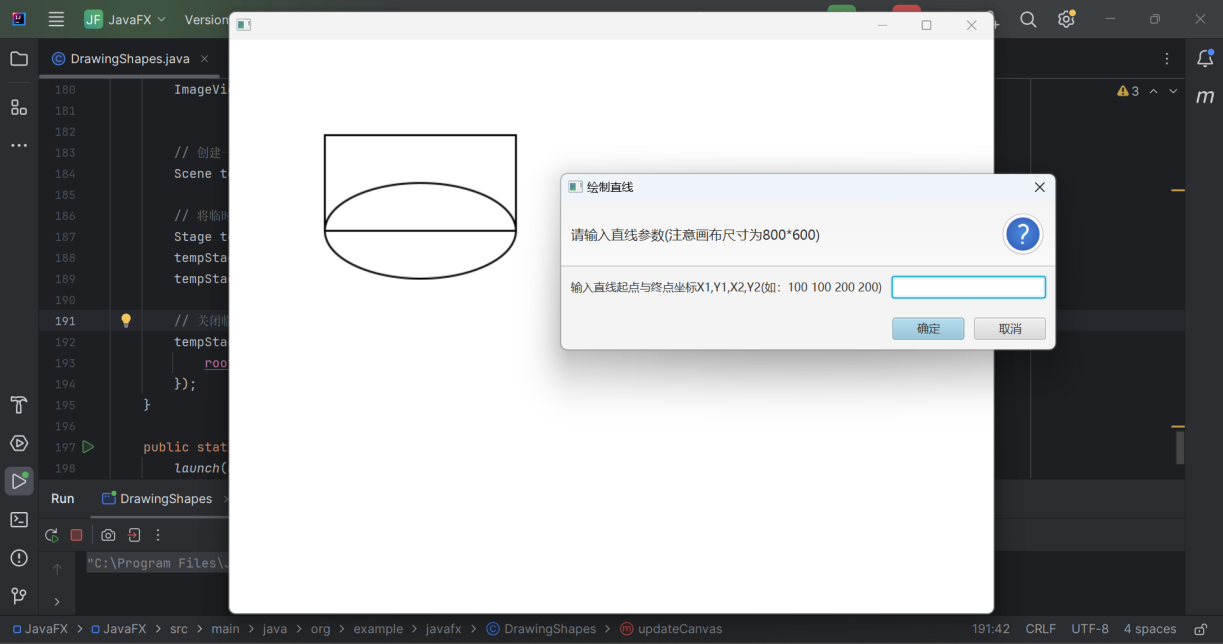
(7).显示每次操作后的结果方法updateCanvas

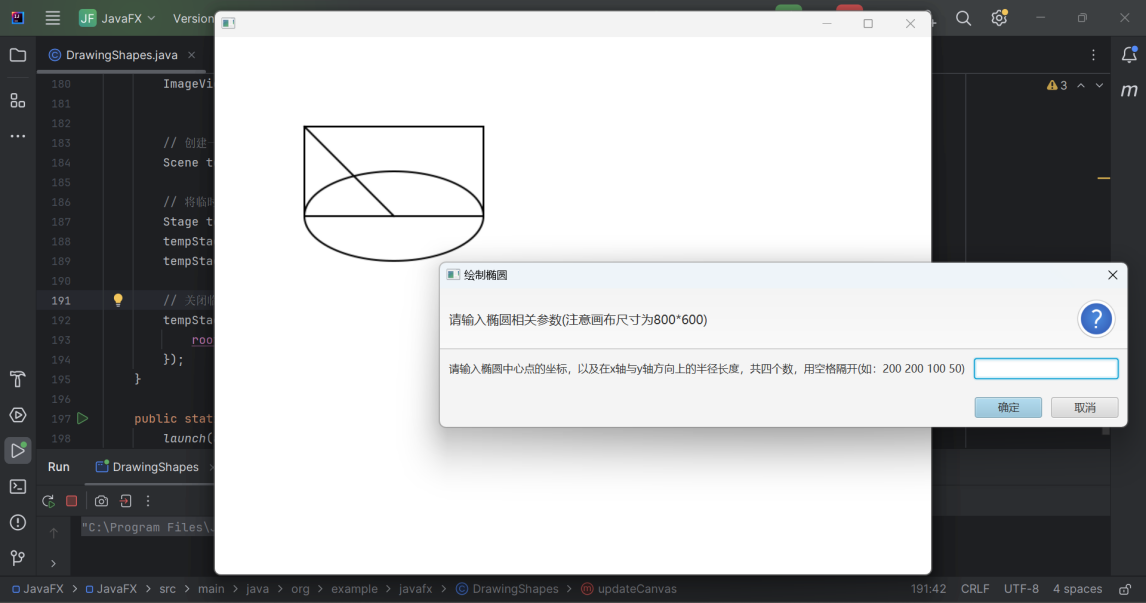


运行结果如下

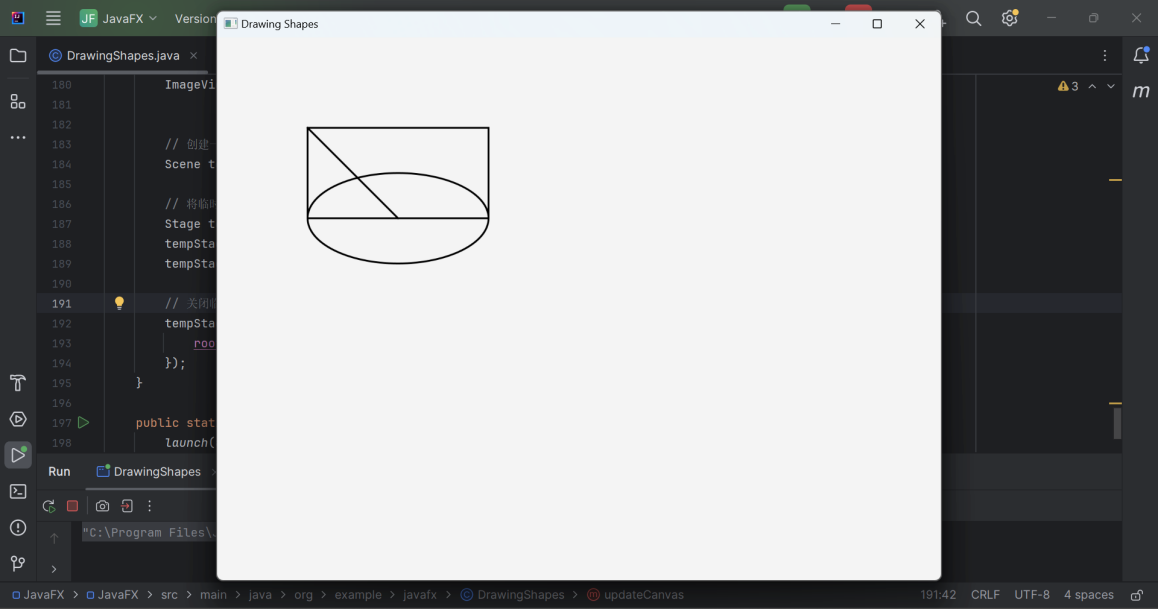






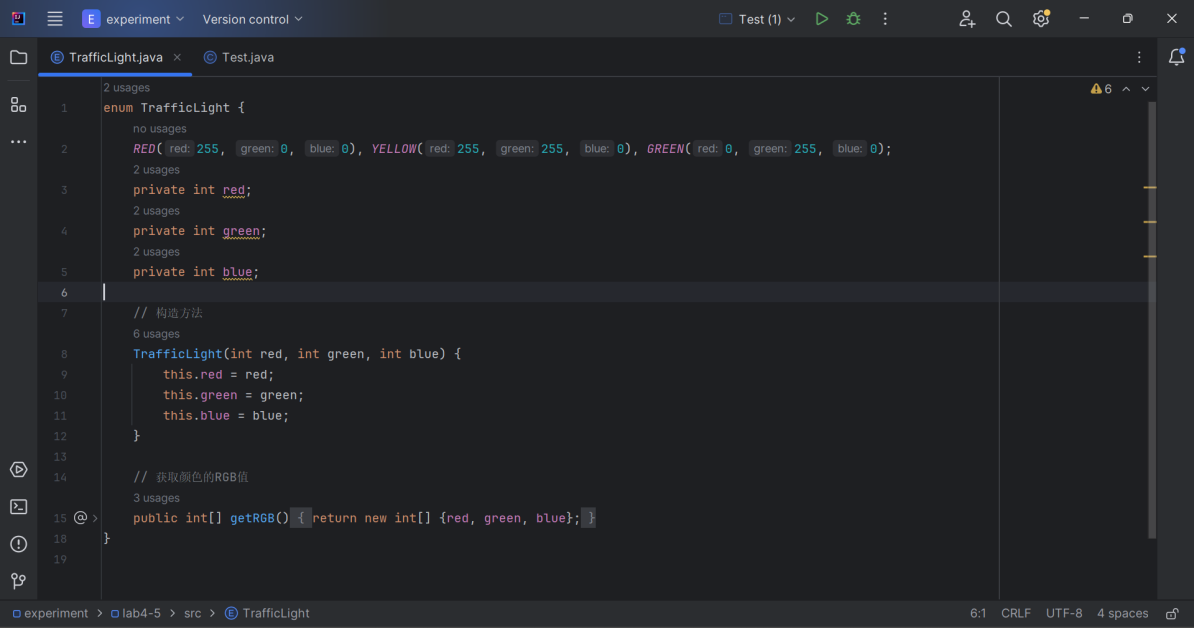


剩余绘画次数均跳过（不进行绘画操作），最终画布如下

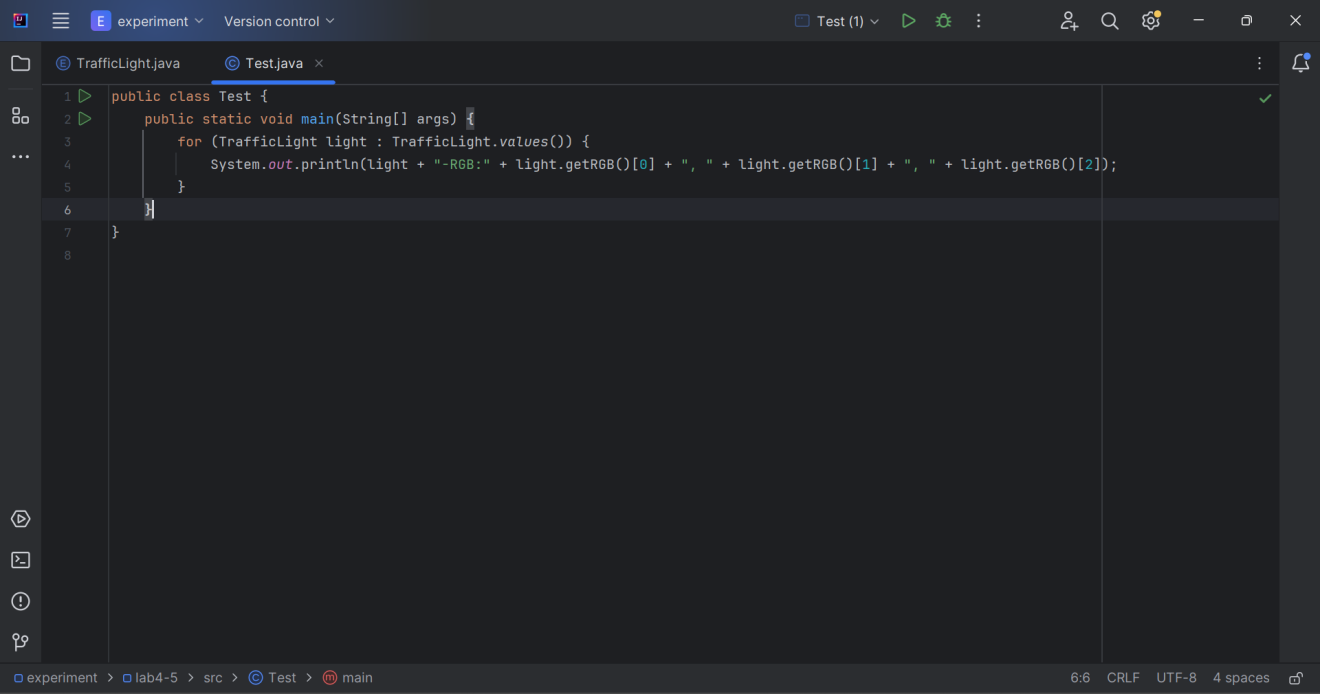


5.根据要求，实现一个枚举类即可

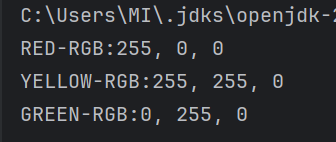
(1).枚举类Traffic



(2).测试类Test



测试结果如下



**三、实验总结与心得记录**

**1.通过实验中的JavaFX绘图程序，我进一步了解了如何使用JavaFX来创建基本的图形界面，并通过用户输入来进行交互；**

**2.在实验中，我通过使用枚举类来表示有限的一组常量，对枚举类有了更深的认识；**

**3.通过实验中设计的学生和课程的继承关系，我加深了对继承概念的理解，并学会了如何使用继承来避免重复编写相似的代码，提高了代码的可重用性和扩展性；**

**4.通过此次实验，我对Java编程的图形界面、枚举和继承等概念有了更深入的理解，提高了编程技能和实践能力。**