

廈門大學



信息学院软件工程系

《JAVA 程序设计》实验报告

实验三

姓名：任宇

学号：33920212204567

学院：信息学院

专业：软件工程

完成时间：2024-03-17

一、 实验目的及要求

- 练习控制结构
- 熟悉面向对象封装特性

二、 实验题目及实现过程

1. 题目一：

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。

- a) 编写一个方法，它判断一个数是否为素数。
- b) 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？
- c) 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。

（一）实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK 版本：JDK17

（二）实现过程

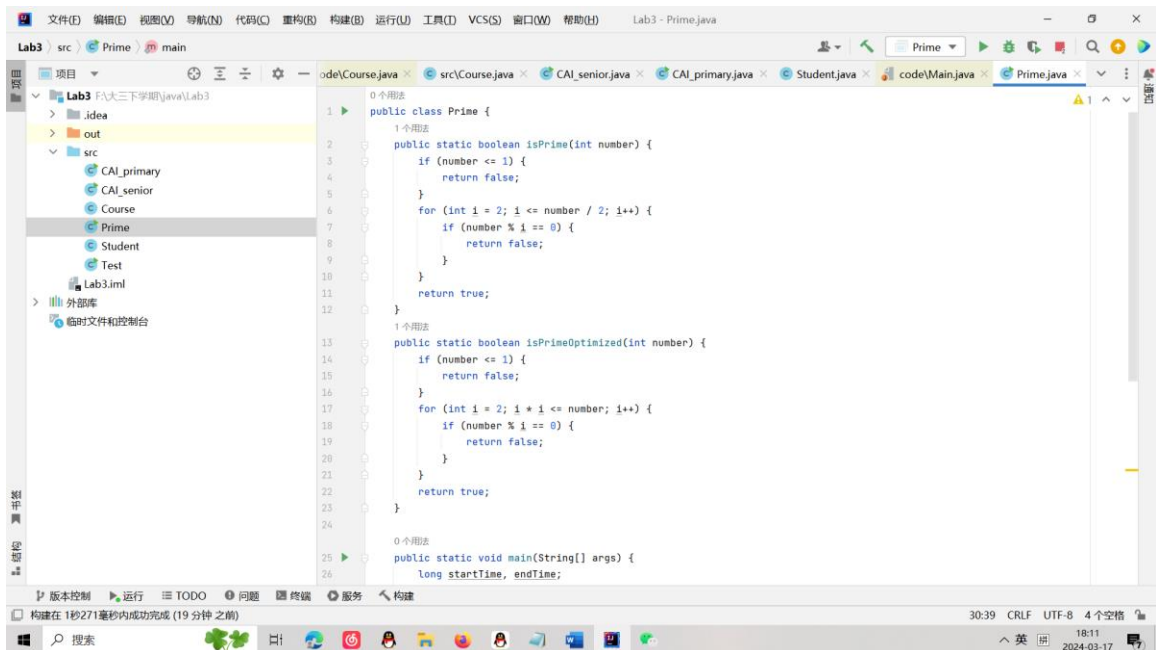
本题只设计了 Prime 类，包含了素数判断的逻辑和程序的入口点。成员方法包括 `isPrime(int number)`、`isPrimeOptimized(int number)` 和 `main(String[] args)`，具体作用如下：

- `isPrime(int number)`：判断一个数是否为素数的基本方法，它遍历从 2 到给定数的一半的所有整数。

- `isPrimeOptimized(int number)`: 优化后判断素数的方法。它只遍历从 2 到给定数的平方根的所有整数，这样减少了检查的范围，提高了效率。
- `main(String[] args)`: 程序的入口点。它首先使用 `isPrime` 方法遍历并打印出小于 10000 的所有素数，并记录执行时间。然后，它使用 `isPrimeOptimized` 方法重复相同的任务，但不打印素数，只是为了比较执行时间。

(三) 过程截图

全屏截图:



Prime 类代码实现:

```
1 public class Prime {
2     1 个用法
3     public static boolean isPrime(int number) {
4         if (number <= 1) {
5             return false;
6         }
7         for (int i = 2; i <= number / 2; i++) {
8             if (number % i == 0) {
9                 return false;
10            }
11        }
12        return true;
13    }
```

```
13 public static boolean isPrimeOptimized(int number) {
14     if (number <= 1) {
15         return false;
16     }
17     for (int i = 2; i * i <= number; i++) {
18         if (number % i == 0) {
19             return false;
20         }
21     }
22     return true;
23 }
24
25 0 个用法
26 public static void main(String[] args) {
27     long startTime, endTime;
28     startTime = System.currentTimeMillis();
29     for (int i = 2; i < 10000; i++) {
30         if (isPrime(i)) {
31             System.out.println(i);
32         }
33     }
34     endTime = System.currentTimeMillis();
35     System.out.println("未优化方法所用时间为: " + (endTime - startTime) + " ms.");
36     startTime = System.currentTimeMillis();
37     for (int i = 2; i < 10000; i++) {
38         if (isPrimeOptimized(i)) {
39             System.out.println(i);
40         }
41     }
42     endTime = System.currentTimeMillis();
43     System.out.println("优化后的方法所用时间为: " + (endTime - startTime) + " ms.");
44 }
```

Prime 类运行过程:



```
运行: CAL_primary x CAL_senior x Prime x
F:\Java\bin\java.exe "-javaagent:F:\IntelliJ IDEA Community Edition 2022.3.2\li
2
3
5
7
11
13
9951
9941
9949
9967
9973
未优化方法所用时间为: 12 ms。
优化后的方法所用时间为: 1 ms。
```

2. 题目二:

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。

（一）实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK 版本：JDK17

（二）实现过程

实验中实现了一个简单的选课系统，包含三个主要的类：Student，Course，和 Test，以及选课过程中随机分配成绩的逻辑。

● Student 类

代表学生，包含基本信息如学号（studentId）、姓名（name）、班级（className）和电话号码（phone）。

构造函数：提供两个重载构造函数，一个不包括电话号码（默认为空字符串），另一个包含电话号码，以适应不同的创建需求。

Get 和 Set 方法：允许外部代码获取和修改学生的属性。

● Course 类

代表课程，包含课程编号（courseId）和课程名称（courseName）。

构造函数：初始化课程编号和课程名称。

Get 和 Set 方法：允许外部代码获取和修改课程的属性。

● Test 类

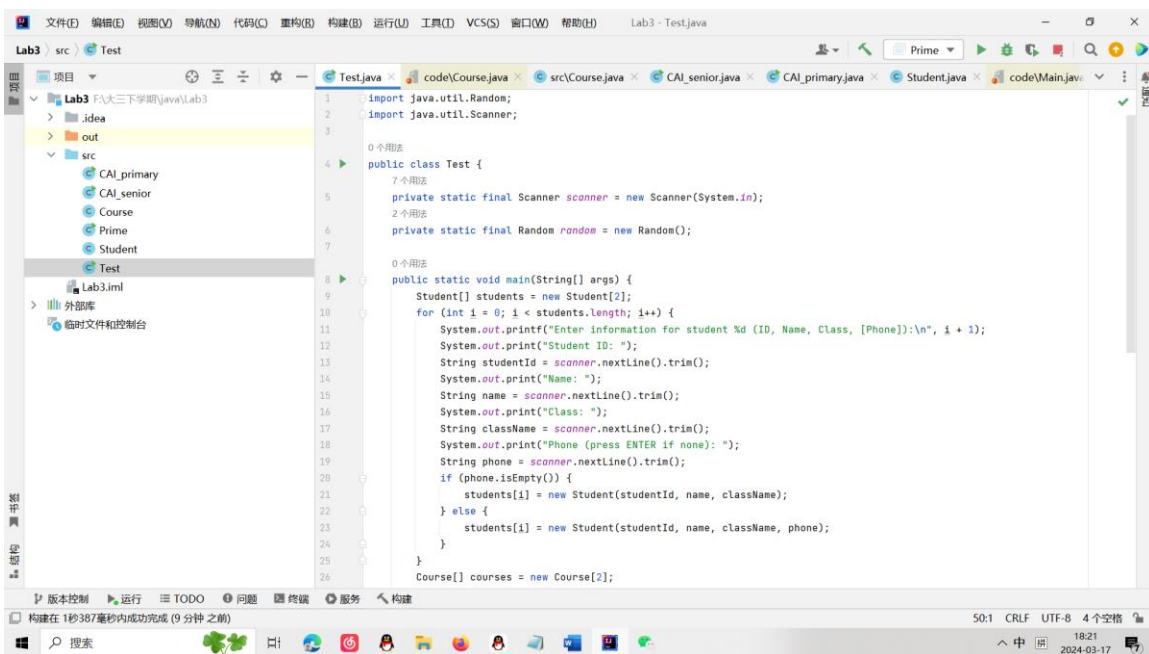
包含 main 方法，是程序的入口点，用于演示如何使用 Student 和 Course 类，并实现选课逻辑。

● 具体实现

程序开始运行时，Test 类的 main 方法首先提示用户输入两个学生的信息，包括学号、姓名、班级和可选的电话号码。对于每个学生，根据用户是否输入了电话号码来决定使用哪个构造函数创建 Student 对象。接着提示用户输入两门课程的信息，包括课程编号和课程名称，使用输入的信息创建 Course 对象。然后，main 方法遍历学生数组，对于每个学生调用 selectCourse 方法。该方法随机选择一门课程并随机生成一个成绩，然后输出学生选择的课程和成绩。

(三) 过程截图

全屏截图：



Student 类代码实现：

```
1 public class Student {  
2     4 个用法  
3     private String studentId;  
4     4 个用法  
5     private String name;  
6     4 个用法  
7     private String className;  
8     4 个用法  
9     private String phone;  
10 }
```

```

7      // 电话号码为空
      1个用法
8      public Student(String studentId, String name, String className) {
9          this.studentId = studentId;
10         this.name = name;
11         this.className = className;
12         this.phone = "";
13     }
14     // 包含电话号码
      1个用法
15     public Student(String studentId, String name, String className, String phone) {
16         this.studentId = studentId;
17         this.name = name;
18         this.className = className;
19         this.phone = phone;
20     }
      0个用法
21     public String getStudentId() { return studentId; }
      1个用法
24     public String getName() { return name; }
      0个用法
27     public String getClassName() { return className; }
      0个用法
30     public String getPhone() { return phone; }
      0个用法
33     public void setStudentId(String studentId) { this.studentId = studentId; }
      0个用法
36     public void setName(String name) { this.name = name; }
      0个用法
39     public void setClassName(String className) { this.className = className; }
      0个用法
42     public void setPhone(String phone) { this.phone = phone; }

```

Course 类代码实现:

```

1      public class Course {
      3个用法
2          private String courseId;
      3个用法
3          private String courseName;
      1个用法
4          public Course(String courseId, String courseName) {
5              this.courseId = courseId;
6              this.courseName = courseName;
7          }
      0个用法
8          public String getCourseId() { return courseId; }
      1个用法
11         public String getCourseName() { return courseName; }
      0个用法
14         public void setCourseId(String courseId) { this.courseId = courseId; }
      0个用法
17         public void setCourseName(String courseName) { this.courseName = courseName; }

```

Test 类代码实现:

```

1      import java.util.Random;
2      import java.util.Scanner;
3
      0个用法
4      public class Test {
      7个用法
5          private static final Scanner scanner = new Scanner(System.in);
      2个用法
6          private static final Random random = new Random();

```

```

8 public static void main(String[] args) {
9     Student[] students = new Student[2];
10     for (int i = 0; i < students.length; i++) {
11         System.out.printf("Enter information for student %d (ID, Name, Class, [Phone]):\n", i + 1);
12         System.out.print("Student ID: ");
13         String studentId = scanner.nextLine().trim();
14         System.out.print("Name: ");
15         String name = scanner.nextLine().trim();
16         System.out.print("Class: ");
17         String className = scanner.nextLine().trim();
18         System.out.print("Phone (press ENTER if none): ");
19         String phone = scanner.nextLine().trim();
20         if (phone.isEmpty()) {
21             students[i] = new Student(studentId, name, className);
22         } else {
23             students[i] = new Student(studentId, name, className, phone);
24         }
25     }
26     Course[] courses = new Course[2];
27     for (int i = 0; i < courses.length; i++) {
28         System.out.printf("Enter information for course %d (CourseID CourseName):\n", i + 1);
29         System.out.print("courseId: ");
30         String courseId = scanner.nextLine().trim();
31         System.out.print("courseName: ");
32         String courseName = scanner.nextLine().trim();
33         courses[i] = new Course(courseId, courseName);
34     }
35
36     // 为每个学生随机选择一门课程并分配成绩
37     for (Student student : students) {
38         selectCourse(student, courses);
39     }
40     scanner.close();
41 }
42
43 1 个用法
44 private static void selectCourse(Student student, Course[] courses) {
45     int courseIndex = random.nextInt(courses.length);
46     Course selectedCourse = courses[courseIndex];
47     int grade = random.nextInt( bound: 101);
48     System.out.printf("%s selected %s with score: %d\n", student.getName(), selectedCourse.getCourseName(), grade);
49 }

```

Test 类运行过程:

```

运行: CAI_primary x CAI_senior x Test x
F:\Java\bin\java.exe "-javaagent:F:\IntelliJ IDEA Community Edition 2022.3.2\lib\
Enter information for student 1 (ID, Name, Class, [Phone]):
Student ID: 339
Name: renyu
Class: 1
Phone (press ENTER if none):
Enter information for student 2 (ID, Name, Class, [Phone]):
Student ID: 119
Name: 张三
Class: 1
Phone (press ENTER if none): 13515002323
Enter information for course 1 (CourseID CourseName):
courseId: 1
courseName: physics
Enter information for course 2 (CourseID CourseName):
courseId: 2
courseName: english
renyu selected physics with score: 75
张三 selected physics with score: 4

```


3. 题目三：

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。

（一）实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK 版本：JDK17

（二）实现过程

实验中设计了 `CAI_primary` 类用于实现题目要求。`CAI_primary` 类包含了程序的入口点（`main` 方法）和核心功能（`generateNewQuestion` 方法）。

- 全局变量

`randomNumbers`：一个 `SecureRandom` 实例，用于生成随机数，确保每次生成的乘法问题都是随机的。

- 核心方法

`main(String[] args)` 方法：程序的入口点。它包含一个无限循环，不断调用 `generateNewQuestion` 方法来生成新的乘法问题。

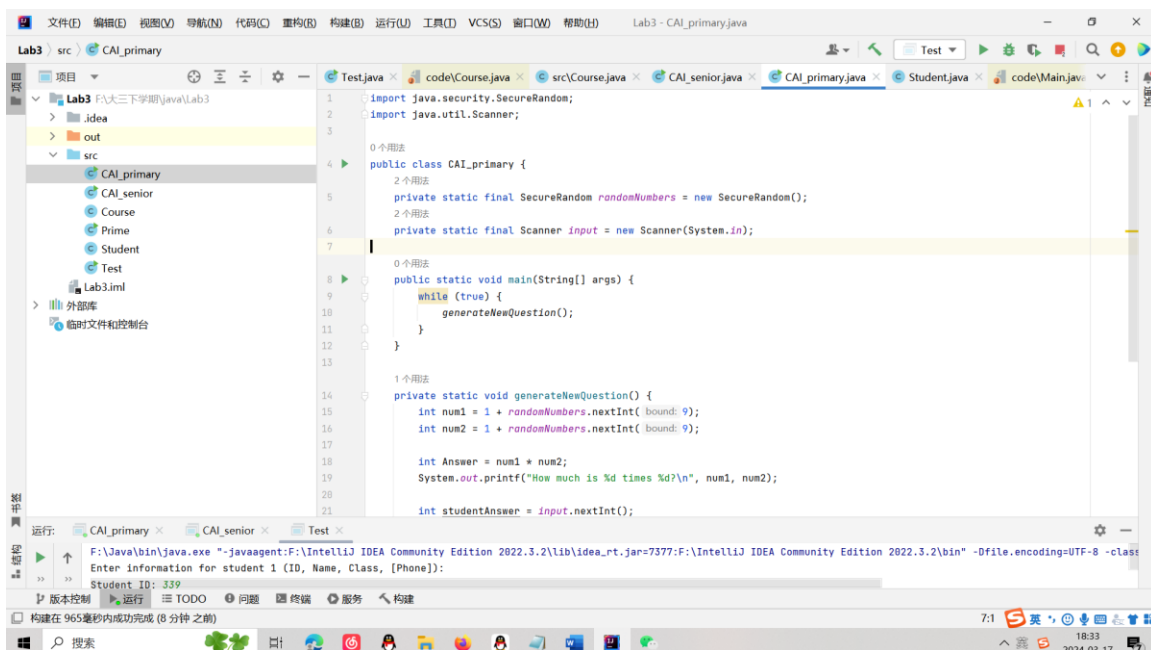
`generateNewQuestion()` 方法：生成新的乘法问题，并处理用户的答案。

- 具体实现

首先，方法使用 `randomNumbers` 对象生成两个 1 到 9 之间的随机整数（`num1` 和 `num2`），然后计算这两个数的乘积（`Answer`）。接下来，方法向用户展示这个乘法问题，并使用 `input` 对象接收用户输入的答案（`studentAnswer`）。如果用户输入的答案不正确，程序会打印出提示信息（“No. Please try again.”），并要求用户重新输入答案，直到正确为止。一旦用户回答正确，程序会打印出鼓励信息（“Very good!”），然后 `main` 方法会触发生成下一个乘法问题，循环继续。

（三）过程截图

全屏截图：



CAI_primary 类代码实现：

```
1 import java.security.SecureRandom;
2 import java.util.Scanner;
3
4 public class CAI_primary {
5     private static final SecureRandom randomNumbers = new SecureRandom();
6     private static final Scanner input = new Scanner(System.in);
7
8     public static void main(String[] args) {
9         while (true) {
10             generateNewQuestion();
11         }
12     }
13 }
```

```
14 private static void generateNewQuestion() {
15     int num1 = 1 + randomNumbers.nextInt( bound: 9);
16     int num2 = 1 + randomNumbers.nextInt( bound: 9);
17
18     int Answer = num1 * num2;
19     System.out.printf("How much is %d times %d?\n", num1, num2);
20
21     int studentAnswer = input.nextInt();
22
23     while (studentAnswer != Answer) {
24         System.out.println("No. Please try again.");
25         studentAnswer = input.nextInt();
26     }
27
28     System.out.println("Very good!");
29 }
30
31 }
```

CAI_primary 类运行过程:

4. 扩展题目:

- (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 $1 \sim 4$ 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。

- (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主题，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。

（一）实验环境

集成开发环境：IDEA Community Edition 2022.3.2

JDK 版本：JDK17

（二）实现过程

实验中设计了 CAI_senior 类在 CAI_primary 的基础上进行改变并满足降低疲劳感和监督学生表现的要求。

- 全局变量

使用静态变量来记录正确答案的总数（correctAnswers）和尝试的总次数（totalAttempts）。

- 核心方法

generateNewQuestion() 方法：生成新的乘法问题并更新正确答案的总数和尝试的总次数。

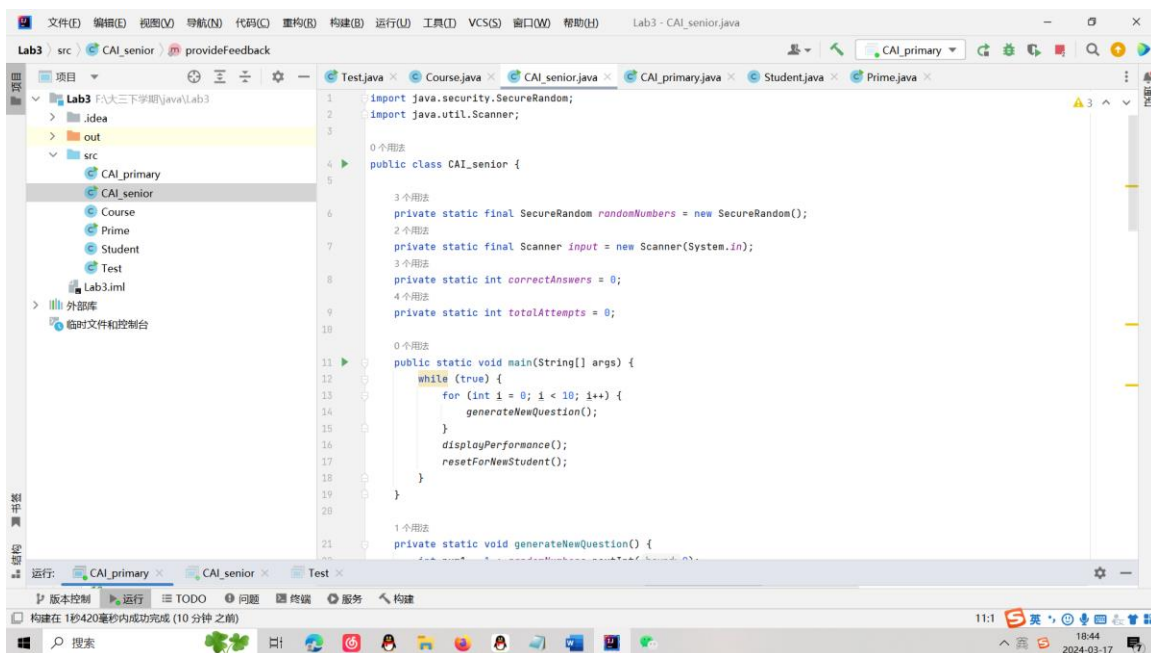
provideFeedback (boolean isCorrect) 方法：根据学生的回答提供随机的即时反馈。

displayPerformance() 方法：计算并显示正确率，同时根据正确率提供不同的反馈。

resetForNewStudent() 方法：重置正确答案和尝试次数的计数器，为下一个学生答题做好准备。

(三) 过程截图

全屏截图：



CAI_senior 类代码实现：

```
1 import java.security.SecureRandom;
2 import java.util.Scanner;
3
4 0 个用法
5 public class CAI_senior {
6
7     3 个用法
8     private static final SecureRandom randomNumbers = new SecureRandom();
```

```

7      private static final Scanner input = new Scanner(System.in);
      3 个用法
8      private static int correctAnswers = 0;
      4 个用法
9      private static int totalAttempts = 0;
10
      0 个用法
11  public static void main(String[] args) {
12      while (true) {
13          for (int i = 0; i < 10; i++) {
14              generateNewQuestion();
15          }
16          displayPerformance();
17          resetForNewStudent();
18      }
19  }

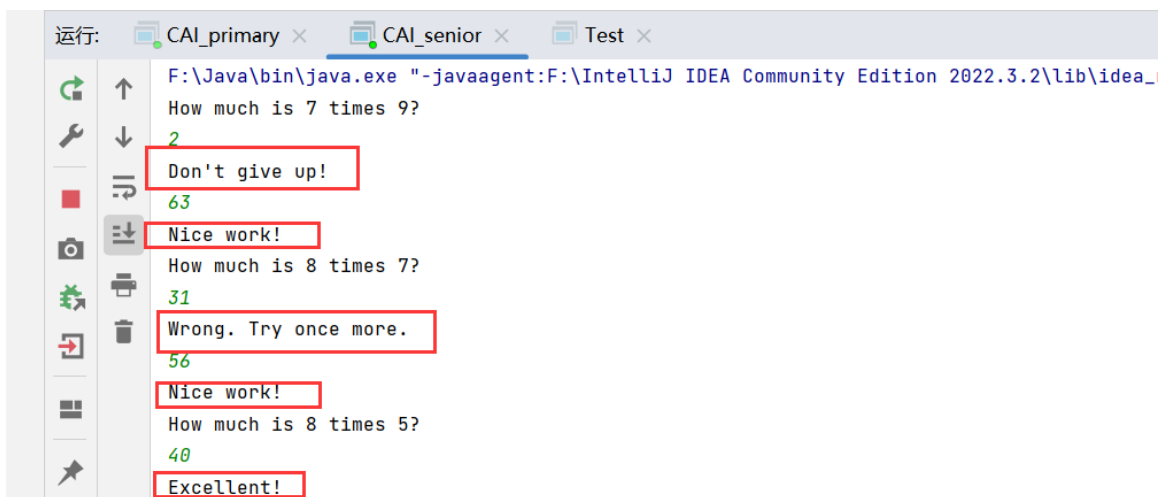
21  private static void generateNewQuestion() {
22      int num1 = 1 + randomNumbers.nextInt( bound: 9);
23      int num2 = 1 + randomNumbers.nextInt( bound: 9);
24
25      int correctAnswer = num1 * num2;
26      System.out.printf("How much is %d times %d?\n", num1, num2);
27
28      int studentAnswer = input.nextInt();
29      totalAttempts++;
30
31      while (studentAnswer != correctAnswer) {
32          provideFeedback( isCorrect: false);
33          studentAnswer = input.nextInt();
34          totalAttempts++;
35      }
36
37      provideFeedback( isCorrect: true);
38      correctAnswers++;
39  }

76  private static void displayPerformance() {
77      double correctRate = ((double) correctAnswers / totalAttempts) * 100;
78      System.out.printf("Correct rate: %.2f%\n", correctRate);
79      if (correctRate < 75) {
80          System.out.println("Please ask your teacher for extra help.");
81      } else {
82          System.out.println("Congratulations, you are ready to go to the next level!");
83      }
84  }
85
      1 个用法
86  private static void resetForNewStudent() {
87      System.out.println("\nNew student can start answering questions.\n");
88      // 重置变量
89      correctAnswers = 0;
90      totalAttempts = 0;
91  }
92  }

```

```
41 private static void provideFeedback(boolean isCorrect) {
42     int responseIndex = 1 + randomNumbers.nextInt( bound: 4);
43     if (isCorrect) {
44         switch (responseIndex) {
45             case 1:
46                 System.out.println("Very good!");
47                 break;
48             case 2:
49                 System.out.println("Excellent!");
50                 break;
51             case 3:
52                 System.out.println("Nice work!");
53                 break;
54             case 4:
55                 System.out.println("Keep up the good work!");
56                 break;
57         }
58     } else {
59         switch (responseIndex) {
60             case 1:
61                 System.out.println("No. Please try again.");
62                 break;
63             case 2:
64                 System.out.println("Wrong. Try once more.");
65                 break;
66             case 3:
67                 System.out.println("Don't give up!");
68                 break;
69             case 4:
70                 System.out.println("No. Keep trying.");
71                 break;
72         }
73     }
74 }
```

CAI_senior 类代码运行情况:



```
运行: CAI_primary x CAI_senior x Test x
F:\Java\bin\java.exe "-javaagent:F:\IntelliJ IDEA Community Edition 2022.3.2\lib\idea_
How much is 7 times 9?
2
Don't give up!
63
Nice work!
How much is 8 times 7?
31
Wrong. Try once more.
56
Nice work!
How much is 8 times 5?
40
Excellent!
```

36

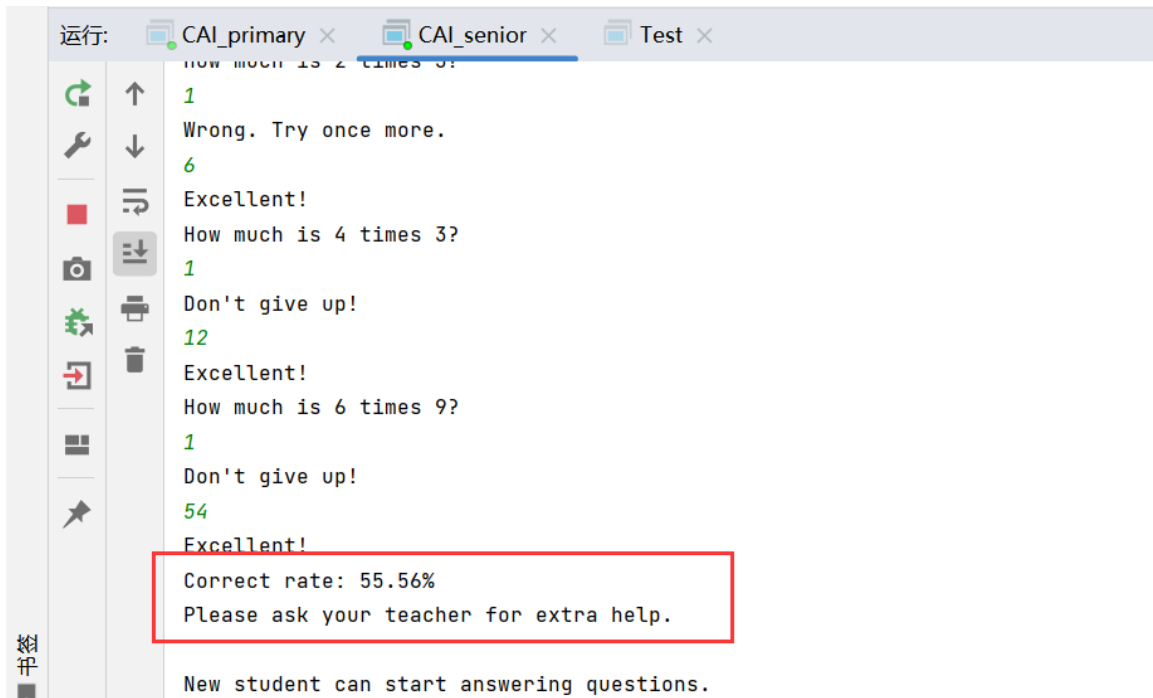
Very good!

Correct rate: 76.92%

Congratulations, you are ready to go to the next level!

New student can start answering questions.

How much is 6 times 6?



三、实验总结与心得记录

通过本次实验，我深刻体会到了控制结构的重要性以及面向对象编程中封装特性的实际应用。通过实现素数判断程序、简单的选课系统，以及计算机辅助教学(CAI)程序，我不仅练习了循环和条件判断等基础控制结构，还学会了如何通过方法封装复杂的逻辑，提高代码的重用性和可维护性。