成绩	

重庆邮电大学

实验报告

2020-2021 学年第 2 学期

计算机科学导论

（第 2 次试验）

班级：　34082003

姓名：　黄凯升

学号：　2020215138

指导老师：　许汀汀

课程名称：　计算机科学导论

实验时间：　2021 年　4　月　8　日

实验地点：　综合实验大楼 A511/A512

1 实验名称

Loops and Files

2 实验目的

- Be able to convert an algorithm using control structures into Java
- Be able to write a while loop
- Be able to write an do-while loop
- Be able to write a for loop
- Be able to use the Random class to generate random numbers.
- Be able to use file streams for I/O
- Be able to write a loop that reads until end of file
- Be able to implement an accumulator and a counter

3 实验内容

Task#1 While loop

Just use a while loop and selection control structures to implement a given pseudo code into Java, simulating a rolling dice.

Task #2 Using Other Types of Loops

Rewrite the while loop in Task #1 into the equivalent forms: do-while loop and for loop.

Task #3 Writing Output to a File

Make a FileWriter and a PrintWriter object and write output whose numbers are formatted by DecimalFormat to a file. And also use try-catch statements.

Task #4 Calculating the Mean

Read a file whose name is given in standard input, with FileReader and BufferedReader. Parse double floating number from the line read from the file and calculate the mean.

Task #5 Calculating the Standard Deviation

Reopen the file, reinitialize variables and calculate the standard deviation using Math.

4 实验方法(原理、流程图)

The development environment is:

- OS: Ubuntu 20.04.2 LTS on Windows 10 (WSL1, Kernel build 19041)
- IDE/Editor: Visual Studio Code
- Java Runtime: OpenJDK 14.0.2 (build 14.0.2+12-Ubuntu-120.04)

For Task #1, it’s a simple task. Just read the pseudo code and implement the algorithm in Java, using while loop.

For Task #2, easy task. Rewrite the while loop into do-while loop and for loop.

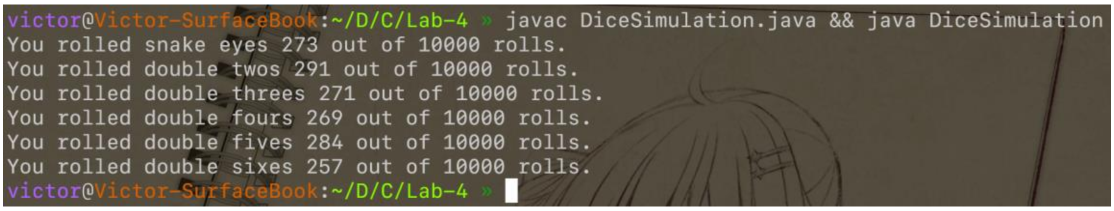
For Task #3, it’s required to make a FileWriter and PrintWriter. As we know opening a file is not a 100% successful operation. So, we need to use try-catch statement to handle the exception. And we should add throws clause to the main method to declare which types of exceptions it can throw. The usage of PrintWriter is similar to System.out. In the end, we close the writer.

For Task #4, we need to read data from a file with FileReader and BufferedReader. Also, we use a try-catch statement. When it opens the file, we read a line first and use a while loop with condition that the line is not null. Then keep reading, converting the line into double floating number and accumulate. Once finished reading, we can close it.

For Task #5, we have to reinitialize some variable and reopen the file. Just read in the same way as Task #4. And we use the built-in methods of Math to calculate the standard deviation.

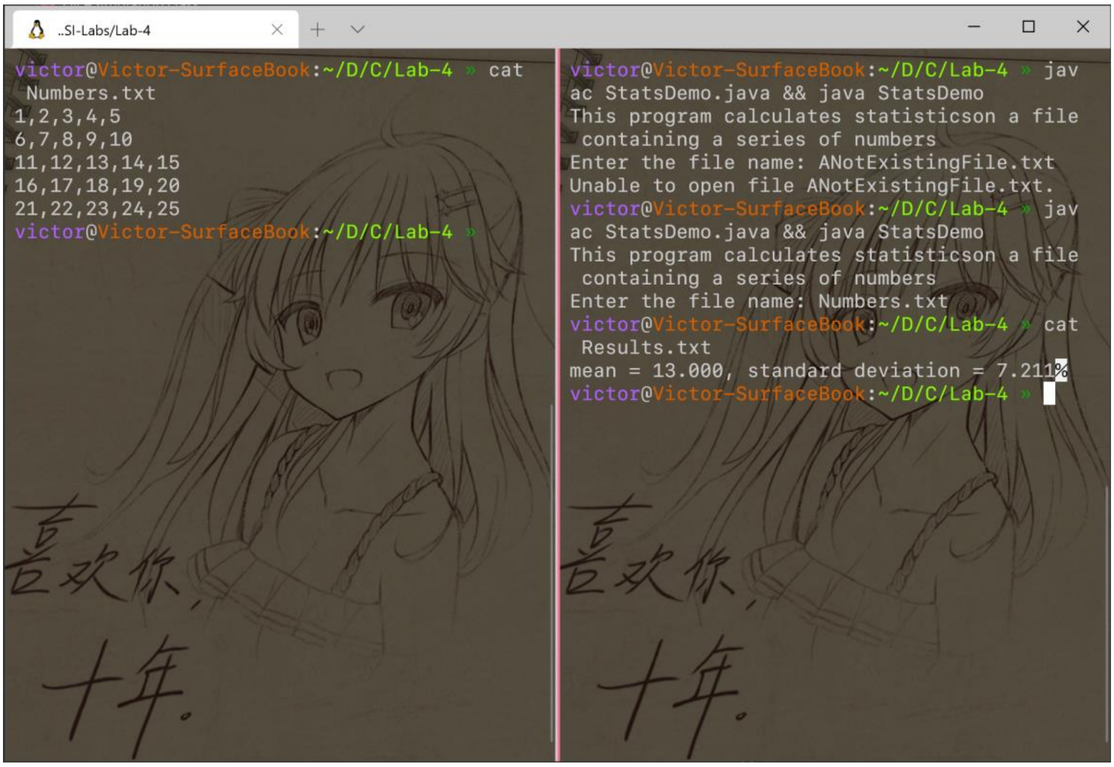
5 实验结论

The lab has finished successfully. The program can completely achieve all goals. Here are some screenshots with several test cases.



```
victor@Victor-SurfaceBook:~/D/C/Lab-4 » javac DiceSimulation.java && java DiceSimulation
You rolled snake eyes 273 out of 10000 rolls.
You rolled double twos 291 out of 10000 rolls.
You rolled double threes 271 out of 10000 rolls.
You rolled double fours 269 out of 10000 rolls.
You rolled double fives 284 out of 10000 rolls.
You rolled double sixes 257 out of 10000 rolls.
victor@Victor-SurfaceBook:~/D/C/Lab-4 »
```

The screenshot above tested Task #1. The output is valid.



```
victor@Victor-SurfaceBook:~/D/C/Lab-4 » cat
 Numbers.txt
1,2,3,4,5
6,7,8,9,10
11,12,13,14,15
16,17,18,19,20
21,22,23,24,25
victor@Victor-SurfaceBook:~/D/C/Lab-4 »
```

```
victor@Victor-SurfaceBook:~/D/C/Lab-4 » jav
ac StatsDemo.java && java StatsDemo
This program calculates statisticson a file
 containing a series of numbers
Enter the file name: ANotExistingFile.txt
Unable to open file ANotExistingFile.txt.
victor@Victor-SurfaceBook:~/D/C/Lab-4 » jav
ac StatsDemo.java && java StatsDemo
This program calculates statisticson a file
 containing a series of numbers
Enter the file name: Numbers.txt
victor@Victor-SurfaceBook:~/D/C/Lab-4 » cat
 Results.txt
mean = 13.000, standard deviation = 7.211%
victor@Victor-SurfaceBook:~/D/C/Lab-4 »
```

This screenshot tested an invalid filename, valid filename and a series of numbers. It works perfectly too. (The white backgrounded “%” that cat command’s output ends with is printed by the shell, which indicates that there’s no new line character in standard output. It’s not a character in file.)

6 实验体会和收获

Java provides a powerful standard library. We can use it to read from file, write to file and parse a line into number… so easily! And the loops have the same syntax as C++, which is very comfortable for C++ programmers.

Looking forward to more Java labs!

The source code is shown below.

7 程序代码

Task #1. DiceSimulation.java

```java
/**
 This class simulates rolling a pair of dice 10,000 times and
 counts the number of times doubles of are rolled for each different
 pair of doubles.
*/
import java.util.Random; //to use the random number generator

public class DiceSimulation {
    public static void main(String[] args) {
        final int NUMBER = 10000; // the number of times to roll the dice
        // a random number generator used in simulating rolling a dice
        Random generator = new Random();

        int die1Value; // number of spots on the first die
        int die2Value; // number of spots on the second die
        int count = 0; // number of times the dice were rolled
        int snakeEyes = 0; // number of times snake eyes is rolled
        int twos = 0; // number of times double two is rolled
        int threes = 0; // number of times double three is rolled
        int fours = 0; // number of times double four is rolled
        int fives = 0; // number of times double five is rolled
        int sixes = 0; // number of times double six is rolled
        // ENTER YOUR CODE FOR THE ALGORITHM HERE

        while (count < NUMBER) {
            die1Value = generator.nextInt(6) + 1;
            die2Value = generator.nextInt(6) + 1;
            if (die1Value == die2Value) {
                if (die1Value == 1)
                    snakeEyes++;
                else if (die1Value == 2)
                    twos++;
                else if (die1Value == 3)
                    threes++;
                else if (die1Value == 4)
                    fours++;
                else if (die1Value == 5)
                    fives++;
                else if (die1Value == 6)
                    sixes++;
            }
            count++;
        }

        System.out.println("You rolled snake eyes " + snakeEyes + " out of " + count + " rolls."
);
        System.out.println("You rolled double twos " + twos + " out of " + count + " rolls.");
        System.out.println("You rolled double threes " + threes + " out of " + count + " rolls."
);
        System.out.println("You rolled double fours " + fours + " out of " + count + " rolls.");
        System.out.println("You rolled double fives " + fives + " out of " + count + " rolls.");
        System.out.println("You rolled double sixes " + sixes + " out of " + count + " rolls.");
    }
}
```

Task #2. DiceSimulation.java (Difference of Loops)

```java
// ENTER YOUR CODE FOR THE ALGORITHM HERE
// do-while loop
do {
    die1Value = generator.nextInt(6) + 1;
    die2Value = generator.nextInt(6) + 1;
    if (die1Value == die2Value) {
        if (die1Value == 1)
            snakeEyes++;
        else if (die1Value == 2)
            twos++;
        else if (die1Value == 3)
            threes++;
        else if (die1Value == 4)
            fours++;
        else if (die1Value == 5)
            fives++;
        else if (die1Value == 6)
            sixes++;
    }
    count++;
} while (count < NUMBER);

// ENTER YOUR CODE FOR THE ALGORITHM HERE
// for loop
for (count = 0; count <= NUMBER; count++) {
    die1Value = generator.nextInt(6) + 1;
    die2Value = generator.nextInt(6) + 1;
    if (die1Value == die2Value) {
        if (die1Value == 1)
            snakeEyes++;
        else if (die1Value == 2)
            twos++;
        else if (die1Value == 3)
            threes++;
        else if (die1Value == 4)
            fours++;
        else if (die1Value == 5)
            fives++;
        else if (die1Value == 6)
            sixes++;
    }
}
```

Task #3,4,5. StatsDemo.java

```
import java.text.DecimalFormat; //for number formatting
import java.util.Scanner; //for keyboard input
//ADD AN IMPORT STATEMENT HERE //for using files
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.FileNotFoundException;
import java.lang.Math;

public class StatsDemo {
    public static void main(String[] args) throws FileNotFoundException, IOException { // ADD A
THROWS CLAUSE HERE
        double sum = 0; // the sum of the numbers
        int count = 0; // the number of numbers added
        double mean = 0; // the average of the numbers
        double stdDev = 0; // the standard deviation of the numbers
        String line; // a line from the file
        double difference; // difference between the value and the mean
        // create an object of type Decimal Format
        DecimalFormat threeDecimals = new DecimalFormat("0.000");
        // create an object of type Scanner
        Scanner keyboard = new Scanner(System.in);
        String filename; // the user input file name
        // Prompt the user and read in the file name
        System.out.println("This program calculates statistics" + "on a file containing a series
 of numbers");
        System.out.print("Enter the file name: ");
        filename = keyboard.nextLine();
        // ADD LINES FOR TASK #4 HERE
        // Create a FileReader object passing it the filename
        // Create a BufferedReader object passing it the FileReader object.
        FileReader fileReader = null;
        BufferedReader bufReader = null;
        try {
            fileReader = new FileReader(filename);
            bufReader = new BufferedReader(fileReader);
        } catch (FileNotFoundException e) {
            System.err.println("Unable to open file " + filename + '.');
            System.exit(1);
        }
        // priming read to read the first line of the file
        // create a loop that continues until you are at the end of the file
        // convert the line to double value, add the value to the sum
        // increment the counter
        // read a new line from the file
        // close the input file
        // store the calculated mean
        line = bufReader.readLine();
        while (line != null) {
            for (var numStr : line.split(",")) {
                sum += Double.parseDouble(numStr);
                count++;
            }
            line = bufReader.readLine();
        }
        bufReader.close();
        mean = sum / count;

        // ADD LINES FOR TASK #5 HERE
        // create a FileReader object passing it the filename
        // create a BufferedReader object passing it the FileReader object.
        try {
            fileReader = new FileReader(filename);
            bufReader = new BufferedReader(fileReader);
        } catch (FileNotFoundException e) {
            System.err.println("Unable to open file " + filename + '.');
            System.exit(1);
        }
        // reinitialize the sum of the numbers
        // reinitialize the number of numbers added
        sum = 0;
        count = 0;
        // priming read to read the first line of the file
        // loop that continues until you are at the end of the file
        // convert the line into a double value and subtract the mean
        // add the square of the difference to the sum
        // increment the counter
        // read a new line from the file
        // close the input file
        // store the calculated standard deviation
        line = bufReader.readLine();
        while (line != null) {
            for (var numStr : line.split(",")) {
                difference = Double.parseDouble(numStr) - mean;
                sum += Math.pow(difference, 2);
                count++;
            }
            line = bufReader.readLine();
        }
        bufReader.close();
        stdDev = Math.sqrt(sum / count);

        // ADD LINES FOR TASK #3 HERE
        // create an object of type FileWriter using “Results.txt”
        // create an object of PrintWriter passing it the FileWriter object.
        // print the results to the output file
        // close the output file
        FileWriter fileWriter = null;
        PrintWriter writer = null;
        try {
            fileWriter = new FileWriter("Results.txt");
            writer = new PrintWriter(fileWriter);
        } catch (IOException e) {
            System.err.println("Unable to write to Results.txt");
            System.exit(1);
        }
        writer.print("mean = " + threeDecimals.format(mean));
        writer.print(", standard deviation = " + threeDecimals.format(stdDev));
        writer.close();
    }
}
```