****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验3

**姓名：宋泽涛**

**学号：25120222201292**

**学院：信息学院**

**专业：软件工程**

**完成时间：2024/3/14**

1. **实验目的及要求**

**实验目的：**

* 练习控制结构
* 熟悉面向对象封装特性
* 熟悉简单的JavaFX图形界面

**实验要求：**

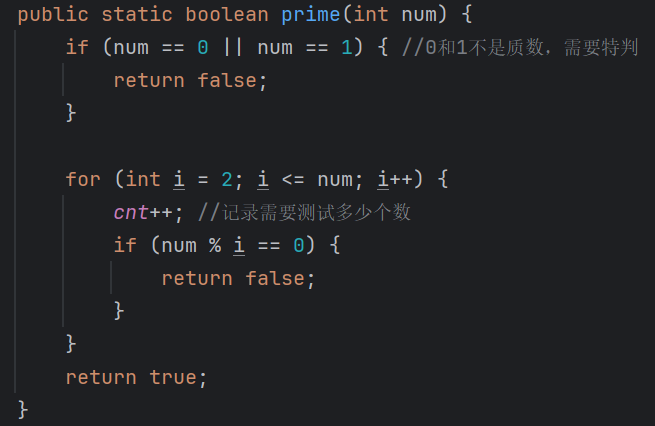
* 下周实验课将工程文件和实验报告打包上传到FTP

**二、实验题目及实现过程**

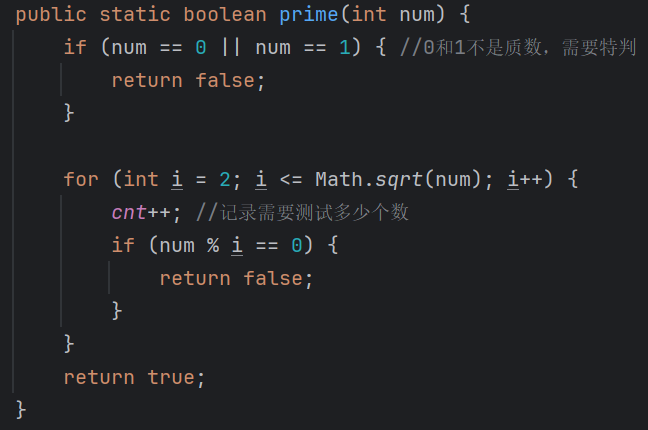
1. 基本题目：
2. （素数）如果某个正整数只能由 1 和自身整除，则这个整数就被称为素数。例如，2、3、5、7 是素数，而 4、6、8、9 不是。根据定义，数字 1 不为素数。
   1. 编写一个方法，它判断一个数是否为素数。
   2. 在程序中使用这个方法，显示小于 10000 的全部素数。为了找出所有不超过 10000 的素数，需要测试多少个数？
   3. 开始时，可能会想到要确定某个数n是否为素数，需进行测试的次数最多为n/2 次，其实只需最多测试 n 的平方根次即可。重新编写这个程序，并以这两种方式运行它。

**实现过程**

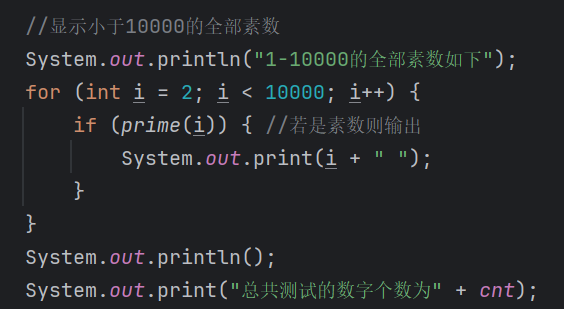
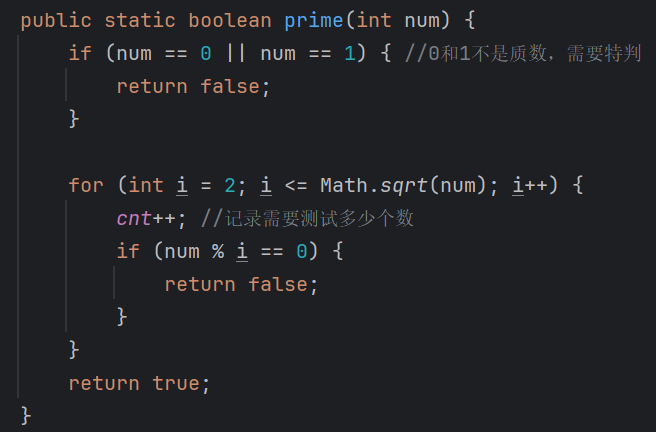
1. **Ta&c实现思路：对题目要求进行分析，题目要求我们在程序中接收某个数字，然后判断它是否为素数。那么，我们可以编写一个boolean类型的方法，这个方法接收一个数字变量，然后在方法内部经过某些手段判断它是否为素数，若为素数，则返回true，否则返回false。思路有了，我们如何去实现呢？根据素数的定义：如果某个正整数只能由1和自身整除，则这个整数就是素数。于是乎，对于每次程序的输入 n ，我们可以通过一个for循环， i 从2开始一直枚举到 n-1 ，每次枚举判断 (n % i == 0) 这个表达式是否为真。若为真则表示这个数字 n 可以被除了1和自身外的数字整除，那么它就不是素数，直接返回false。若枚举完所有的元素，表达式 (n % i == 0) 均为假，那么这个数就是素数，返回true。代码如下：**



1. **考虑优化：实际上可以证明，我们用 i 进行枚举的时候，并不需要从2枚举到 n-1 ，只需要枚举到sqrt(n)即可。这样子就可以大幅减少枚举的时间复杂度。代码如下：**

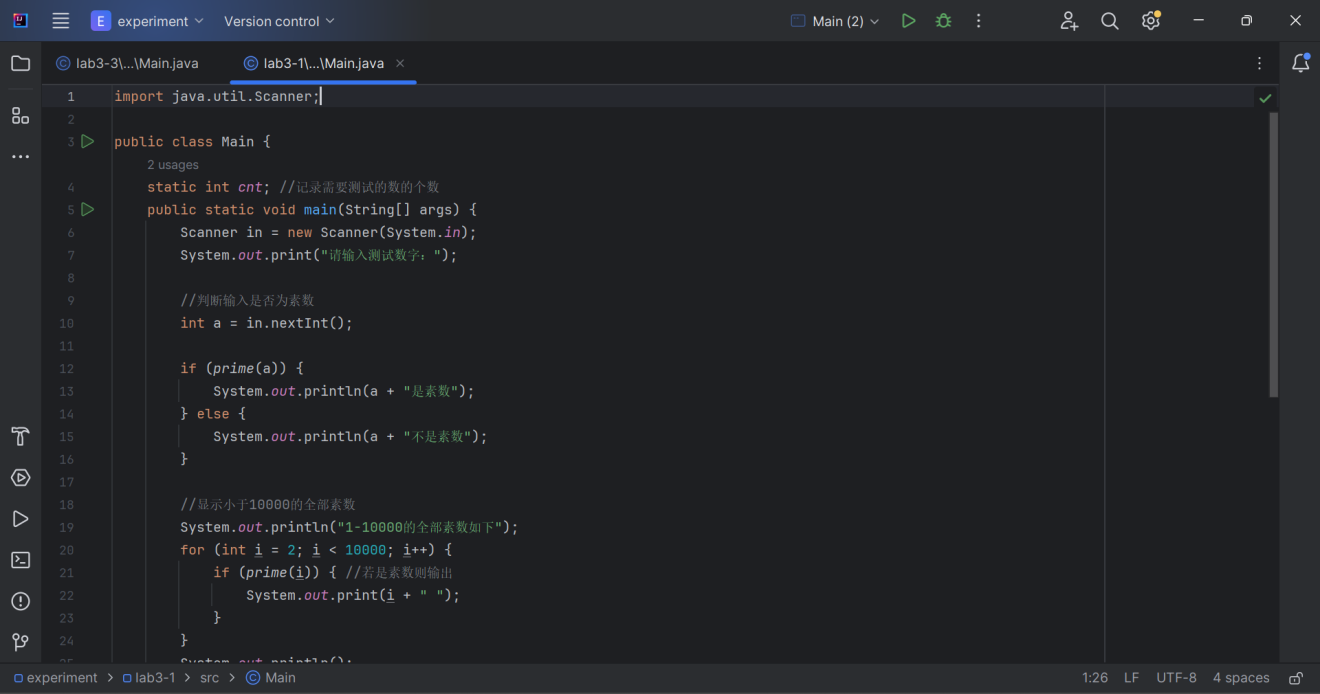


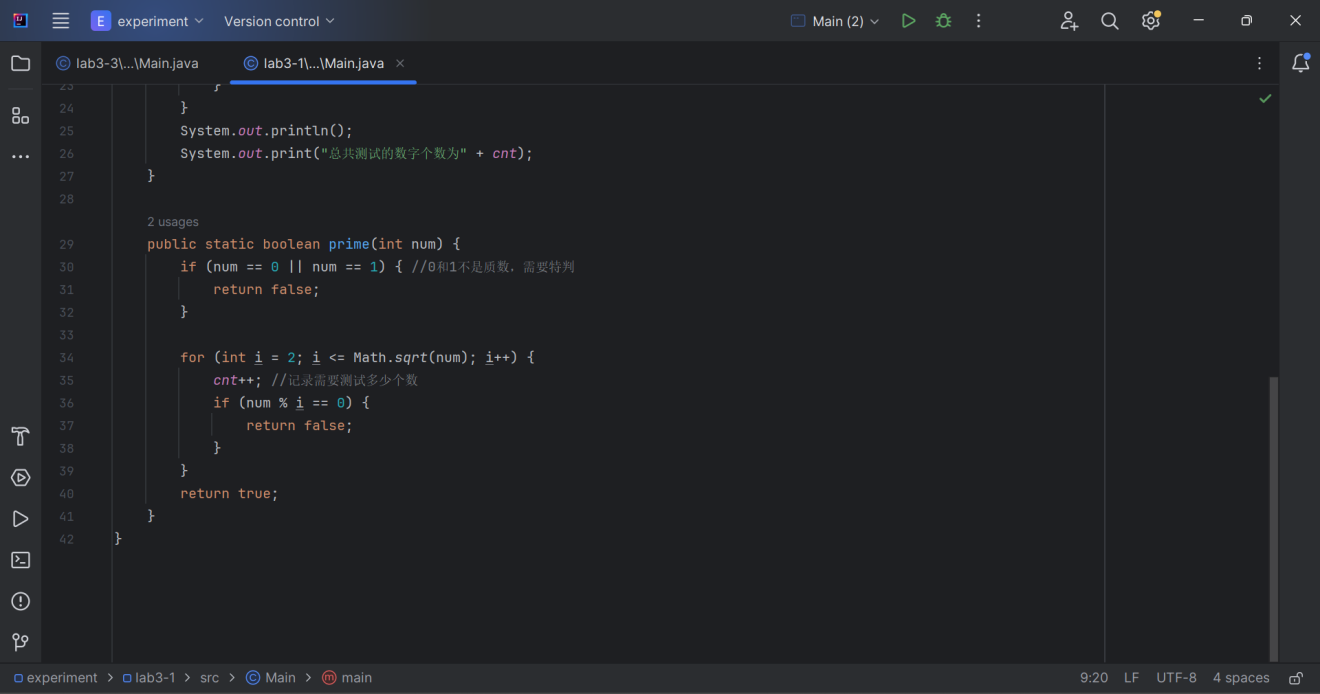
1. **Tb实现思路：有了以上判断素数的方法后，为了找出不超过10000的素数，只需要从2开始枚举，每次枚举调用一次判断素数的方法，若为素数则计数器++即可。代码如下：**



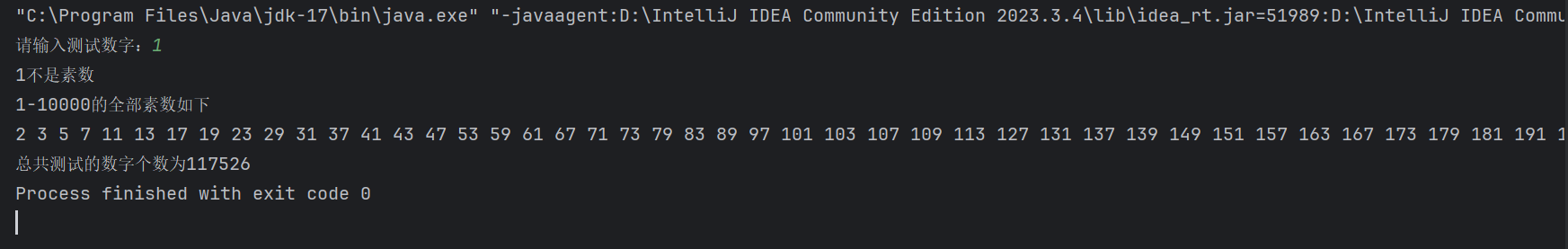
1. **最终执行结果与代码截图**

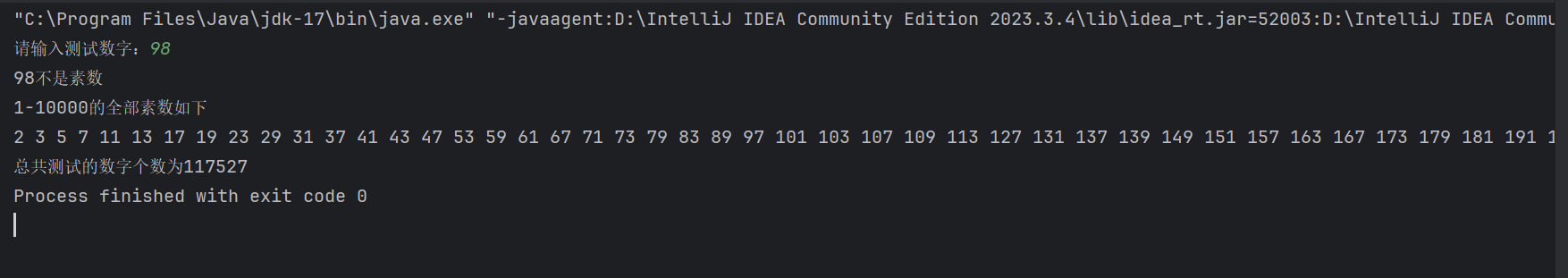
**代码截图如下**

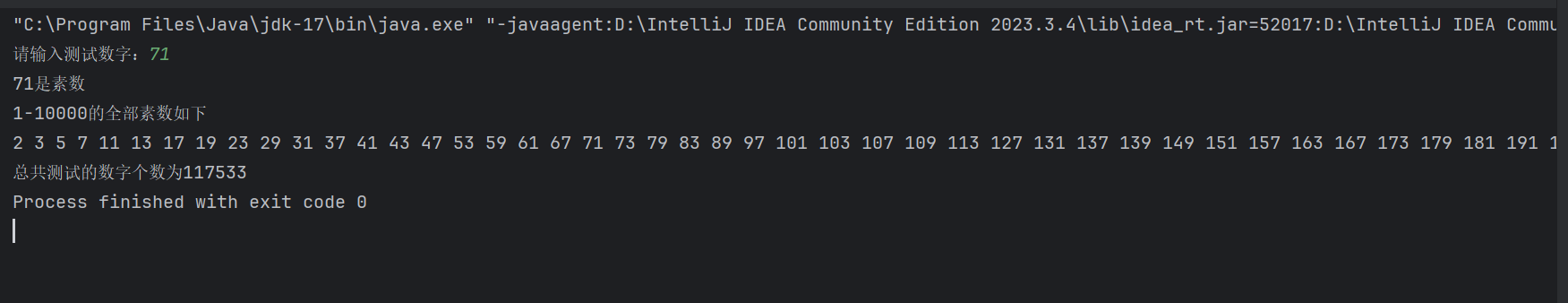
****

****

**执行结果截图**

****

****

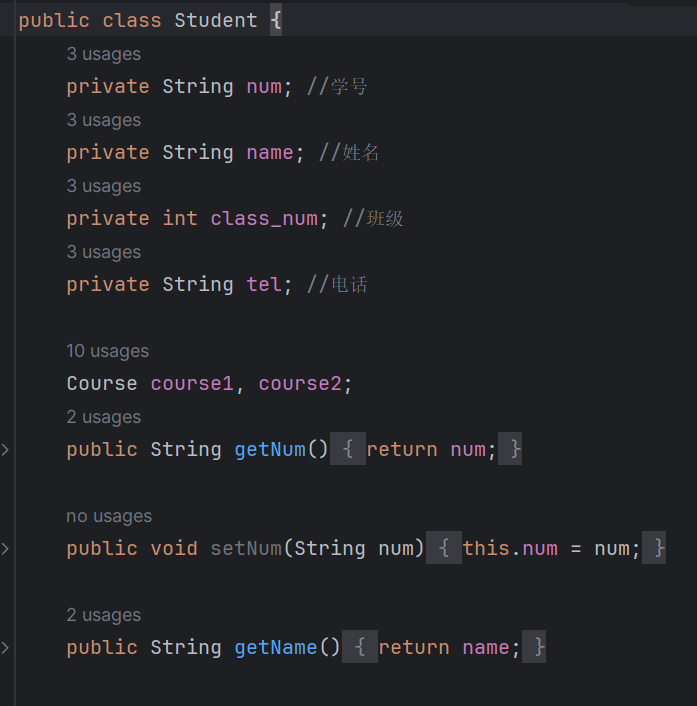
****

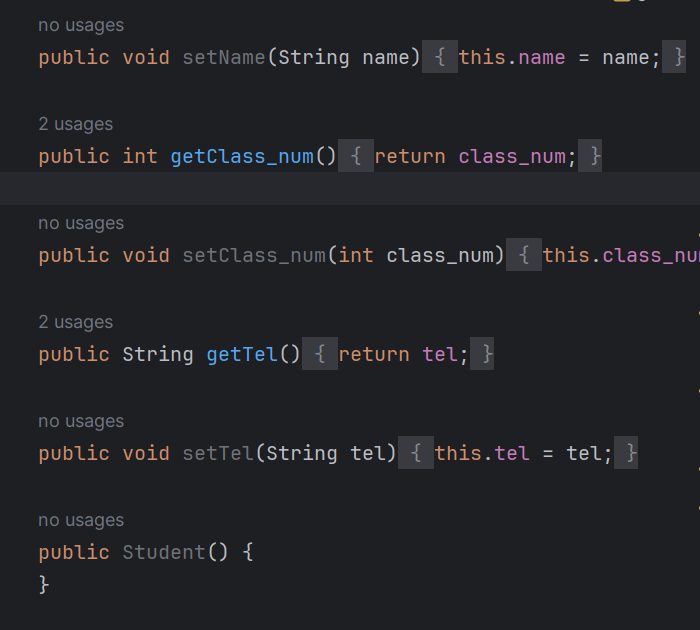
1. 按照要求编程实现学生一个简答的选课系统：
   1. 定义学生类，学生基本属性包括：学号、姓名、班级、电话，电话可以为空；（设计重载构造函数）
   2. 定义课程类，课程基本信息包括：课程编号、课程名称；
   3. 定义Test类
      1. 提示用户输入2个学生信息，并根据用户输入创建2个学生类对象；
      2. 提示用户输入2门课程信息，并根据用户输入创建2个课程类对象；
      3. 输出选课结果以及每个学生对应课程的成绩。

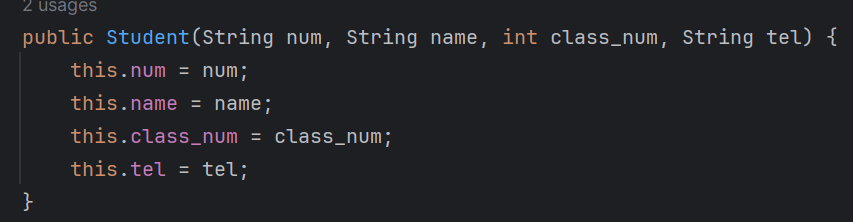
**实现过程**

1. **实现思路：分析题目需求，我们需要定义两个类：学生类与课程类。在学生类中定义基本属性包括：学号、姓名、班级、电话等；在课程类中定义基本信息包括：课程编号、课程名称。在定义这些类型时需要注意要将其私有化，封装好，对外提供get&set方法。根据题目要求，我们还要设计重载构造函数，对于这个需求我们只需要定义类相关属性的有参构造器即可。具体学生类与课程类的实现如下。**

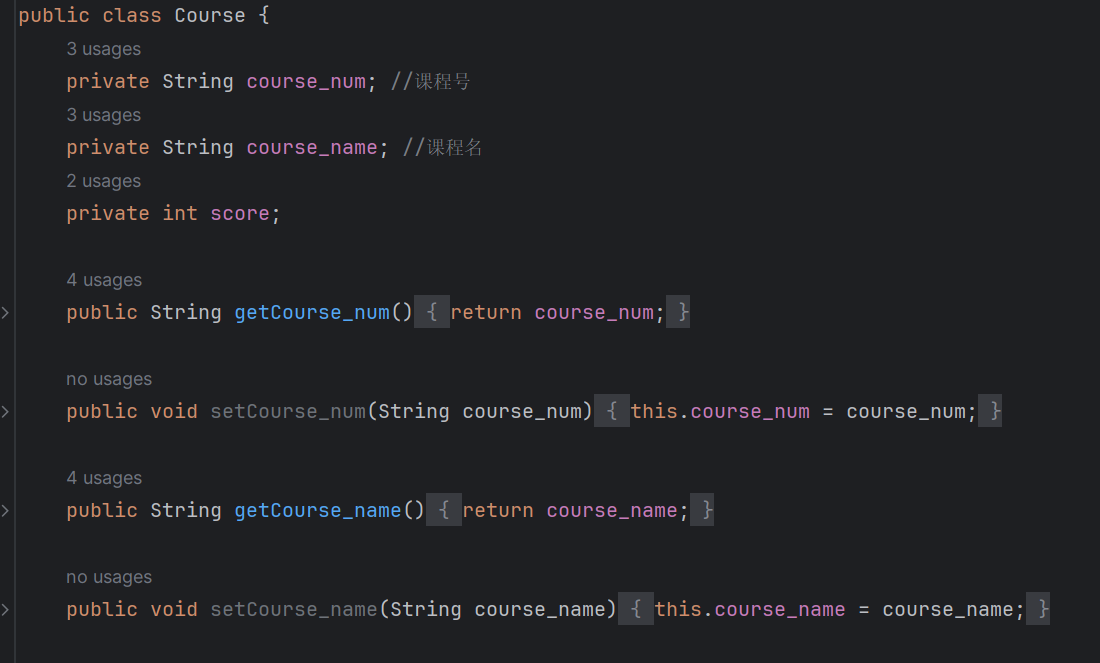
**学生类如下**







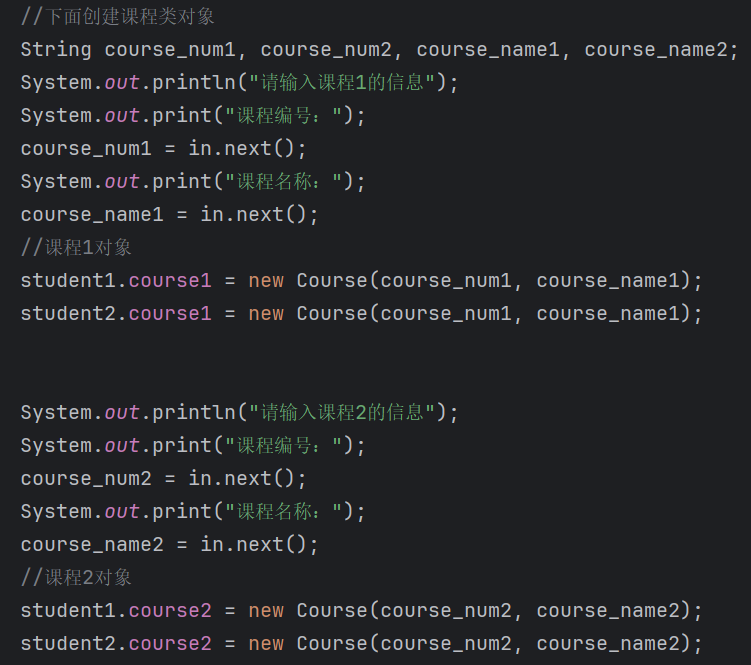
**课程类如下**





**定义好这两个类之后，我们可以在test类中编写相关程序，对这两个类进行实例化，再处理选课以及课程分数问题。**

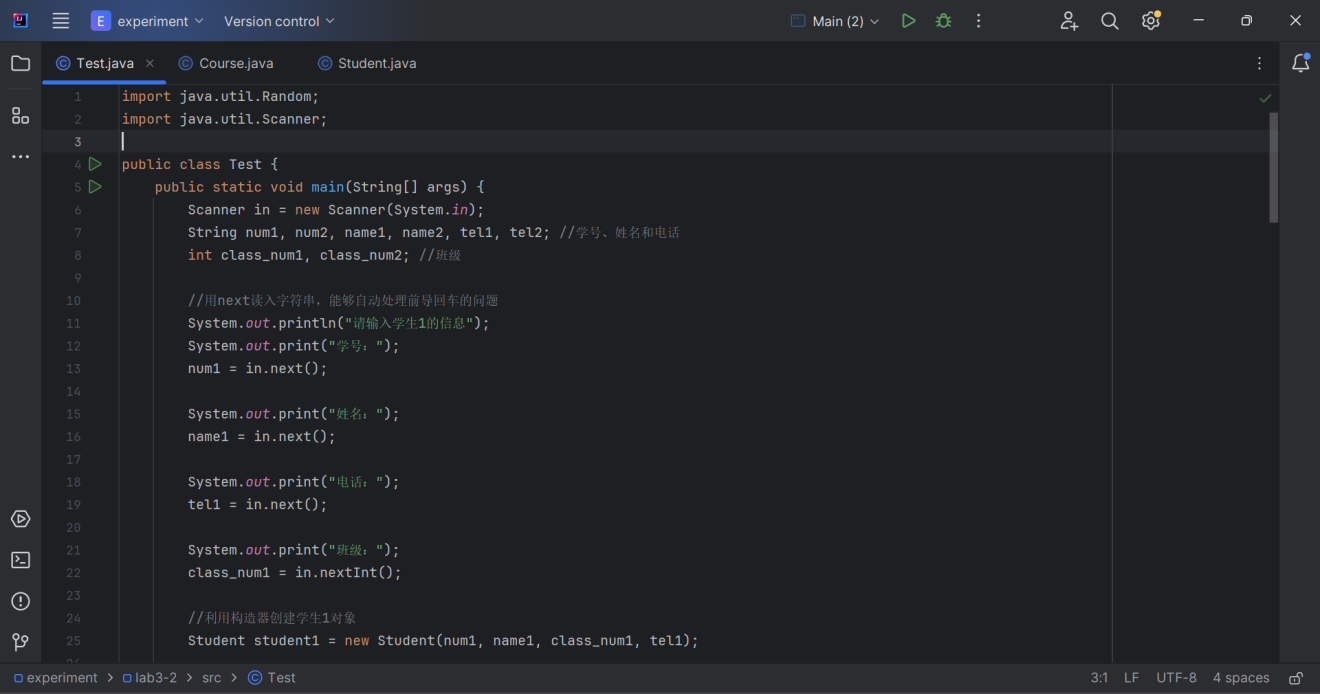
**在主程序中，我们通过读入学生和课程的信息对两个学生对象和两个课程对象进行初始化，然后通过random类创建random对象产生随机数，用来得到学生对应课程的分数。那么我们如何来存放这个分数呢？我们可以在学生类中定义课程类，再通过主函数调用课程类的有参构造器实例化学生类中的课程类对象，将产生的随机数存放在课程类对象的score成员里即可。具体实现代码如下：**

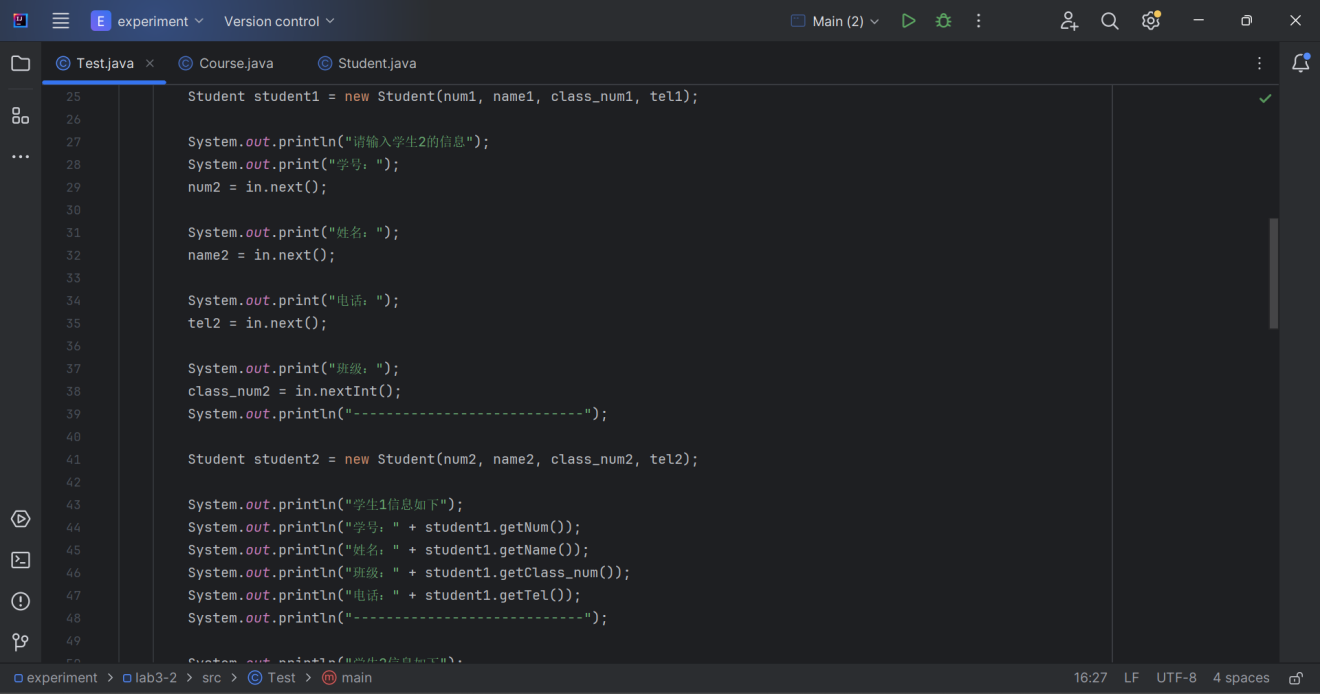
****

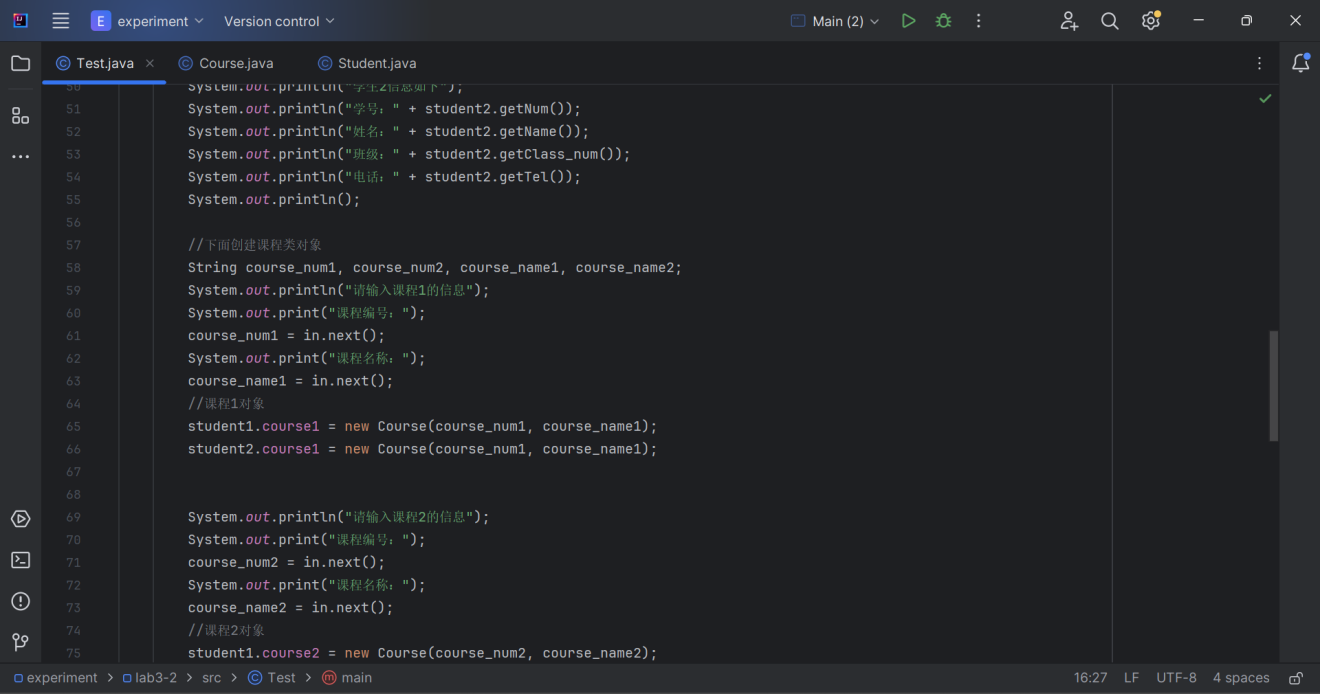
****

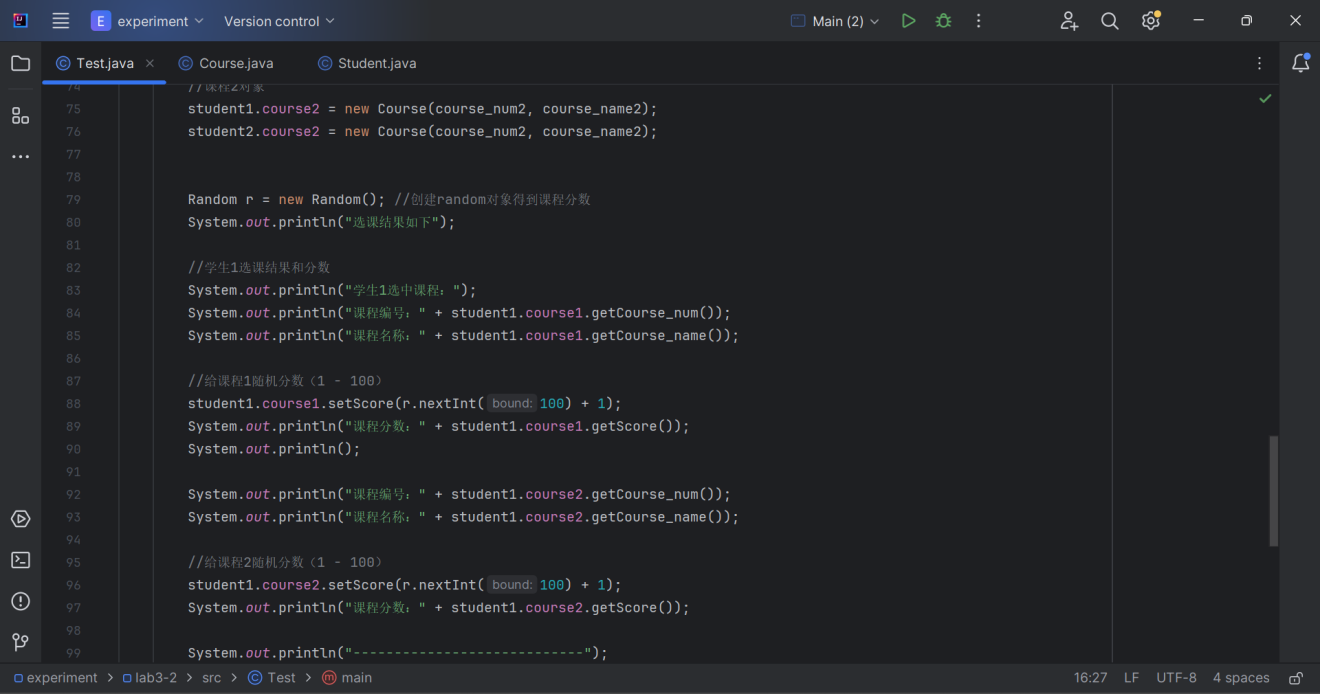
1. **最终执行结果与代码截图**

**代码截图如下**

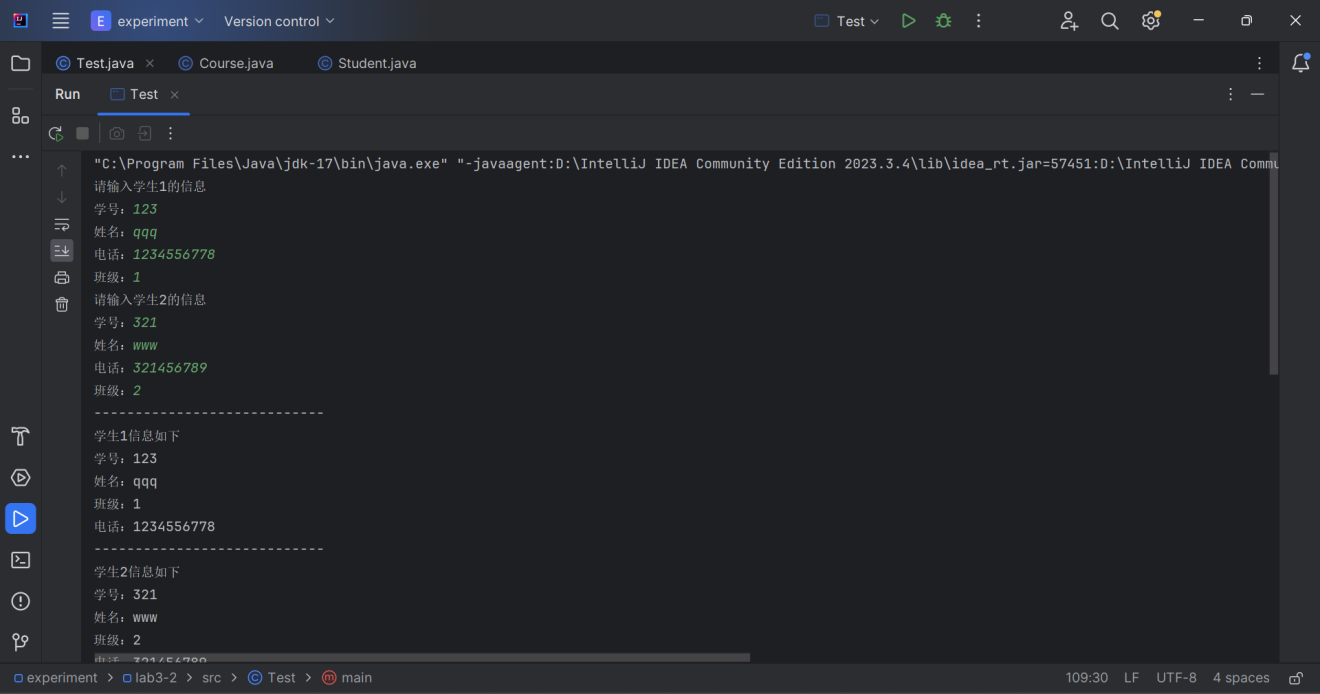
****

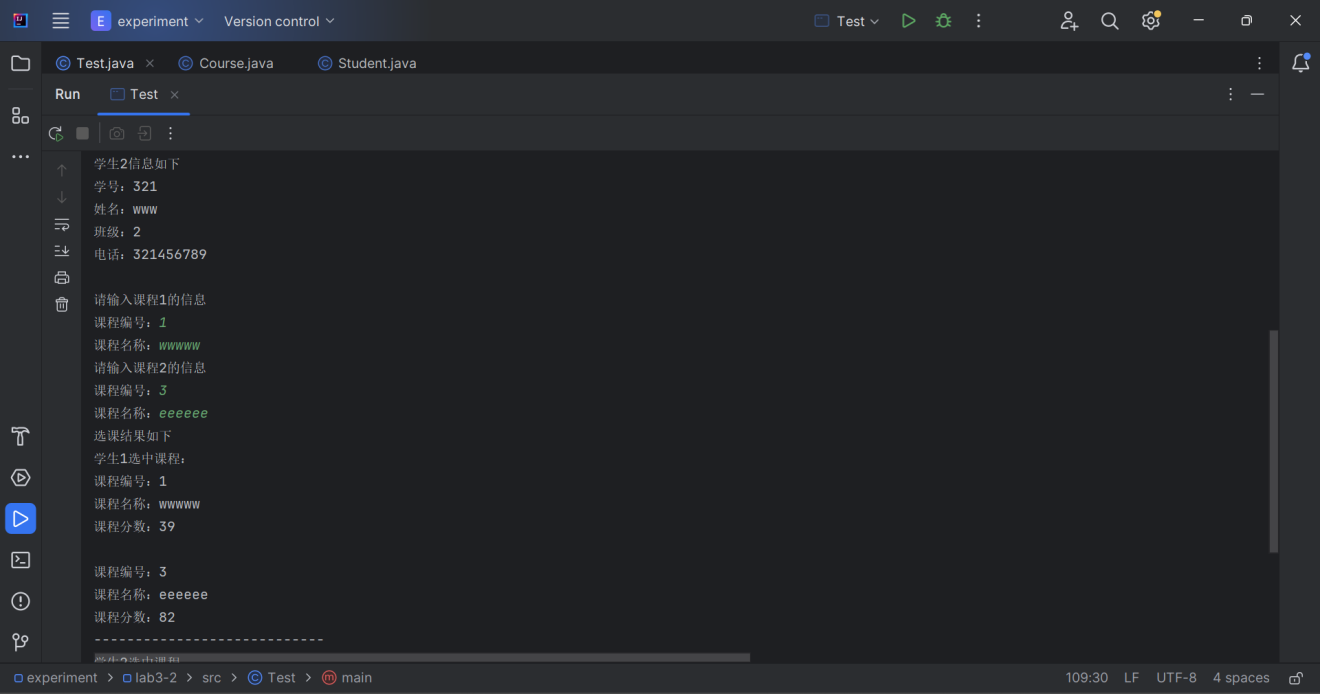
****

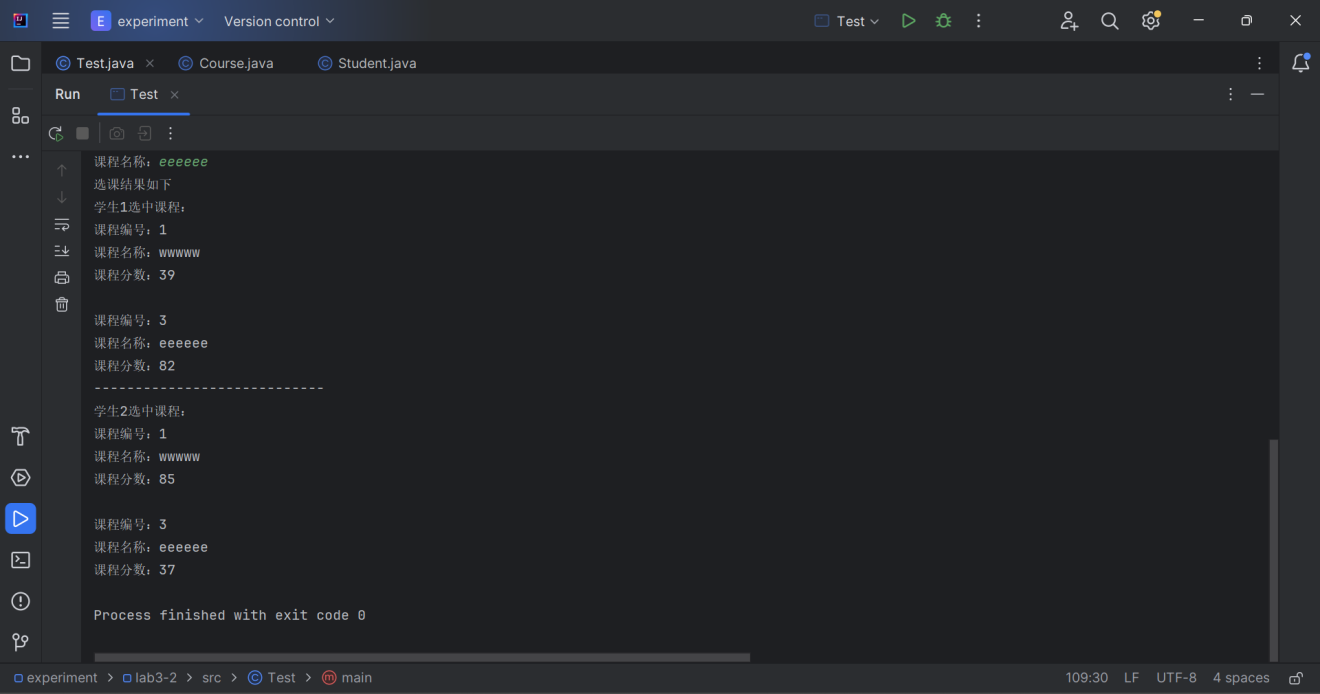
****

****

**执行结果如下**

****

****

****

1. （计算机辅助教学）计算机在教育领域的使用被称为“计算机辅助教学”（CAI） 。编写一个程序，以帮助小学生学习乘法。利用一个 SecureRandom 对象来产生两个一位正整数。程序需向用户提示一个问题、例如：

How much is 6 times 7?

然后、学生应输入答案。接下来，需检查答案的正确性。如果回答正确，则显示消息“Very good！”并给出另一个乘法问题。如果答错，则应显示消息“No. Please try again。”，然后让学生回答同个问题、直到答对为止。产生每一个新问题时，应使用一个独立的方法。这个方法应在程序开始执行时调用一次，然后在学生正确回答问题后再调用一次。

1. 扩展题目（选做）:
2. (CAI：降低学生的疲劳感）CAI 所面临的一个问题是学生的疲劳感。通过变换计算机的响应，使学生保持注意力，可以降低疲劳感。修改练习题1中的程序，为每一个答案附带各种评语。针对回答正确的评语有：

Very good!

Excellent!

Nice work!

Keep up the good work!

针对回答错误的评语有：

No. Please try again.

Wrong. Try once more.

Don't give up!

No. Keep trying.

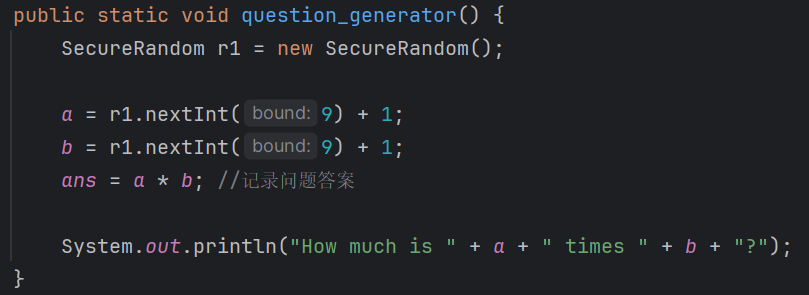
利用随机数生成方法选择 1~4 中的一个数，并用它来为每个正确或错误的答案选择 4 种可能的评语之一。利用一条 switch 语句来提供这些评语。

1. (CAI：监督学生的表现）更复杂的 CAI 系统可以监督某段时间学生的表现。是否进入一个新的学习驻题，是以学生成功完成了前面的主题为基础的。修改练习题 3中的程序，统计学生答对和答错的次数。学生回答完 10 个问题后，程序应计算正确率（百分比）。如果正确率小于 75%，则显示“Please ask your teacher for extra help.”，然后重置程序，让另一名学生答题；如果正确率超过 75%，则显示“Congratulations, you are ready to go to the next level！”，并重置程序，让另一名学生答题。

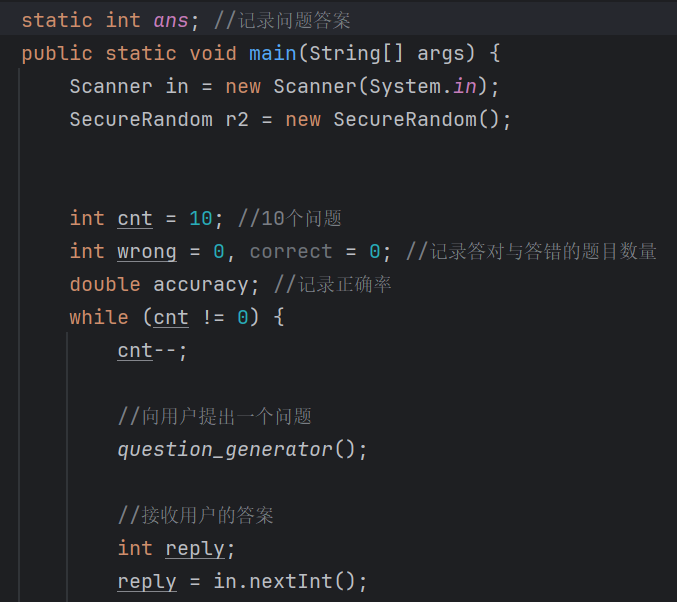
**实现过程**

1. **实现思路：根据基础题目的要求，要完成这个程序，我们只需要一个类。在主方法中，我们首先需要给用户提示问题，让用户输入问题的答案。至于如何提出问题，我们需要编写一个question\_generator方法，用于生成相关问题。对于每个产生的问题，我们还需要存该问题的答案。这一点我们可以通过全局静态变量来实现。对于拓展题目，我们只需要在基础题目的基础上将用户的答题反馈通过switch分支语句实现即可。然后我们再定义一个accuracy变量，用来存放用户的正确率。至于如何统计正确率，我们只需要通过循环去实现即可。**
2. **具体实现细节**

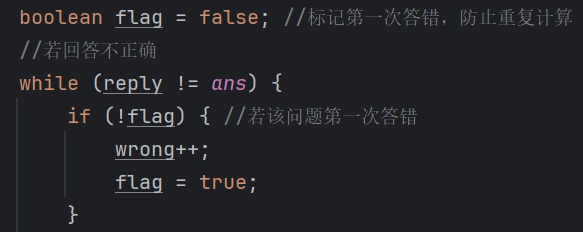
**我们首先来看question\_generator方法**

****

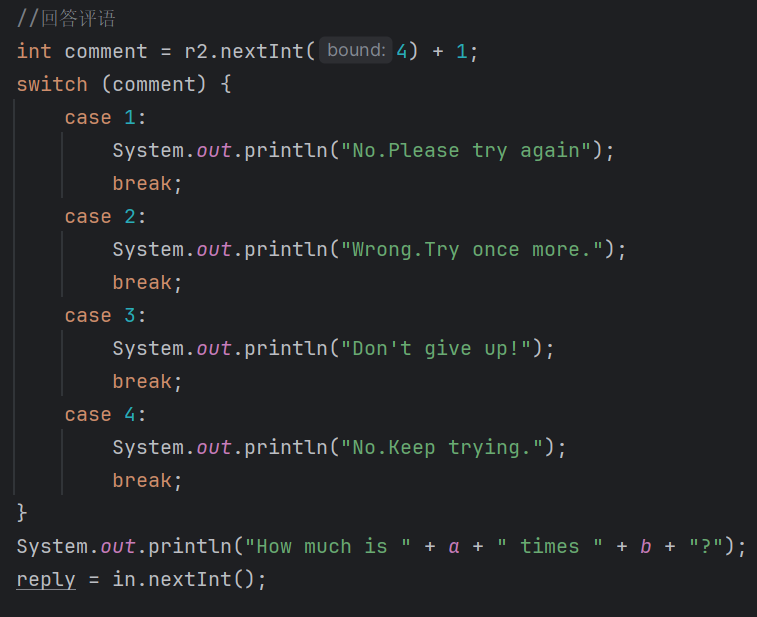
**在方法内部，我们创建了一个securerandom对象，用于产生1 - 9的随机数，然后将问题的答案记录到静态变量ans中。我们再回到主方法中：**

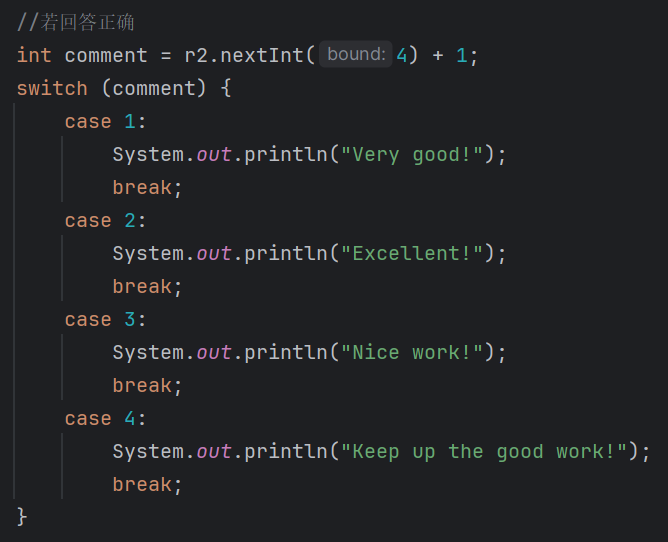
****

**在这里我们用cnt记录总共十个问题，wrong记录答错题目数量，reply接收用户的答案。但这里存在一个问题，我们在循环中如何判断用户答错问题的个数不被重复计算？（用户可能多次答错同一个问题）我们可以用一个boolean变量进行标记。**

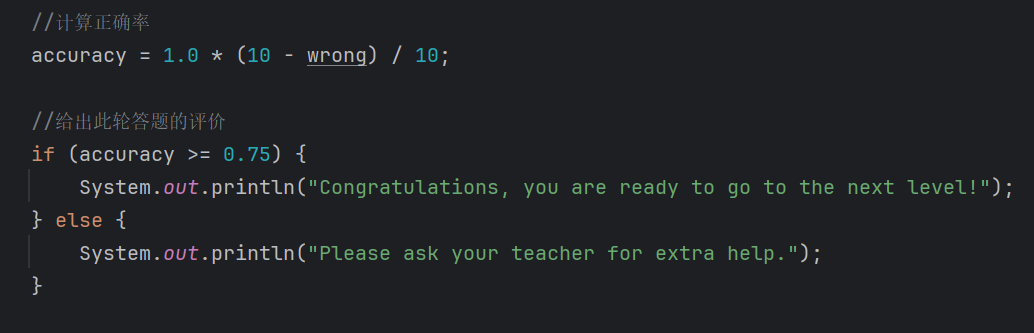
****

**在之后使用switch分支语句对用户给出反馈即可。**

****

****

**计算正确率**

****

1. **实验总结与心得记录**
2. **通过本次实验，我对java的控制结构、面向对象封装特性有了更深入的理解和熟练度的提升；**
3. **选课系统的设计让我体会到了面向对象编程的灵活性和可扩展性，但此次代码写的较为冗余，可进一步思考如何优化实现；**
4. **在扩展题目中，通过switch分支语句和利用随机数生成不同的评语，使我对java特性有了更深的理解。**