

# 廈門大學



## 信息学院软件工程系

### 《JAVA 程序设计》实验报告

#### 实验 3

姓名：黄勛

学号：22920212204392

学院：信息学院

专业：软件工程

完成时间：2023.3.15

## 一、实验目的及要求

- 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

## 二、实验题目及实现过程

实验环境: Windows 10 21H2、jdk17

(基本题目) 题目 1:

### (一) 实验题目

(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。

- 编写一个方法, 它判断一个数是否为素数。
- 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?
- 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。

### (二) 实现过程 (prime.java)

思路:

- 判断一个数是否为素数的方法是, 从 2 到该数-1 的范围内, 依次将该数进行取模运算, 如果能被某个数整除, 则说明该数不是素数, 否则就是素数。需要特别判断该数是否小于 2。
- 使用上述判断素数的方法, 遍历从 2 到 9999 的所有数, 判断每个数是否是素数, 是则输出该数。需要测试 9999 个数。
- 对于一个数  $n$ , 可以使用 2 到  $n$  的平方根之间的所有数进行取模运算, 如果都不能整除, 则该数为素数。同样需要特别判断该数是否小于 2。相比测试  $n/2$  次, 该方法效率更高。

代码：

该程序只有一个类 prime，其成员都是私有的静态方法，主要方法包括 is\_Prime\_by\_half() 和 is\_Prime\_by\_sqrt()，分别用来判断一个数是否为质数，其中前者采用了从 2 到  $n/2$  逐一判断的方法，后者采用了从 2 到  $\sqrt{n}$  逐一判断的方法。

```
3      // 判断 n 是否为质数（方法一：从 2 到 n/2 逐一判断）
      1 个用法
4      private static boolean is_Prime_by_half(int n)
5      {
6          if(n == 1)
7              return false; // 1 不是质数
8          for(int i = 2; i <= n / 2; i++)
9          {
10             if(n % i == 0)
11                 return false; // 只要有一个因子，就不是质数
12             }
13             return true; // 未找到因子，是质数
14     }
```

```
16     private static boolean is_Prime_by_sqrt(int n)
17     {
18         if(n == 1)
19             return false;
20         for(int i = 2; i <= Math.sqrt(n); i++)
21         {
22             if(n % i == 0)
23                 return false;
24             }
25             return true;
26     }
```

程序的主方法 main() 中，分别使用这两种方法判断小于 10000 的质数，并计算出每种方法判断的用时。在程序的输出中，首先输出了从  $n/2$  方法和  $\sqrt{n}$  方法分别计算得到的质数和用时，最后输出了总质数和总用时。

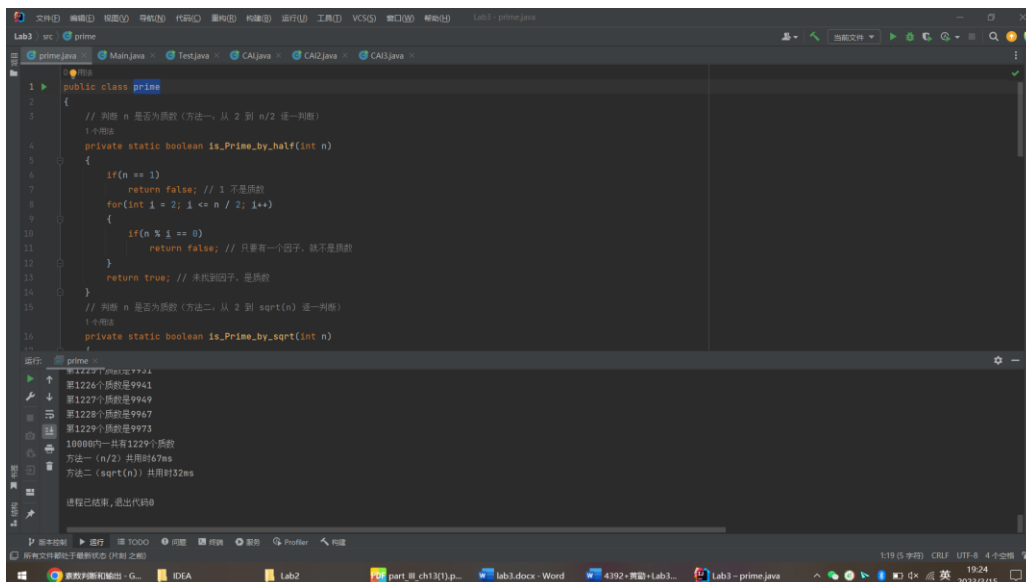
```

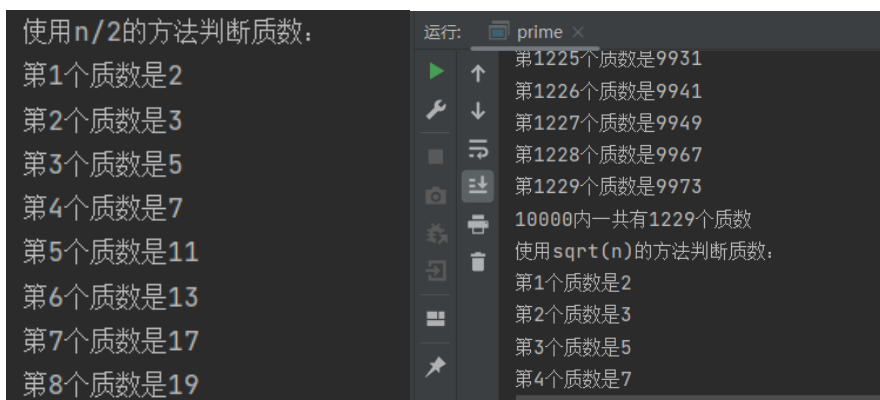
29      System.out.println("使用n/2的方法判断质数：");
30      int prime_num = 0; // 统计质数个数
31      long startTime_1 = System.currentTimeMillis(); // 记录开始时间
32      for(int i = 1; i < 10000; i++)
33      {
34          if(is_Prime_by_half(i))
35          {
36              System.out.printf("第%d个质数是%d\n", ++prime_num, i);
37          }
38      }
39      long endTime_1 = System.currentTimeMillis(); // 记录结束时间
40      long totalTime_1 = endTime_1 - startTime_1; // 计算用时
41      System.out.printf("10000内一共有%d个质数\n", prime_num);
42
43      System.out.println("使用sqrt(n)的方法判断质数：");
44      prime_num = 0;
45      long startTime_2 = System.currentTimeMillis();
46      for(int i = 1; i < 10000; i++)
47      {
48          if(is_Prime_by_sqrt(i))
49          {
50              System.out.printf("第%d个质数是%d\n", ++prime_num, i);
51          }
52      }
53      long endTime_2 = System.currentTimeMillis();
54      long totalTime_2 = endTime_2 - startTime_2;
55      System.out.printf("10000内一共有%d个质数\n", prime_num);
56      System.out.println("方法一 (n/2) 共用时" + totalTime_1 + "ms");
57      System.out.println("方法二 (sqrt(n)) 共用时" + totalTime_2 + "ms");

```

### (三) 过程截图

最终结果以及最终运行时间比较，最后可以得出用 sqrt(n)的方法可以快不少。





```
使用n/2的方法判断质数：
第1个质数是2
第2个质数是3
第3个质数是5
第4个质数是7
第5个质数是11
第6个质数是13
第7个质数是17
第8个质数是19

运行: prime x
第1225个质数是9931
第1226个质数是9941
第1227个质数是9949
第1228个质数是9967
第1229个质数是9973
10000内一共有1229个质数
使用sqrt(n)的方法判断质数：
第1个质数是2
第2个质数是3
第3个质数是5
第4个质数是7
第5个质数是11
```

## 题目 2:

### (一) 实验题目

编写一个应用程序，在 Canvas 上填充 10 个不同颜色的圆。位置、大小和颜色随机。

### (二) 实现过程 (Lab3\_2 工程项目)

思路：1) 创建一个 JavaFX 应用程序。可以通过创建一个继承自 `javafx.application.Application` 类的 Java 类来实现。在 `start()` 方法中创建一个 `Scene` 和一个 `Canvas` 对象并将它们添加到 `Stage` 上。

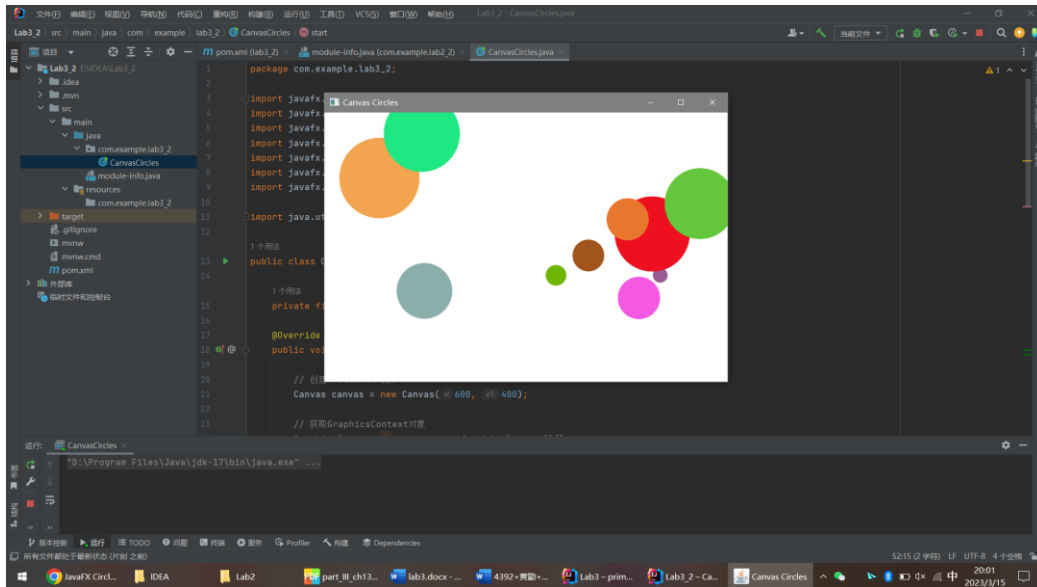
2) 在 `Canvas` 上绘制 10 个不同颜色的圆。这里可以使用 `Canvas` 的 `GraphicsContext` 对象绘制圆。为了使每个圆的位置、大小和颜色随机，需要使用 Java 的随机数生成器类（例如 `java.util.Random`）来生成随机值。

3) 在绘制圆之前，需要确保 `Canvas` 的宽度和高度足够大，以容纳所有圆。可以使用 `Scene` 的宽度和高度来计算 `Canvas` 的大小，并将其设置为 `Scene` 的根节点。

4) 对于每个圆，使用随机数生成器生成随机位置、大小和颜色。然后，使用 `GraphicsContext` 的 `fillOval()` 方法绘制圆。

5) 最后，运行 JavaFX 应用程序并查看结果。

### (三) 过程截图



准备工作:

```
17      @Override
18      public void start(Stage stage) {
19
20          // 创建一个Canvas对象
21          Canvas canvas = new Canvas(600, 400);
22
23          // 获取GraphicsContext对象
24          GraphicsContext gc = canvas.getGraphicsContext2D();
25
26          // 创建一个Pane对象，将Canvas对象添加到Pane中
27          Pane pane = new Pane(canvas);
28
29          // 创建Scene对象，并将Pane对象设置为根节点
30          Scene scene = new Scene(pane, 600, 400);
31
32          // 设置窗口标题
33          stage.setTitle("Canvas Circles");
34
35          // 将Scene对象设置到Stage中
36          stage.setScene(scene);
37      }
```

生成圆的主要代码:

```
41 // 随机生成10个圆
42 for (int i = 0; i < CIRCLE_COUNT; i++) {
43     // 随机生成圆心坐标和半径大小
44     double x = random.nextDouble() * canvas.getWidth();
45     double y = random.nextDouble() * canvas.getHeight();
46     double radius = random.nextDouble() * 50 + 10;
47
48     // 随机生成颜色
49     Color color = Color.rgb(random.nextInt( bound: 256), random.nextInt( bound: 256), random.nextInt( bound: 256));
50
51     // 在Canvas上绘制圆
52     gc.setFill(color);
53     gc.fillOval( x - radius, y - radius, radius * 2, radius * 2);
54 }
```

### 题目 3:

#### (一) 实验题目

按照要求编程实现学生一个简答的选课系统:

- a) 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)
- b) 定义课程类, 课程基本信息包括: 课程编号、课程名称;
- c) 定义 Test 类
  - i. 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;
  - ii. 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;
  - iii. 输出选课结果以及每个学生对应课程的成绩。

#### (二) 实现过程 (Test.java)

这个选课系统需要设计三个类: 学生类(Student)、课程类(Course)和测试类(Test)。

学生类(Student):

属性: 学号(id)、姓名(name)、班级(className)、电话(phoneNumber)

构造函数: 重载构造函数, 可以根据不同的属性值创建学生对象

方法: 提供 getXX 方法, 方便其他类使用学生对象的属性

课程类(Course):

属性: 课程编号(courseCode)、课程名称(courseName)

构造函数: 重载构造函数, 可以根据不同的属性值创建课程对象

方法: 提供 getXX 方法, 方便其他类使用课程对象的属性

测试类(Test):

方法: 包括程序入口 main 方法, 用于测试学生选课系统

### 实现步骤:

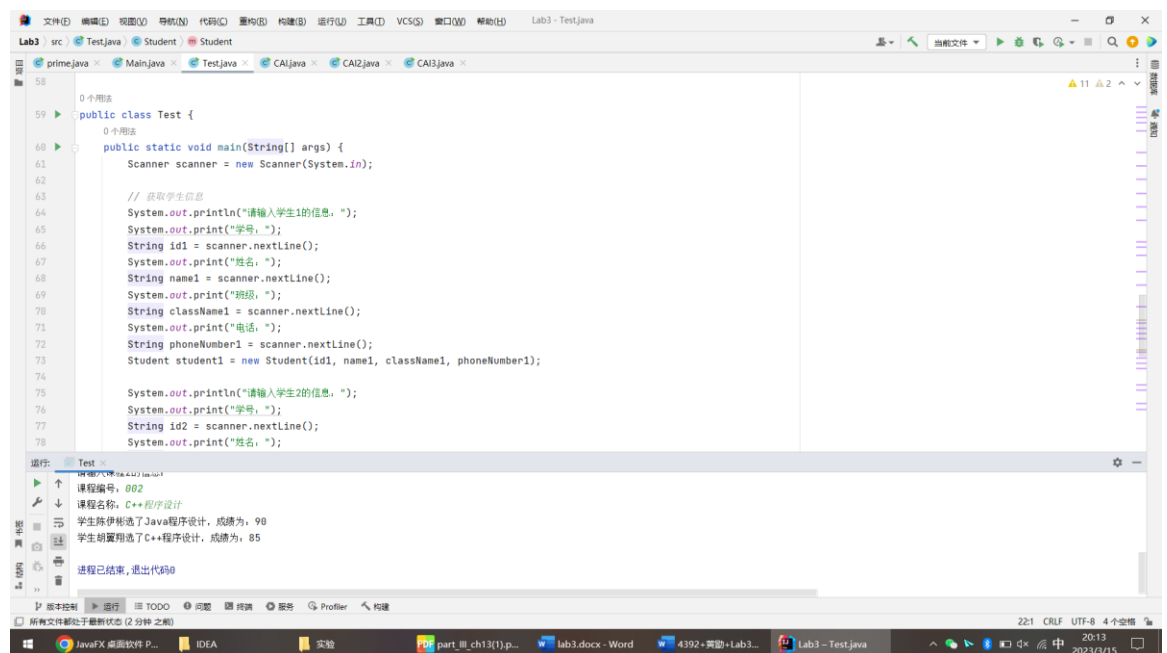
提示用户输入学生信息, 并根据用户输入创建两个学生类对象

提示用户输入课程信息, 并根据用户输入创建两个课程类对象

输出选课结果以及每个学生对应课程的成绩

学生类和课程类是独立的类, 而测试类则调用了学生类和课程类的对象来实现选课系统的功能。在测试类中, 用户输入的信息可以通过 Scanner 类来获取, 然后根据输入创建对应的学生和课程对象。最后, 根据用户的选择, 输出选课结果以及每个学生对应课程的成绩。

### (三) 过程截图



The screenshot displays an IDE window titled 'Lab3 - Test.java'. The code defines a 'Test' class with a 'main' method that uses a 'Scanner' to collect user input for two students. The output console shows the program's execution, including prompts for student information and the resulting course selection and scores for two students.

```
58  
59 public class Test {  
60     public static void main(String[] args) {  
61         Scanner scanner = new Scanner(System.in);  
62  
63         // 获取学生信息  
64         System.out.println("请输入学生1的信息: ");  
65         System.out.print("学号: ");  
66         String id1 = scanner.nextLine();  
67         System.out.print("姓名: ");  
68         String name1 = scanner.nextLine();  
69         System.out.print("班级: ");  
70         String className1 = scanner.nextLine();  
71         System.out.print("电话: ");  
72         String phoneNumber1 = scanner.nextLine();  
73         Student student1 = new Student(id1, name1, className1, phoneNumber1);  
74  
75         System.out.println("请输入学生2的信息: ");  
76         System.out.print("学号: ");  
77         String id2 = scanner.nextLine();  
78         System.out.print("姓名: ");
```

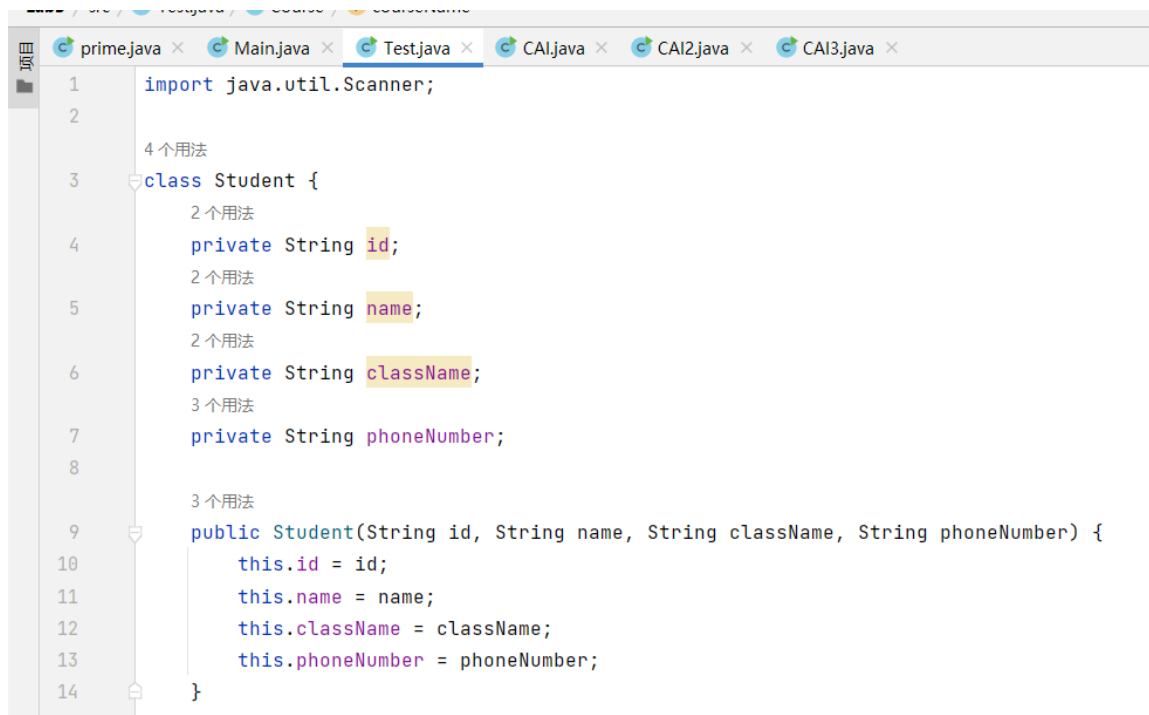
运行: Test  
课程编号: 002  
课程名称: C++ 程序设计  
学生陈伊彬选了Java程序设计, 成绩为: 90  
学生胡翼翔选了C++程序设计, 成绩为: 85  
进程已结束, 退出代码0

### 完整流程展示:






## Student 类:

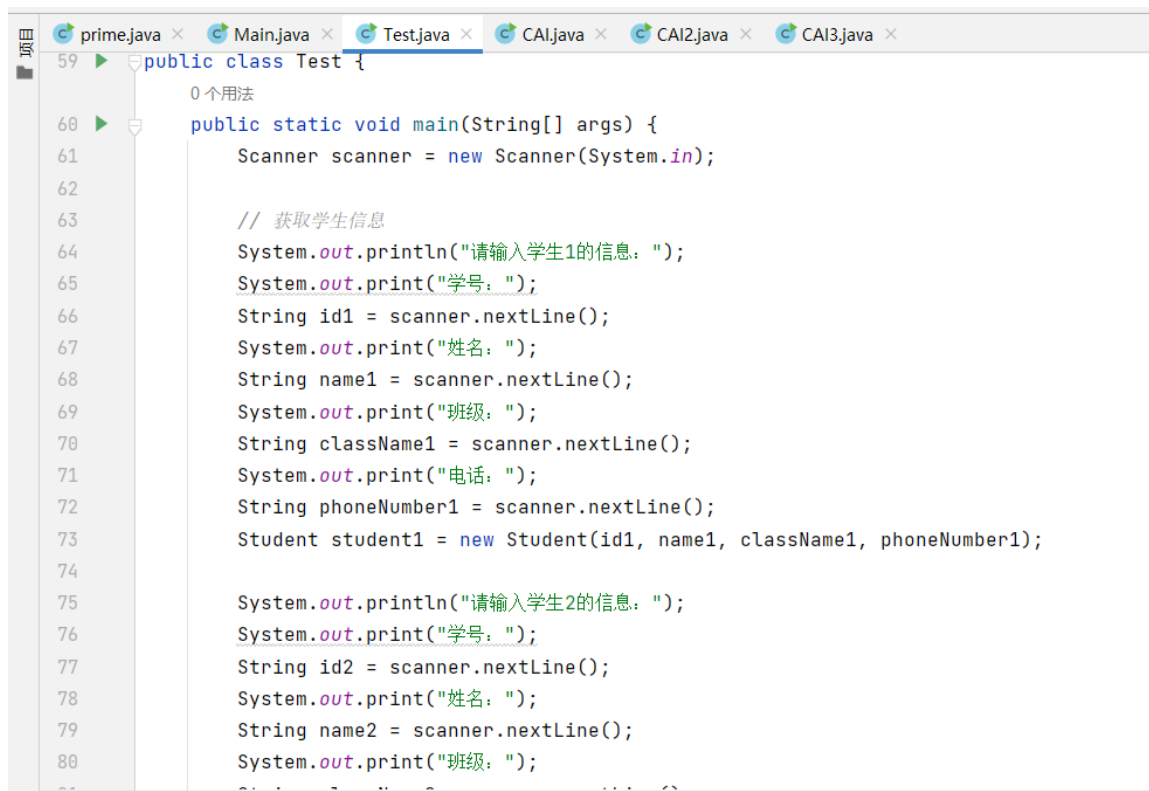


## Class 类:



```
41 class Course {  
42     private String courseCode;  
43     private String courseName;  
44  
45     public Course(String courseCode, String courseName) {  
46         this.courseCode = courseCode;  
47         this.courseName = courseName;  
48     }  
49  
50     public String getCourseCode() {  
51         return courseCode;  
52     }  
53  
54     public String getCourseName() {  
55         return courseName;  
56     }  
57 }
```

## Test 类:



```
59 public class Test {
    0 个用法
60 public static void main(String[] args) {
61     Scanner scanner = new Scanner(System.in);
62
63     // 获取学生信息
64     System.out.println("请输入学生1的信息: ");
65     System.out.print("学号: ");
66     String id1 = scanner.nextLine();
67     System.out.print("姓名: ");
68     String name1 = scanner.nextLine();
69     System.out.print("班级: ");
70     String className1 = scanner.nextLine();
71     System.out.print("电话: ");
72     String phoneNumber1 = scanner.nextLine();
73     Student student1 = new Student(id1, name1, className1, phoneNumber1);
74
75     System.out.println("请输入学生2的信息: ");
76     System.out.print("学号: ");
77     String id2 = scanner.nextLine();
78     System.out.print("姓名: ");
79     String name2 = scanner.nextLine();
80     System.out.print("班级: ");
```

## (拓展题目) 题目 1:

### (一) 实验题目

(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。

### (二) 实现过程

这是一个简单的 CAI (Computer Assisted Instruction) 系统，主要用来出题和判断用户输入的答案是否正确。代码只包含了一个类，所以整个系统只有一个类。

类名：CAI

成员变量：

rnd: SecureRandom 类型，用于生成随机数。

scan: Scanner 类型，用于从控制台读取用户输入。

方法：

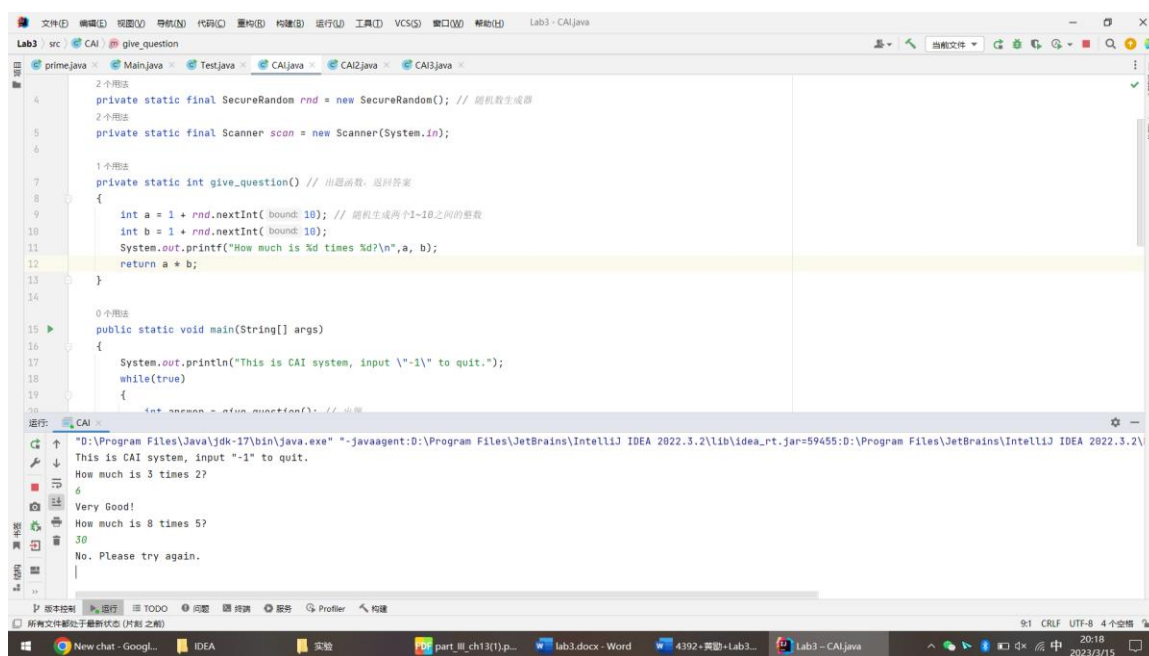
give\_question(): 出题方法，返回一个 int 类型的答案。

main(): 程序入口，控制用户输入和答案判断的流程。

在 main()方法中，程序首先输出欢迎信息，然后进入一个无限循环。循环中，程序首先调用 give\_question()方法生成一个新的问题，然后等待用户输入答案。如果用户输入的答案正确，程序输出"Very Good!"，并继续下一轮循环。如果用户输入的答案不正确，程序会循环提示用户输入，直到输入正确为止。如果用户输入"-1"，程序退出循环，输出结束信息。

整个系统只有一个类，因为它很简单，只有一个任务：出题和判断用户输入的答案是否正确。

### (三) 过程截图



```
Lab3 - CAI.java
4 private static final SecureRandom rnd = new SecureRandom(); // 随机数生成器
5 private static final Scanner scan = new Scanner(System.in);
6
7 private static int give_question() // 出题函数，返回答案
8 {
9     int a = 1 + rnd.nextInt(10); // 随机生成两个1-10之间的整数
10    int b = 1 + rnd.nextInt(10);
11    System.out.printf("How much is %d times %d?\n", a, b);
12    return a * b;
13 }
14
15 public static void main(String[] args)
16 {
17     System.out.println("This is CAI system, input \'-1\' to quit.");
18     while(true)
19     {
20         int answer = give_question(); // 出题
21         int input = scan.nextInt();
22         if (input == -1) break;
23         if (input == answer)
24             System.out.println("Very Good!");
25         else
26             System.out.println("No. Please try again.");
27     }
28 }
```

运行: CAI

```
"D:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA 2022.3.2\lib\idea_rt.jar=59455:D:\Program Files\JetBrains\IntelliJ IDEA 2022.3.2\
This is CAI system, input \'-1\' to quit.
How much is 3 times 2?
6
Very Good!
How much is 8 times 5?
30
No. Please try again.
```

```

运行: CAI x
"D:\Program Files\Java\jdk-17\bin\java.exe" "-ja
This is CAI system, input "-1" to quit.
How much is 3 times 2?
6
Very Good!
How much is 8 times 5?
30
No. Please try again.

```

## 出题方法:

```

3 public class CAI {
    2 个用法
4     private static final SecureRandom rnd = new SecureRandom(); // 随机数生成器
    2 个用法
5     private static final Scanner scan = new Scanner(System.in);
6
    1 个用法
7     private static int give_question() // 出题函数, 返回答案
8     {
9         int a = 1 + rnd.nextInt( bound: 10); // 随机生成两个1~10之间的整数
10        int b = 1 + rnd.nextInt( bound: 10);
11        System.out.printf("How much is %d times %d?\n",a, b);
12        return a * b;
13    }

```

## 用户回答:

```

System.out.println("This is CAI system, input \"-1\" to quit.");
while(true)
{
    int answer = give_question(); // 出题
    int input = scan.nextInt();
    if(input == answer) // 判断答案是否正确
    {
        System.out.println("Very Good!");
    }
    else if(input == -1)
        break;
    else {
        do // 答案不正确, 循环提示输入直到回答正确
        {
            System.out.println("No. Please try again.");
            input = scan.nextInt();
        }
        while(input != answer);
        System.out.println("Very Good!");
    }
}

```

## 题目 2:

### (一) 实验题目

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。

### (二) 实现过程

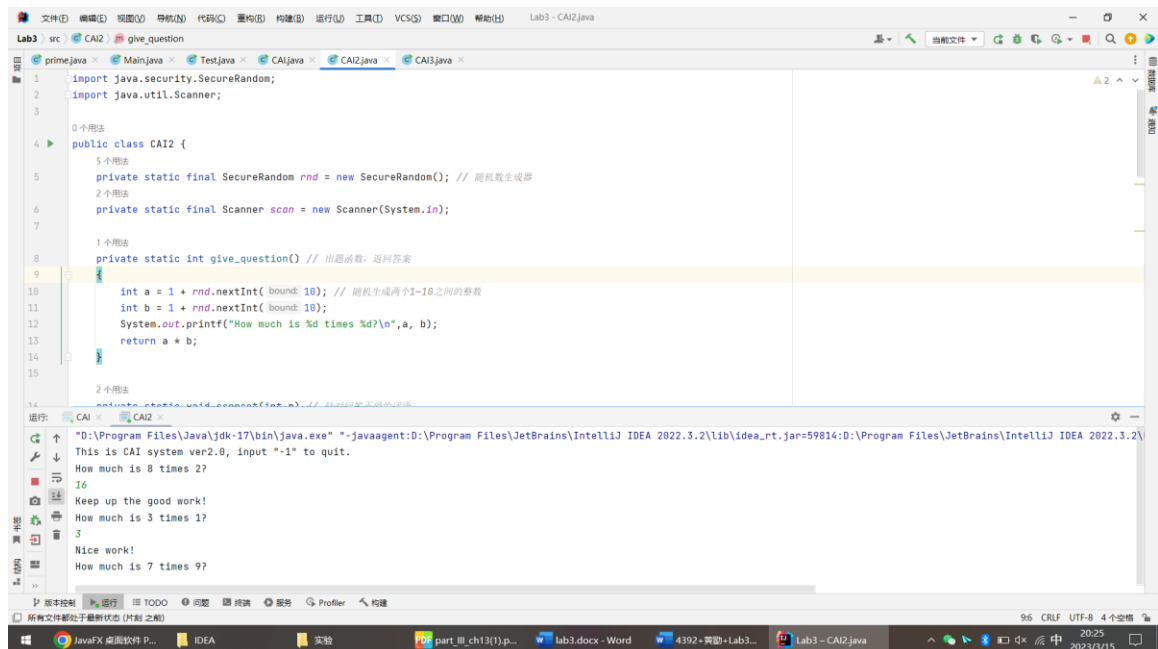
需要设计一个名为"CAI2"的类。

类中的修改主要包括:

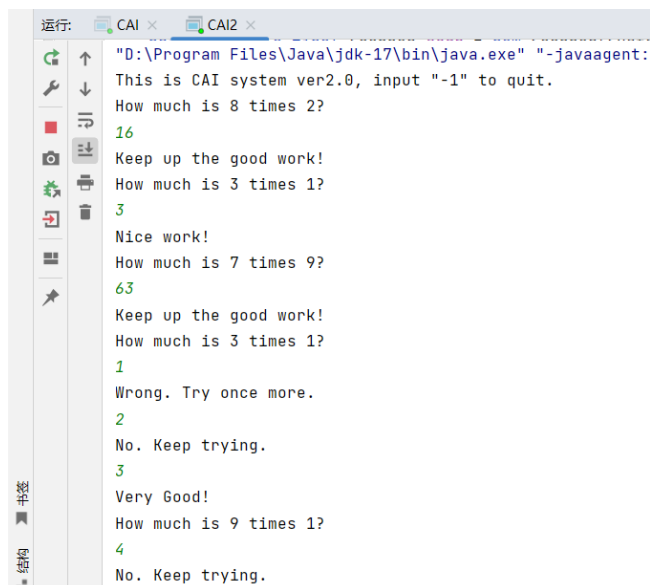
"correct"方法和"wrong"方法,分别用于回答正确和错误的情况下的提示信息输出。这些信息是通过一个 switch 语句随机选取的。

程序的 main 函数中,通过一个无限循环来不断出题并等待用户的输入。如果用户输入的答案与正确答案一致,则随机输出一条“回答正确”的提示信息;如果用户输入的是-1,则退出程序;否则,通过一个 do-while 循环不断提示用户重新输入,直到用户输入的答案与正确答案一致。在每次回答正确后,再次随机输出一条“回答正确”的提示信息。

### (三) 过程截图



回答优化:



随机评语算法（使用 switch）：

```
16      private static void correct(int n) // 针对回答正确的评语
17      {
18          switch (n) {
19              case 1 -> System.out.println("Very Good!");
20              case 2 -> System.out.println("Excellent!");
21              case 3 -> System.out.println("Nice work!");
22              case 4 -> System.out.println("Keep up the good work!");
23          }
24      }
25
26      1个用法
27      private static void wrong(int n) // 针对回答错误的评语
28      {
29          switch (n) {
30              case 1 -> System.out.println("No. Please try again.");
31              case 2 -> System.out.println("Wrong. Try once more.");
32              case 3 -> System.out.println("Don't give up!");
33              case 4 -> System.out.println("No. Keep trying.");
34          }
35      }
```

### 题目 3:

#### (一) 实验题目

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。

#### (二) 实现过程

编写一个 CAI3 类，根据题目要求主要是修改 main 主方法的流程，具体实现如下：

CAI3 类

sum: 表示用户总共做了几道题

right: 表示用户总共做对了几道题



give\_question(): 随机生成两个 1-10 之间的整数，并输出问题，返回问题的答案

correct(int n): 根据 n 的值，输出回答正确的评语

wrong(int n): 根据 n 的值，输出回答错误的评语

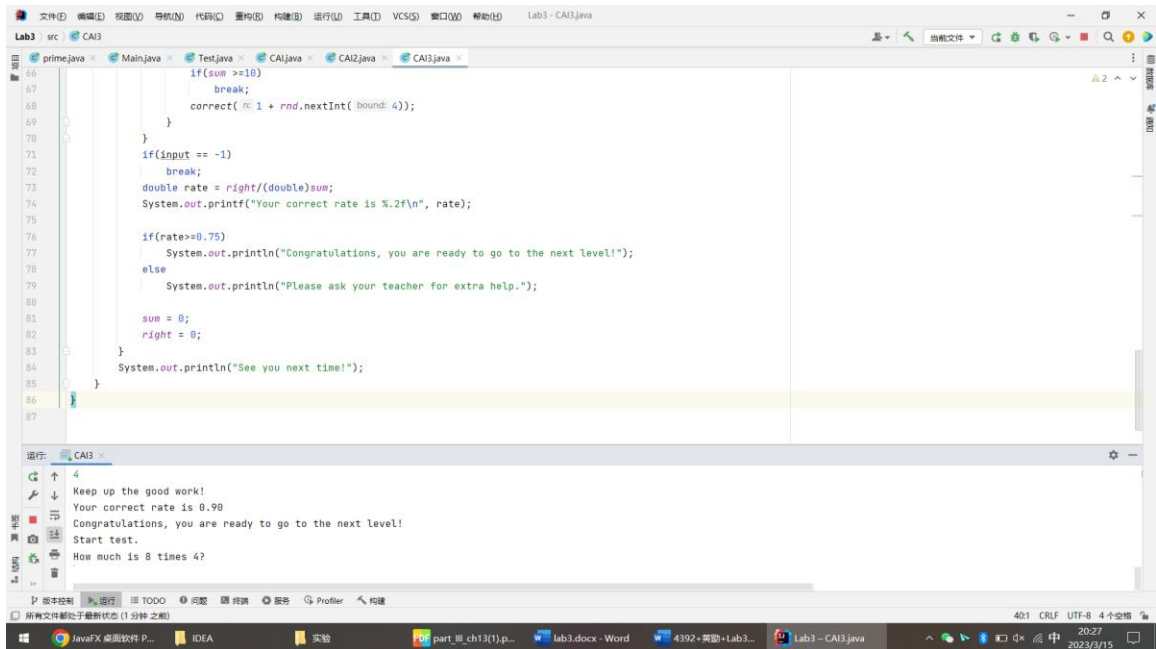
main(String[] args): 主方法，控制程序的运行逻辑，实现算术测试的流程

main 方法:

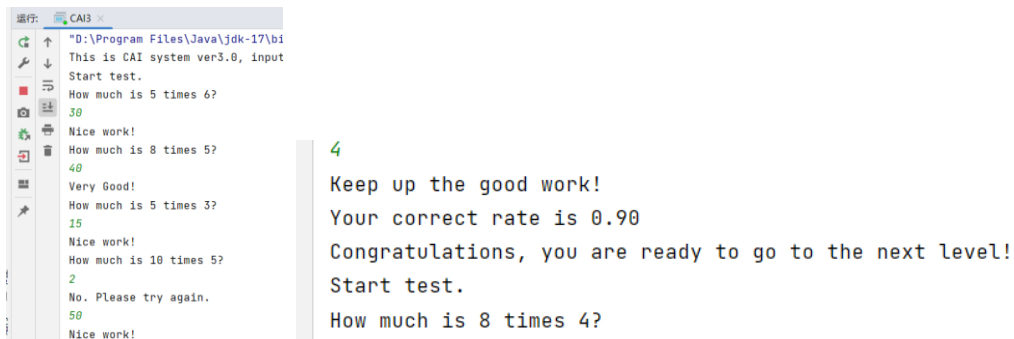
- 输出欢迎语和提示用户如何退出程序
- while 循环，一直进行算术测试，直到用户输入-1
- 输出开始测试的提示
- while 循环，一直出题，直到用户做对 10 道题或者输入-1
- 生成题目，保存答案
- 等待用户输入
- 如果用户输入答案正确，则给出正确评语，right+1，sum+1
- 如果用户输入-1，则跳出循环，回到外层 while 循环
- 如果用户输入答案错误，进入 do-while 循环，给出错误评语，sum+1，等待用户重新输入
- 如果用户最后答对了，给出正确评语，right+1
- 如果用户答对了 10 道题，跳出循环
- 根据用户做对题目的比例给出最终评价
- 如果做对比例大于等于 0.75，给出恭喜语；否则给出请教老师的建议
- 重置 sum 和 right 的值，等待下一次测试
- 输出结束语

以上就是整个程序的类设计思路。

### (三) 过程截图



### 使用过程:



### 主要修改评分过程:

```

double rate = right/(double)sum;
System.out.printf("Your correct rate is %.2f\n", rate);

if(rate>=0.75)
    System.out.println("Congratulations, you are ready to go to the next level!");
else
    System.out.println("Please ask your teacher for extra help.");

sum = 0;
right = 0;

```

### 三、实验总结与心得记录

首先，在本次实验中，我们使用了循环和条件语句等控制结构来实现不同的功能，如出题、评价回答、统计正确率等。通过这些控制结构的使用，我们可以更加灵活地控制程序的流程，实现所需的功能。同时，本次实验也让我们更加熟悉了控制结构的使用方法。

其次，本次实验中还涉及了面向对象封装特性。通过定义类、成员变量和成员方法等方式，我们将程序中的不同功能模块封装在不同的类中，以实现更好的模块化和复用性。这种面向对象的编程方式在实际开发中具有很大的优势，可以更好地提高代码的可维护性和可扩展性。

最后，本次实验还涉及了 JavaFX 图形界面的使用。通过使用 JavaFX 提供的各种控件，我们可以快速构建出漂亮的图形用户界面，提升程序的交互体验。同时，JavaFX 也提供了各种事件处理器和动画效果等功能，使得界面交互更加生动有趣。

综上所述，本次实验让我们掌握了控制结构的使用、面向对象封装特性以及 JavaFX 图形界面的开发技术，这些都是在实际开发中非常实用的技能。通过本次实验的实践和总结，我们更加深入地理解了这些知识点，并能够更加熟练地应用到实际项目中。