厦門大學


# 信息学院软件工程系

## 《JAVA 程序设计》实验报告

实验三

姓名：庾晓萍

学号：20420192201952

学院：信息学院

专业：软件工程

完成时间：2022/3/12

# 一、实验目的及要求

（一）实验目的

1、练习控制结构

2、熟悉简单的枚举

3、熟悉简单的 JavaFX 图形界面

（二）实验要求

1、下周前二前将工程文档和实验报告打包上传到 FTP

# 二、实验题目及实现过程

一、基本题目：

题目 1：

(Prime Numbers) A positive integer is prime if it's divisible by only 1 and itself. For example, 2, 3, 5 and 7 are prime, but 4, 6, 8 and 9 are not. The number 1, by definition, is not prime.

(1) Write a method that determines whether a number is prime.

(2) Use this method in an application that determines and displays all the prime numbers less than 10,000. How many numbers up to 10,000 do you have to test to ensure that you've found all the primes?

(3) Initially, you might think that n/2 is the upper limit for which you must test to see whether a number n is prime, but you need only go as high as the square root of n. Rewrite the program, and run it both ways.

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

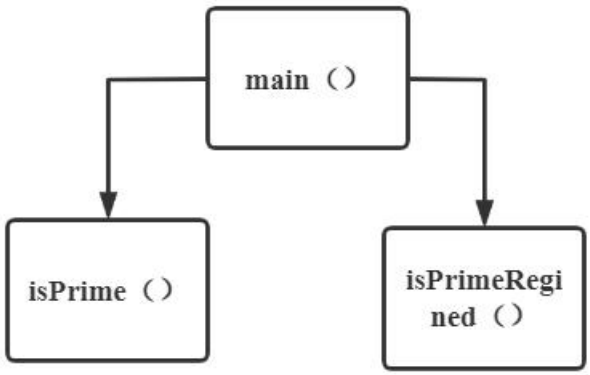
## （二） 实现过程

（1）设计类

设计了名为 PrimeNumbers 的类，其中包括 main 方法，用来调用判断素数的方法，并进行输出。包括 isPrime 方法和 isPrimeRegined 方法。前一种方法设置 n/2 是判断一个数是否是素数的测试上限，后一种方法设置 sqrt(n)为判断一个数是否是素数的测试上限。

（2）调用关系

调用关系如下图，即 main 方法可以对 isPrime 方法和 isPrimeRegined 方法进行调用。




（3）说明

事实上，正如题目所说，设置 n/2 是判断一个数是否是素数的测试上限，与设置 sqrt(n)为判断一个数是否是素数的测试上限，两者输出结果是相同的。

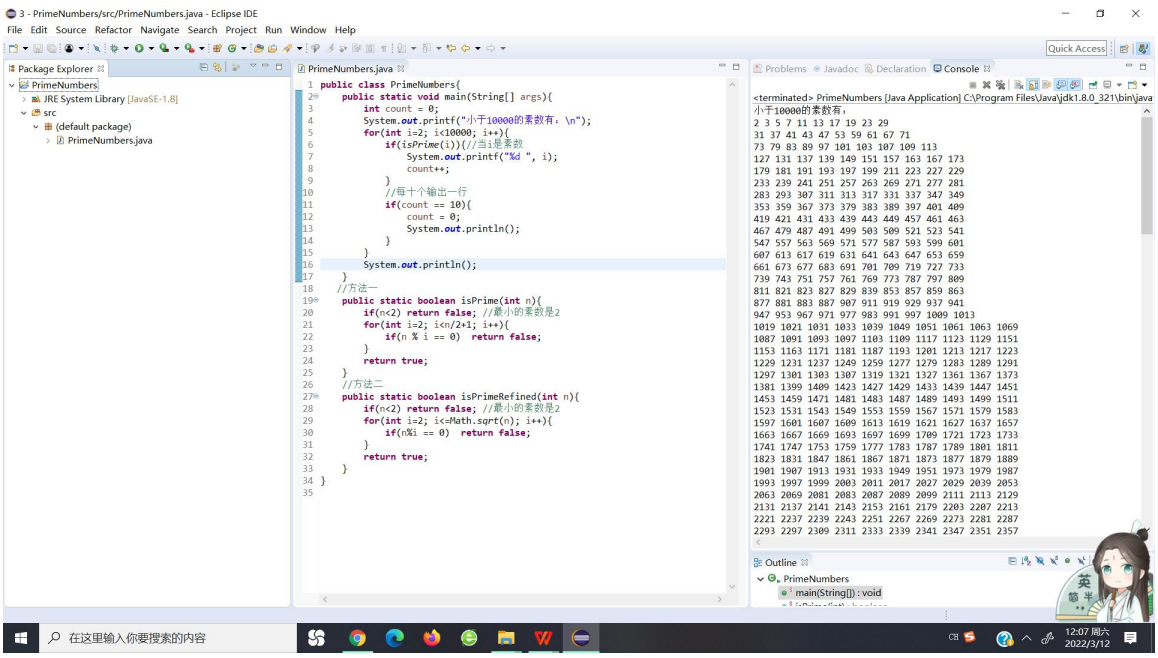
也就是如果我们要判断一个数是否是素数，只需要在 2-sqrt(n)的范围内判断这些数会不会整除 n 便可以实现判断该数是否是素数了。

（4）实验代码

```
public class PrimeNumbers{
    public static void main(String[] args){
        int count = 0;
        System.out.printf("小于10000的素数有：\n");
        for(int i=2; i<10000; i++){
            if(isPrime(i)){//当i是素数
                System.out.printf("%d ", i);
                count++;
            }
            //每十个输出一行
            if(count == 10){
                count = 0;
                System.out.println();
            }
        }
        System.out.println();
    }
   //方法一
    public static boolean isPrime(int n){
        if(n<2) return false; //最小的素数是2
        for(int i=2; i<n/2+1; i++){
            if(n % i == 0)  return false;
        }
        return true;
    }
    //方法二
    public static boolean isPrimeRefined(int n){
        if(n<2) return false; //最小的素数是2
        for(int i=2; i<=Math.sqrt(n); i++){
            if(n%i == 0)  return false;
        }
        return true;
    }
}
```

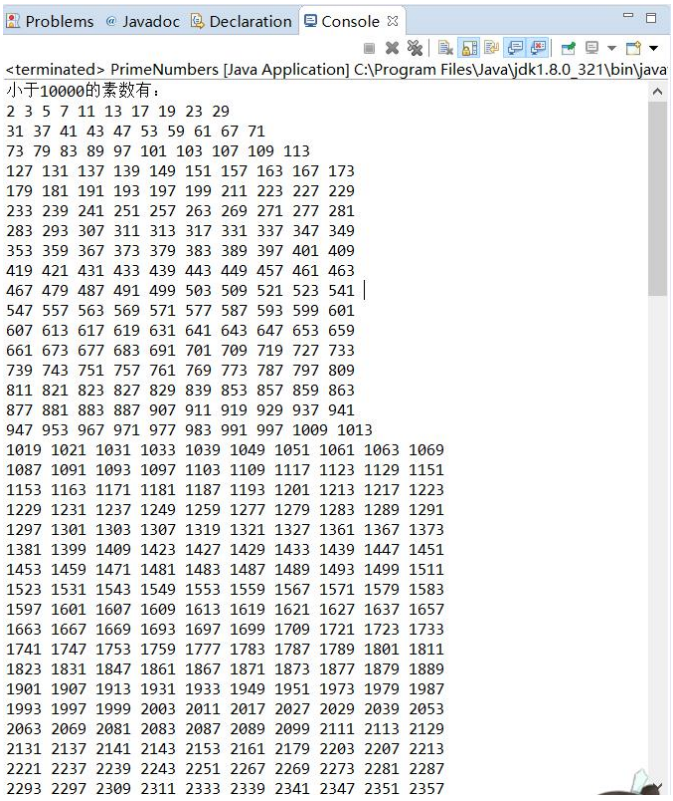
## （三） 过程截图

（1）全屏截图

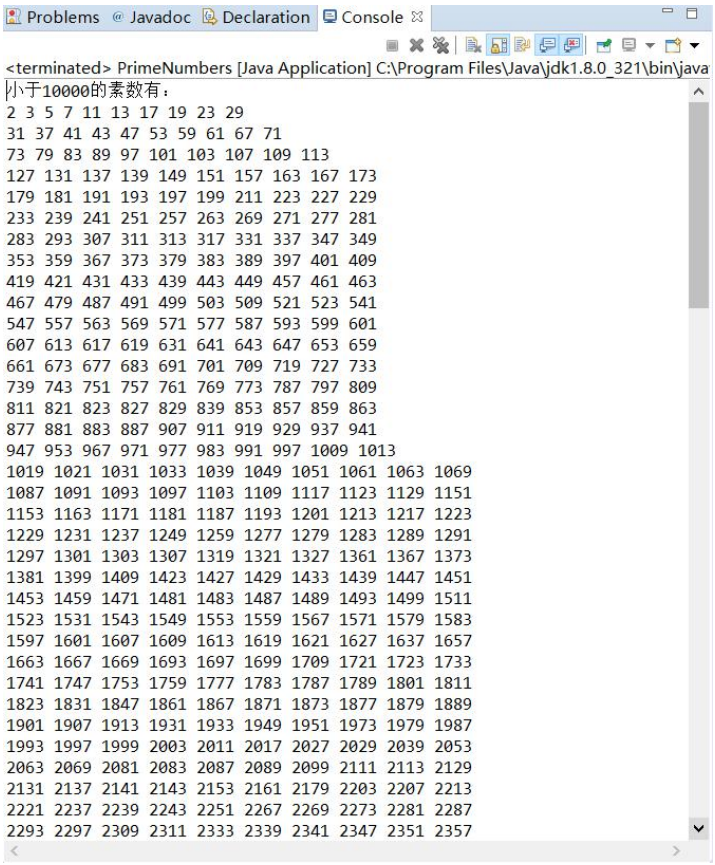



（2）运行结果

1. 调用 isPrime 方法输出小于 10000 的素数




2. 调用 isPrimeRegined 方法输出小于 10000 的素数




题目 2：

(Coin Tossing) Write an application that simulates coin tossing. Let the program toss a coin each time the user chooses the “Toss Coin” menu option. Count the number of times each side of the coin appears. Display the results. The program should call a separate method flip that takes no arguments and returns a value from a Coin enum (HEADS and TAILS). [Note: If the program realistically simulates coin tossing, each side of the coin should appear approximately half the time.]

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

## （二） 实现过程

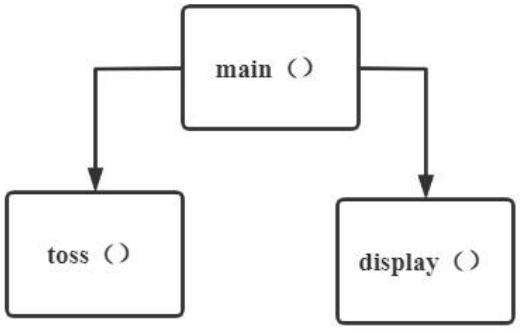
（1）设计类

设计了名为用于投掷硬币的 CoinTossing 类，包括 main 方法，用来显示选择菜单，读取用户输入，并调用相关方法。其中 headsCount 用于记录扔硬币时正面朝上的次数，tailsCount 用于记录扔硬币时反面朝上的次数。调用 toss 方法来进行单次的硬币扔掷，调用 display 方法来显示最终结果。

为了实现用户单次的硬币扔掷，定义一个全局的 SecureRandom 随机数生成器，在 toss 中调用 random.nextInt 方法，于 0 和 1 之间随机生成一个整数。用来模拟扔硬币时正面或反面朝上。

（2）调用关系

调用关系如下图，即 main 方法可以对 toss 方法和 display 方法进行调用。




（3）说明

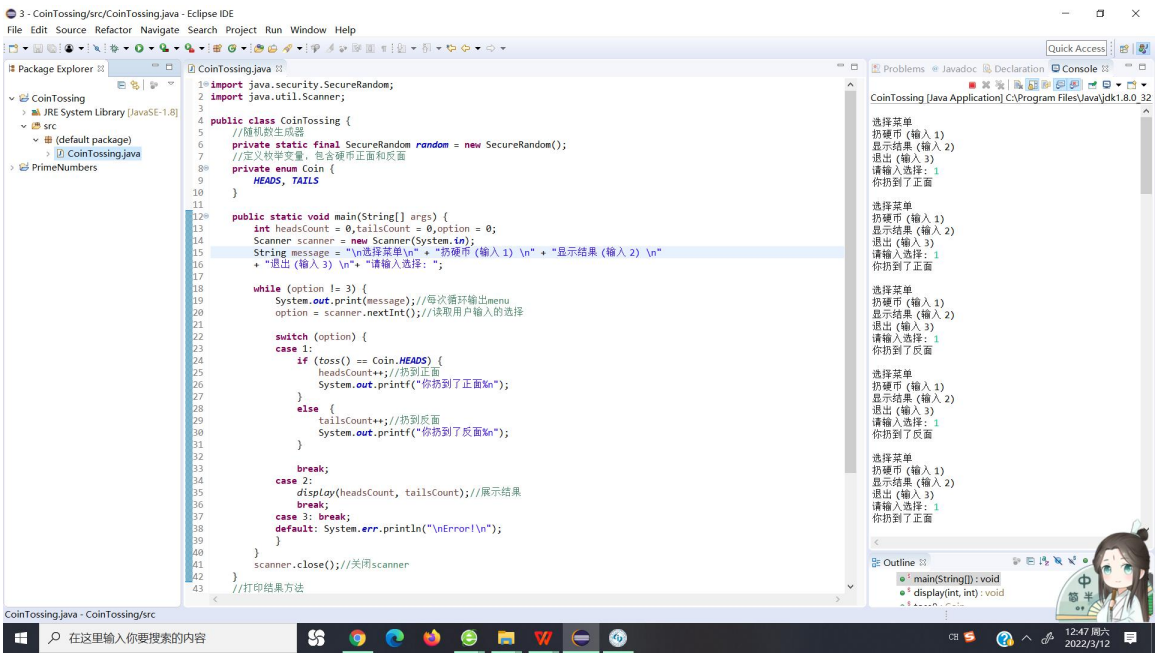
正如实验结果展示的，程序真实地模拟了抛硬币，硬币的每一面出现大约一半的时间。

（4）实验代码

```java
import java.security.SecureRandom;
import java.util.Scanner;

public class CoinTossing {
    //随机数生成器
    private static final SecureRandom random = new SecureRandom();
    //定义枚举变量，包含硬币正面和反面
    private enum Coin {
        HEADS, TAILS
    }

    public static void main(String[] args) {
        int headsCount = 0,tailsCount = 0,option = 0;
        Scanner scanner = new Scanner(System.in);
        String message = "\n选择菜单\n" + "扔硬币（输入 1）\n" + "显示结果（输入 2）\n"
        + "退出（输入 3）\n"+ "请输入选择：";

        while (option != 3) {
            System.out.print(message);//每次循环输出menu
            option = scanner.nextInt();//读取用户输入的选择

            switch (option) {
            case 1:
                if (toss() == Coin.HEADS) {
                    headsCount++;//扔到正面
                    System.out.printf("你扔到了正面%n");
                }
                else  {
                    tailsCount++;//扔到反面
                    System.out.printf("你扔到了反面%n");
                }

                break;
            case 2:
                display(headsCount, tailsCount);//展示结果
                break;
            case 3: break;
            default: System.err.println("\nError!\n");
            }
        }
        scanner.close();//关闭scanner
    }

    //打印结果方法
    public static void display(int headsCount, int tailsCount) {
        System.out.println("Results:\n");
        System.out.printf("HEADS: %d\n", headsCount);
        System.out.printf("TAILS: %d\n", tailsCount);
    }
    //单独的翻转硬币方法
    public static Coin toss() {
        //随机生成一个整数，值域 [0, 2)
        return random.nextInt(2) == 0 ? Coin.HEADS : Coin.TAILS;
    }
}
```

## （三） 过程截图

（1）全屏截图




（2）运行结果

1. 用户进行若干次的硬币扔掷

```
选择菜单
扔硬币 (输入 1)
显示结果 (输入 2)
退出 (输入 3)
请输入选择: 1
你扔到了正面

选择菜单
扔硬币 (输入 1)
显示结果 (输入 2)
退出 (输入 3)
请输入选择: 1
你扔到了正面

选择菜单
扔硬币 (输入 1)
显示结果 (输入 2)
退出 (输入 3)
请输入选择: 1
你扔到了反面

选择菜单
扔硬币 (输入 1)
显示结果 (输入 2)
退出 (输入 3)
请输入选择: 1
你扔到了反面

选择菜单
扔硬币 (输入 1)
显示结果 (输入 2)
退出 (输入 3)
请输入选择: 1
你扔到了正面
```

2. 显示结果，正反面扔到的次数几乎相同

```
Results:

HEADS: 13
TAILS: 12
```

题目 3：

（Random 2D Dynamic Art with Canvas) write an app that continuously fills 10 Circles of different colors on a Canvas. Use random positions, sizes, color.

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

## （二） 实现过程

（1）设计类

① Main 类中，继承了 Application 抽象类并重写了 start()方法，在该方法中，Stage 就是 JavaFX 工具中用来表示整个图形工具界面窗口的类，在该类中需要加入一个 Scene（场景）来进行填充，而所有的组件、元素都是构建在 Scene 中的。另外，在 JavaFX 8 中支持代码与布局和样式分离，所以在文件中通过 FXMLLoader 的 load()方法引入了一个外联的 sample.fxml 文件，在此 fxml 文件中就可以专心编写图形界面布局和组件相关功能。

```
import javafx.application.*;
import javafx.fxml.*;
import javafx.scene.*;
import javafx.stage.*;

public class Main extends Application {
    public static void main(String[] args) {
        //通过Application抽象类的launch()方法启动程序
        launch(args);
    }
    //重新start方法
    public void start(Stage stage) throws Exception {
        //使用FMXLLoader加载器，加载名为“sample.fxml”的文件
        Parent root = (Parent)FXMLLoader.load(getClass().getResource("sample.fxml"));
        Scene scene = new Scene(root);//创建一个场景
        stage.setTitle("Main");//为图形界面窗口设置标题
        stage.setScene(scene);//为图形界面窗口设置场景
        stage.show();//将图形界面设置为可见
    }
}
```

② sample.fxml 文件中设置了画布的高度为 700，宽度为 1000，控制文件为 Controlloer.java。

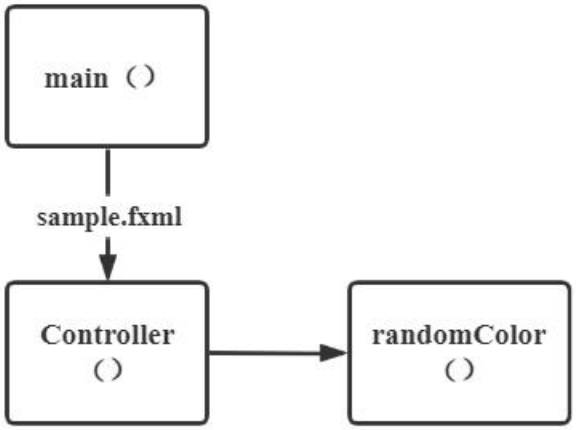
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- TimelineAnimation.fxml -->
<!-- FXML for a Circle that will be animated by the controller.  -->
<!--引入JavaFX工具相关的类-->
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.shape.Circle?>

<Pane id="Pane" fx:id="pane" prefHeight="700.0"
      prefWidth="1000.0" xmlns:fx="http://javafx.com/fxml/1"
      xmlns="http://javafx.com/javafx/8.0.60"
      fx:controller="Controller">
    <children>
    </children>
</Pane>
```

③ 在 Controller 类中，使用 SecureRandom 来生成随机数。通过给圆圈的横坐标纵坐标，半径，颜色设定随机值，来得到 10 个随机位置，大小，颜色的圆圈。

```
import java.security.SecureRandom;
import javafx.fxml.FXML;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.paint.Paint;
import javafx.scene.shape.Circle;

public class Controller {
    //用@FXML注解来表示属性可以被Fxml格式文件访问
    @FXML Pane pane;
    private SecureRandom random = new SecureRandom();//随机数生成器

    //绘制圆圈
    public void initialize () {
        for (int i = 0; i < 10; i++) {
            Circle circle = new Circle();//新建一个圆圈
            circle.setCenterX((this.random.nextInt(500) + 201));//随机设置圆圈的横坐标
            circle.setCenterY((this.random.nextInt(300) + 201));//随机设置圆圈的纵坐标
            circle.setRadius(this.random.nextInt(100));//随机设置圆圈的半径
            circle.setFill((Paint)randomColor());//随机填充圆圈的颜色
            this.pane.getChildren().add(circle);//添加圆圈到屏幕
        }
    }
    //随机生成颜色
    private Color randomColor() {
        //参数分别代表red, green, blue, opacity)
        return Color.rgb(this.random.nextInt(256), this.random.nextInt(256), this.random.nextInt(256),
                this.random .nextInt(101) / 100.0D);
    }
}
```

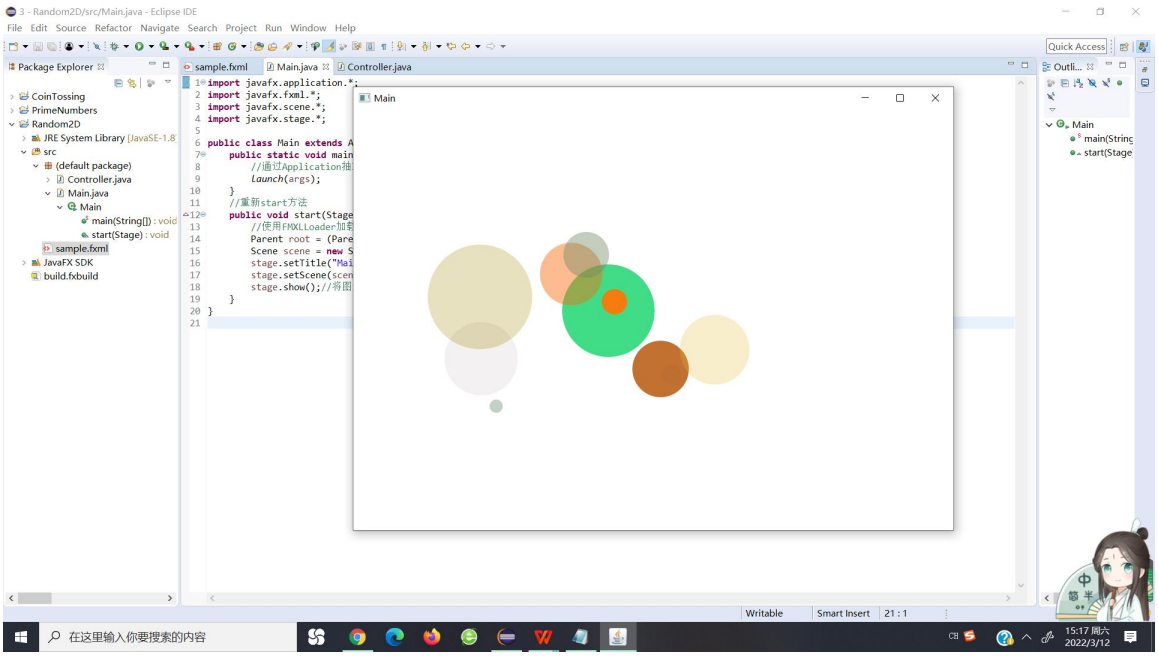
（2）调用关系

调用关系如下图，即 Main 方法可以通过加载 sample.fxml 来对 Controller 方法进行调用。Controller 方法中，通过 randomColor 方法来实现圆圈的随机颜色。



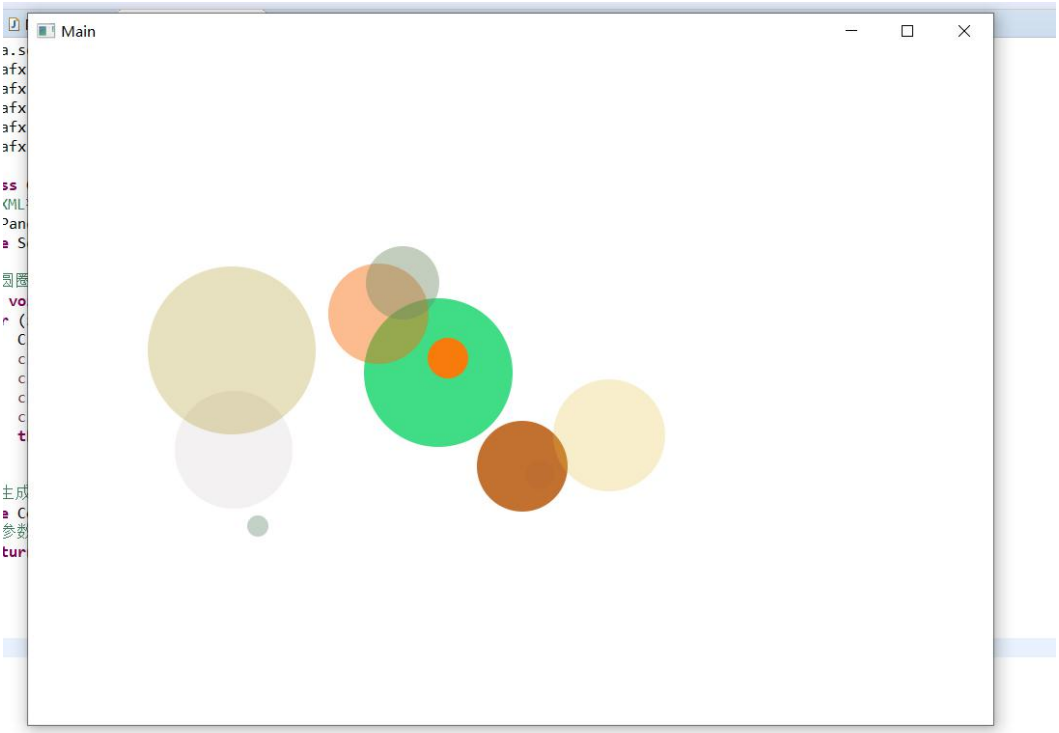

## （三） 过程截图

（1）全屏截图




（2）运行结果

得到 10 个随机位置，大小，颜色的圆圈。




二、拓展题目：

题目 1：

(6.35 Computer-Assisted Instruction) The use of computers in education is referred to as computer-assisted instruction (CAI). Write a program that will help an elementary school student learn multiplication. Use a SecureRandom object to produce two positive one-digit integers. The program should then prompt the user with a question, such as：

```
How much is 6 times 7?
```

The student then inputs the answer. Next, the program checks the student’ s answer. If it’s correct, display the message "Very good!" and ask another multiplication question. If the answer is wrong, display the message "No. Please try again." and let

the student try the same question repeatedly until the student finally gets it right. A separate method should be used to generate each new question. This method should be called once when the application begins execution and each time the user answers the question correctly.

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

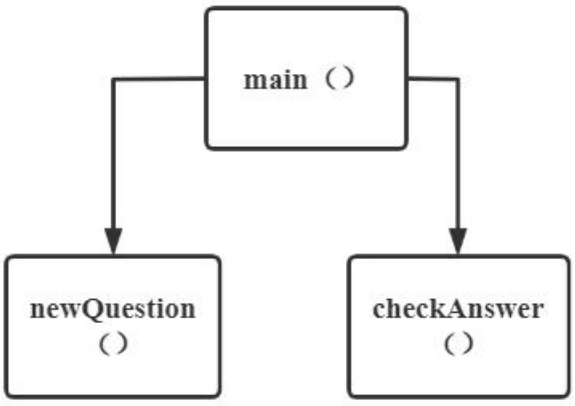
编程语言：Java；

## （二） 实现过程

（1）设计类

ComputerAssistedInstruction 类是关于学习乘法程序的 1.0 版本。在 main 方法中打印问题到控制台，实现用户输入答案，并对答案进行判断。main 方法调用一个单独的方法 newQuestion 来生成每个新的问题，调用 checkAnswer 方法来检查用户输入的答案是否正确。

（2）调用关系



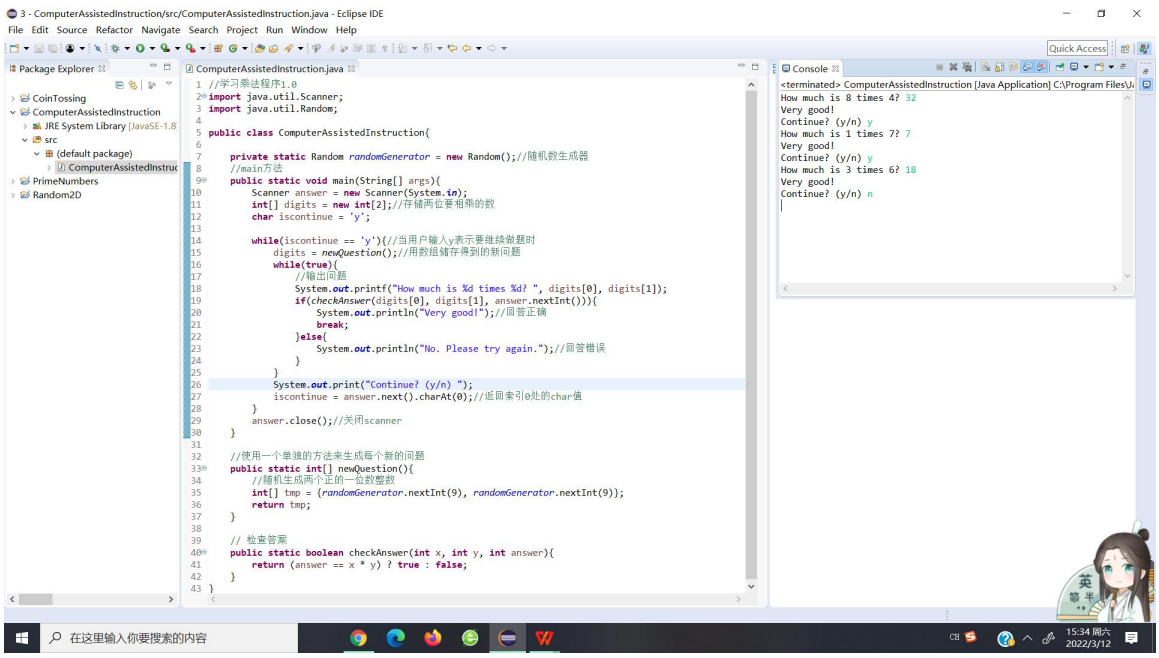
（3）实验代码

ComputerAssistedInstruction.java

```java
//学习乘法程序1.0
import java.util.Scanner;
import java.util.Random;

public class ComputerAssistedInstruction{

    private static Random randomGenerator = new Random();//随机数生成器
    //main方法
    public static void main(String[] args){
        Scanner answer = new Scanner(System.in);
        int[] digits = new int[2];//存储两位要相乘的数
        char iscontinue = 'y';

        while(iscontinue == 'y'){//当用户输入y表示要继续做题时
            digits = newQuestion();//用数组储存得到的新问题
            while(true){
                //输出问题
                System.out.printf("How much is %d times %d? ", digits[0], digits[1]);
                if(checkAnswer(digits[0], digits[1], answer.nextInt())){
                    System.out.println("Very good!");//回答正确
                    break;
                }else{
                    System.out.println("No. Please try again.");//回答错误
                }
            }
            System.out.print("Continue? (y/n) ");
            iscontinue = answer.next().charAt(0);//返回索引0处的char值
        }
        answer.close();//关闭scanner
    }

    //使用一个单独的方法来生成每个新的问题
    public static int[] newQuestion(){
        //随机生成两个正的一位数整数
        int[] tmp = {randomGenerator.nextInt(9), randomGenerator.nextInt(9)};
        return tmp;
    }

    // 检查答案
    public static boolean checkAnswer(int x, int y, int answer){
        return (answer == x * y) ? true : false;
    }
}
```

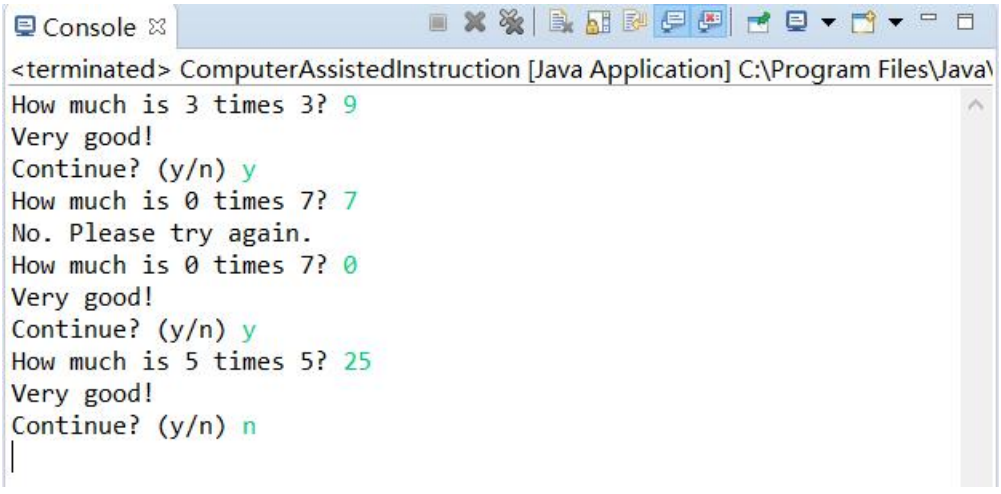
## （三） 过程截图

（1）全屏截图




（2）运行结果

程序提示用户出现一个问题，由学生输入答案。接下来，该程序会检查学生的答案。如果它是正确的，显示信息“Very good！”再问一个乘法法的问题。如果答案是错的，请显示信息“No. Please try again.”让学生反复尝试同样的问题，直到学生最终做对了。




题目 2：

(6.36 Computer-Assisted Instruction: Reducing Student Fatigue) One problem in CAI environments is student fatigue. This can be reduced by varying the computer's responses to hold the student's attention. Modify the program of Exercise 6.35 so that various comments are displayed for each answer as follows:

Possible responses to a correct answer:

```
Very good!
Excellent!
Nice work!
Keep up the good work!
```

Possible responses to an incorrect answer:

```
No. Please try again.
Wrong. Try once more.
Don't give up!
No. Keep trying.
```

Use random-number generation to choose a number from 1 to 4 that will be used to select one of the four appropriate responses to each correct or incorrect answer. Use a switch statement to issue the responses.

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

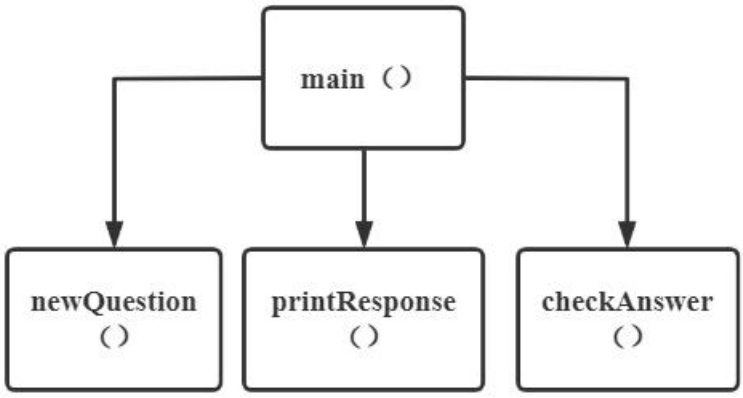
编程语言：Java；

## （二） 实现过程

（1）设计类

ReducingStudentFatigue 类是关于学习乘法程序的 2.0 版本。在 main 方法中打印问题到控制台，实现用户输入答案，并对答案进行判断。main 方法调用一个单独的方法 newQuestion 来生成每个新的问题，调用 checkAnswer 方法来检查用户输入的答案是否正确，调用 printResponse 方法来从数组随机选择回复并打印到控制台。使用随机数生成从 1 到 4 中选择一个数字，它将用于选择对每个正确或错误答案的四个适当的回答中的一个。使用开关语句来发出响应。

（2）调用关系




（3）实验代码

① Main 方法部分

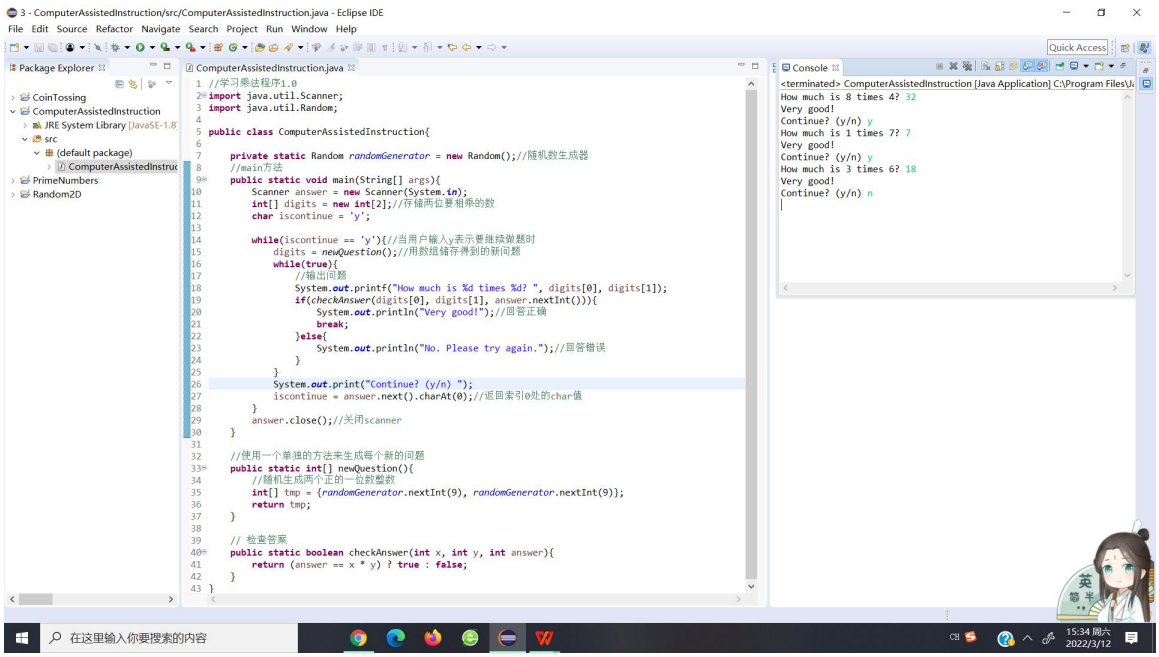
```java
//学习乘法程序2.0
import java.util.Scanner;
import java.util.Random;
import java.util.ArrayList;

public class ReducingStudentFatigue{
    private static Random randomGenerator = new Random();//随机数生成器
    //储存答案正确与错误时可能打印的反馈
    private static ArrayList<String> success = new ArrayList<String>();
    private static ArrayList<String> failure = new ArrayList<String>();
    //main方法
    public static void main(String[] args){
        Scanner answer = new Scanner(System.in);
        int[] digits = new int[2];//存储两位要相乘的数
        char iscontinue = 'y';
        success.add("Very Good!");
        success.add("Excellent!");
        success.add("You go girlfriend!!");
        success.add("Nice Work!");
        success.add("Keep up the good work!");
        failure.add("No. Please try again.");
        failure.add("Wrong. Try once more.");
        failure.add("Don't give up.");
        failure.add("No. Keep trying.");

        while(iscontinue == 'y'){//当用户输入y表示要继续做题
            digits = newQuestion();//用数组储存得到的新问题
            while(true){
                //输出问题
                System.out.printf("How much is %d times %d? ", digits[0], digits[1]);
                if(checkAnswer(digits[0], digits[1], answer.nextInt())){
                    printResponse("success");//回答正确，随机输出表扬
                    break;
                }
                else  printResponse("failure");//回答错误，随机输出安慰

            }
            System.out.print("Continue? (y/n) ");
            iscontinue= answer.next().charAt(0);//返回索引0处的char值
        }
        answer.close();//关闭scanner
    }
```

② 另外三个方法部分

```
    //使用一个单独的方法来生成每个新的问题
    public static int[] newQuestion(){
        //随机生成两个正的一位数整数
        int[] tmp = {randomGenerator.nextInt(9), randomGenerator.nextInt(9)};
        return tmp;
    }
    // 检查答案
    public static boolean checkAnswer(int x, int y, int answer){
        return (answer == x * y) ? true : false;
    }
    // 从数组随机选择回复并打印到控制台
    public static void printResponse(String condition){
        switch(condition){
            case "success":
                System.out.println(success.get(randomGenerator.nextInt(success.size())));
                break;
            case "failure":
                System.out.println(failure.get(randomGenerator.nextInt(failure.size())));
                break;
            default:
                break;
        }
    }
}
```

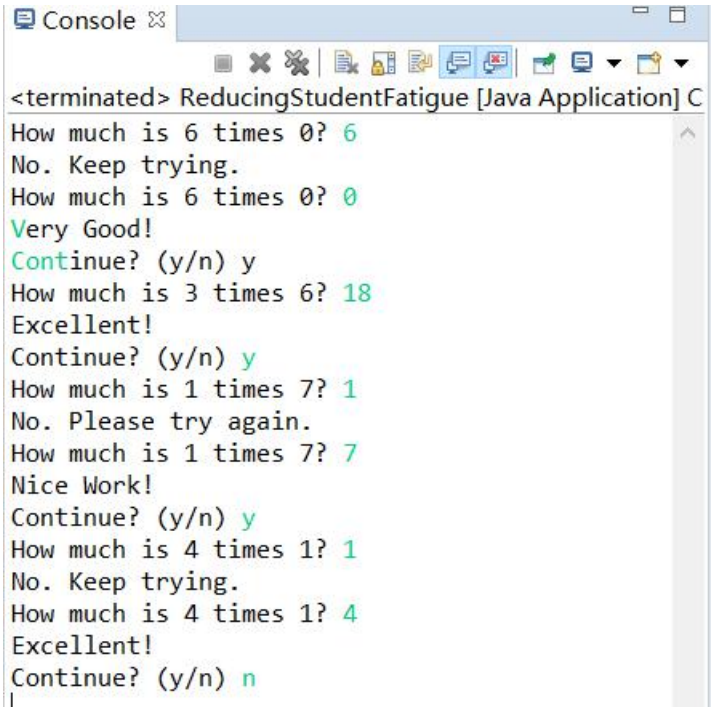
## （三） 过程截图

（1）全屏截图




（2）运行结果

2.0 版本中，改变了计算机的反应面对正确或者错误的答案的反应，程序将随机输出回答。




题目 3：

(6.37 Computer-Assisted Instruction: Monitoring Student Performance) More sophisticated computer-assisted instruction systems monitor the student's performance over a period of time. The decision to begin a new topic is often based on the student's success with previous topics. Modify the program of Exercise 6.36 to count the number of correct and incorrect responses typed by the student. After the student types 10 answers, your program should calculate the percentage that are correct. If the percentage is lower than 75%, display "Please ask your teacher for extra help.", then reset the program so another student can try it. If the percentage is 75% or higher, display "Congratulations, you are ready to go to the next level!", then reset the program so another student can try it.

## （一） 实验环境

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

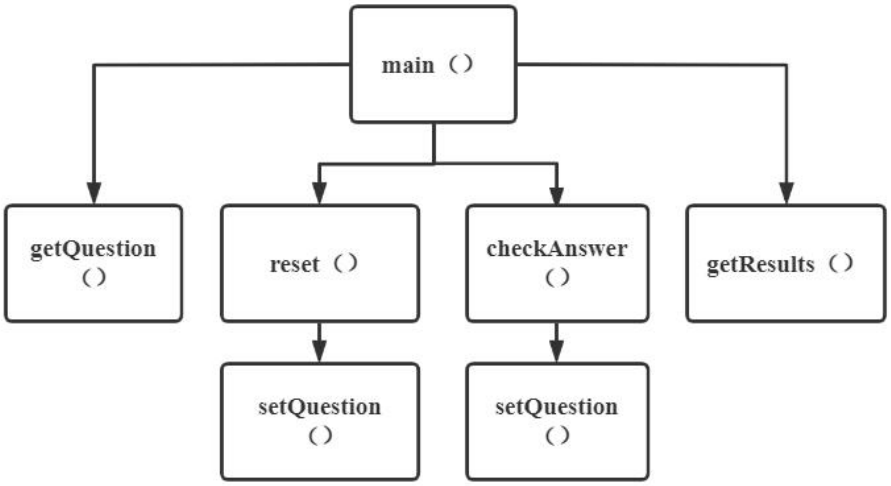
编程语言：Java；

## （二） 实现过程

（1）设计类

MonitoringStudentPerformance 类是关于学习乘法程序的 3.0 版本，将计算学生输入的正确和错误回答的数量，来给出不同的结果。在 main 方法中打印问题到控制台，实现用户输入答案，并对答案进行判断。main 方法调用一个单独的方法 setQuestion 来生成每个新的问题；调用 getQuestion 来打印问题；调用 checkAnswer 方法来检查用户输入的答案是否正确，随机选择回复并打印到控制台；调用 reset 方法重置分数，并开始新的一轮；调用 getResults 方法打印结果。

（2）调用关系




（3）实验代码

① Main 方法部分

MonitoringStudentPerformance.java

```java
//学习乘法程序3.0
import java.util.Random;
import java.util.Scanner;
import java.util.ArrayList;

public class MonitoringStudentPerformance{
    private Random randomGenerator = new Random();//随机数生成器
  //储存答案正确与错误时可能打印的反馈
    private ArrayList<String> success = new ArrayList<String>();
    private ArrayList<String> failure = new ArrayList<String>();

    private int[] values = new int[2];//存储两位要相乘的数
    private int score = 0;//储存一轮的分数

    //main方法
    public static void main(String[] args){
        Scanner answer = new Scanner(System.in);
        MonitoringStudentPerformance round = new MonitoringStudentPerformance();
        int count = 0;
        char iscontinue = 'y';
        while(iscontinue == 'y'){
            //一轮问题为10个
            while(count != 10){
                round.getQuestion();//打印问题
                round.checkAnswer(answer.nextInt());
                ++count;
            }
            round.getResults();//打印结果
            System.out.print("Continue? (y/n) ");
            iscontinue = answer.next().charAt(0);//返回索引0处的char值
            if(iscontinue == 'y') {
                count=0;//已回答问题个数重置回0
                round.reset();
            }
        }
        answer.close();//关闭scanner
    }
```

② 构造函数部分

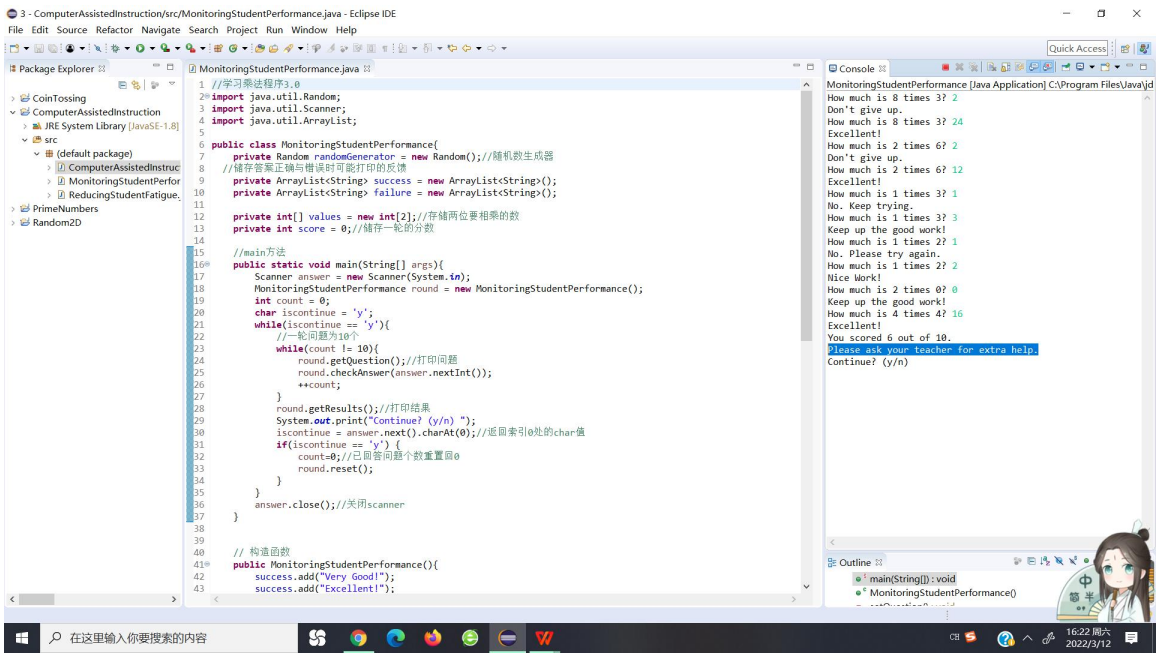
```java
// 构造函数
public MonitoringStudentPerformance(){
    success.add("Very Good!");
    success.add("Excellent!");
    success.add("Nice Work!");
    success.add("Keep up the good work!");
    failure.add("No. Please try again.");
    failure.add("Wrong. Try once more.");
    failure.add("Don't give up.");
    failure.add("No. Keep trying.");
    setQuestion(); // 生成新的问题的两个乘数
}
```

② 其他五个方法部分

```
    // 生成新的问题的两个乘数
    private void setQuestion(){
        values[0] = randomGenerator.nextInt(9);
        values[1] = randomGenerator.nextInt(9);
    }
    // 打印问题
    public void getQuestion(){
        System.out.printf("How much is %d times %d? ", values[0], values[1]);
    }
    //打印结果
    public void getResults(){
        System.out.printf("You scored %d out of 10.\n", score);
        if(score >= (10 * 0.75)){
            System.out.println("Congratulations you are ready to go the next level!");
        }else{
            System.out.println("Please ask your teacher for extra help.");
        }
    }
    //检查答案，随机生成反应
    public void checkAnswer(int answer){
        if(answer == values[0] * values[1]){
            System.out.println(success.get(randomGenerator.nextInt(success.size())));
            setQuestion();//如果正确，生成新问题
            ++score;//分数加1
        }
        else System.out.println(failure.get(randomGenerator.nextInt(failure.size())));
    }
    // 重置分数，并开始新的一轮
    public void reset(){
        score = 0;
        setQuestion();
    }
}
```

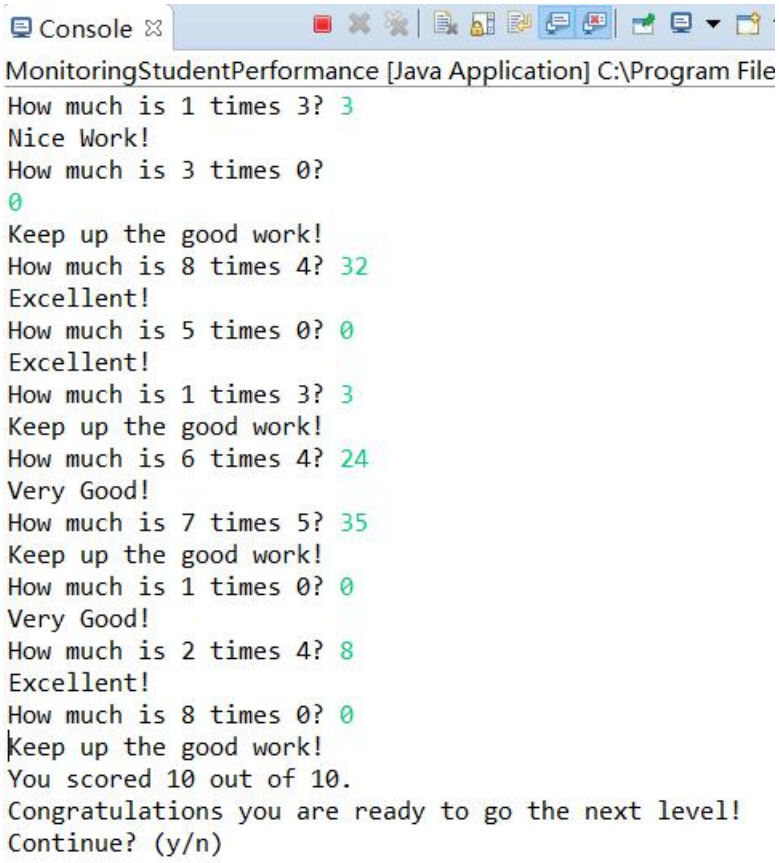
## （三） 过程截图

（1）全屏截图

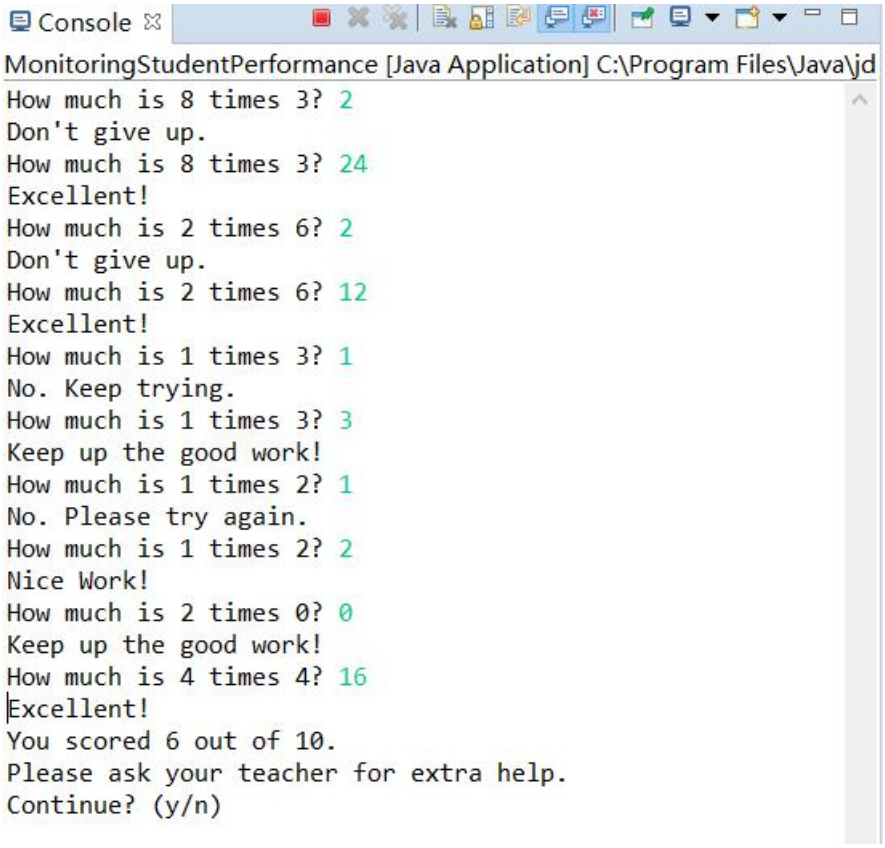



（2）运行结果

3.0 版本中，在学生输入 10 个答案后，程序将计算出正确的百分比。如果百分比是 75%或更高，将显示“Congratulations you are ready to go the next level!”，然后重置程序，让另一个学生可以尝试它。




如果比例低于 75%，将显示“Please ask your teacher for extra help.”。然后重置程序，让另一个学生可以尝试它。




## 三、实验总结与心得记录

在本次实验过程中，我练习了控制结构，熟悉了 java 的语法，熟悉了 java 类的定义，实例化和调用。我也熟悉了简单的枚举和简单的 JavaFX 图形界面，体会到了 JAVA 语言的优点。

# 四、源文件清单

1、项目 PrimeNumbers 用于**题目 1：PrimeNumbers**。

包含 PrimeNumbers.java 文件

2、项目 CoinTossing 用于**题目 2：CoinTossing。**

包含 CoinTossing.java 文件。

3、项目 Random2D 用于**题目 3：Random2D**。

包含 Main.java、Controller.java、sample.fxml 文件。

4、项目 ComputerAssisted Instruction 用于**拓展题目：Computer-Assisted Instruction**。

包含 ComputerAssistedInstruction.java、ReducingStudentFatigue.java、MonitoringStudentPerformance.java，分别代表学习乘法程序 1.0、2.0、3.0。