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 5 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 5 月 6 日

实验地点：综合实验大楼 A511/A512

1 实验名称

Arrays

2 实验目的

- Be able to declare and instantiate arrays
- Be able to fill an array using a for loop
- Be able to access and process data in an array
- Be able to write a sorting method
- Be able to use an array of objects

3 实验内容

Task#1 Average Class

Create a class called Average, which stores an array of integers and the arithmetic average of the array. The constructor should prompt to input the data. Write a method to calculate mean and a method to do a descending sort with selection sort algorithm. Also write a `toString` method to make a string description of the class, including sorted array and mean.

Task #2 Average Driver

Create a class called AverageDriver, which contains a main method. And it just creates an Average instance and prints it.

Task #3 Arrays of Objects

Declare an array of Songs, and Song is a pre-defined class. Fill the array with songs read from file. And then print them.

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, we should define a class which has two private fields: an integer array called data and a real number called mean. In the constructor, because Java uses 0 as the start of index but people use 1. We should print the value which Java index is added by 1 to the user. Then it should call `selectionSort` and `calculateMean` methods. To implement the method `calculateMean`, we can just get a sum of array and make a division. And for the method `selectionSort`, we just scan the array for *n* times (*n* is the length of array). Each time we scan the rest of array to find a maximum value, and then swap it with the value where we just started to scan. The time complexity of selection sort is $\mathrm{O}(n^2)$. For the method `toString`, an important part is to make a description string of the array. We can't get expected result with `Array.toString`. And to avoid implicit type conversion, Java does not support joining an integer array with string. So, I used StringJoiner to do this stuff. And in the end, I used `String.format` to make a formatted string.

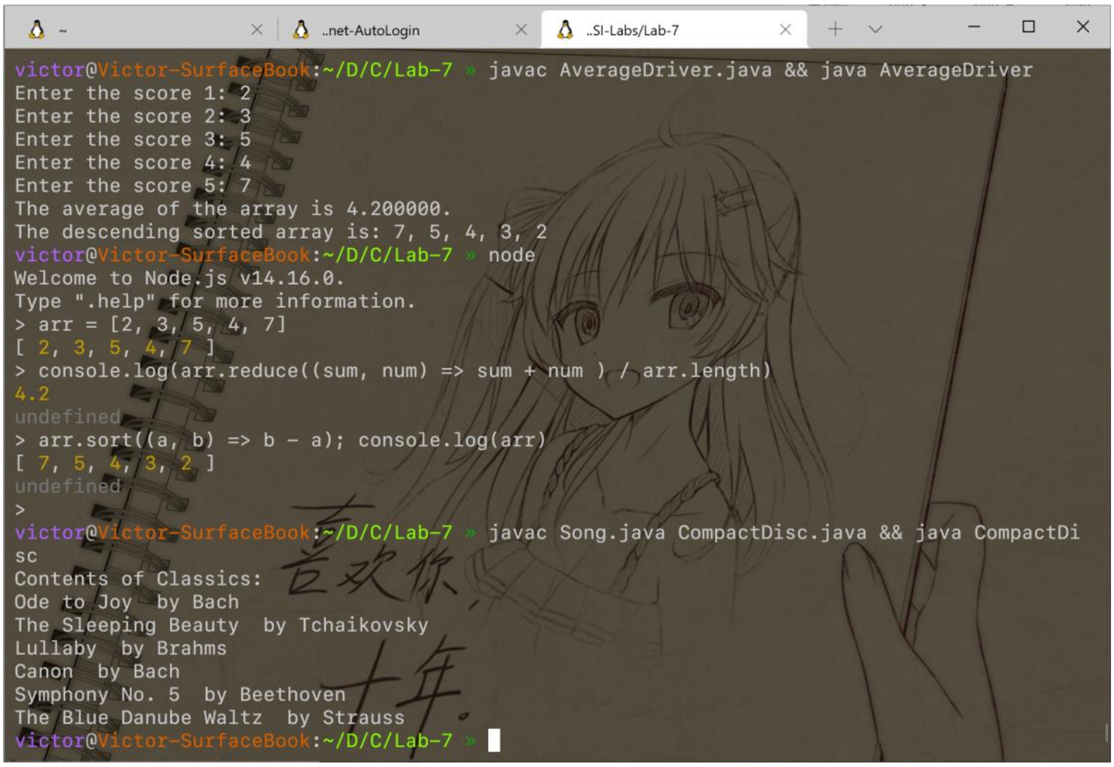
For Task #2, just simply define a class called AverageDriver, create an instance of Average and then print it.

For Task #3, we should create an array of Songs. It's not an array of primitive type. So, we should use new operator to create the array. And for the rest we can use it in the same way as arrays of primitive type.

5 实验结论

The lab has finished successfully. The programs can completely achieve all goals. Here is the screenshot.

I used Node.js to calculate the mean and make sorted array.



```
victor@Victor-SurfaceBook:~/D/C/Lab-7 » javac AverageDriver.java && java AverageDriver
Enter the score 1: 2
Enter the score 2: 3
Enter the score 3: 5
Enter the score 4: 4
Enter the score 5: 7
The average of the array is 4.200000.
The descending sorted array is: 7, 5, 4, 3, 2
victor@Victor-SurfaceBook:~/D/C/Lab-7 » node
Welcome to Node.js v14.16.0.
Type ".help" for more information.
> arr = [2, 3, 5, 4, 7]
[ 2, 3, 5, 4, 7 ]
> console.log(arr.reduce((sum, num) => sum + num ) / arr.length)
4.2
undefined
> arr.sort((a, b) => b - a); console.log(arr)
[ 7, 5, 4, 3, 2 ]
undefined
>
victor@Victor-SurfaceBook:~/D/C/Lab-7 » javac Song.java CompactDisc.java && java CompactDi
sc
Contents of Classics:
Ode to Joy  by Bach
The Sleeping Beauty  by Tchaikovsky
Lullaby  by Brahms
Canon  by Bach
Symphony No. 5  by Beethoven
The Blue Danube Waltz  by Strauss
victor@Victor-SurfaceBook:~/D/C/Lab-7 »
```

6 实验体会和收获

Array is a very basic data structure in many programming languages. It enables the programming language to store a bunch of data. And with the loop, programmers can proceed the data easily. And we also practiced a very simple sorting algorithm. I think this lab is also a simple introduction to data structures and algorithms.

7 程序代码

AverageDriver.java:

```java
import java.util.Scanner;
import java.util.StringJoiner;

class Average {
    private int data[];
    private double mean;

    public Average() {
        this.data = new int[5];
        Scanner scanner = new Scanner(System.in);
        for (int i = 0; i < this.data.length; i++) {
            System.out.printf("Enter the score %d: ", i + 1);
            data[i] = scanner.nextInt();
        }
        calculateMean();
        selectionSort();
    }

    public void calculateMean() {
        int sum = 0;
        for (int i : this.data)
            sum += i;
        this.mean = 1.0 * sum / this.data.length;
    }

    public String toString() {
        StringJoiner joiner = new StringJoiner(", ");
        StringBuilder builder = new StringBuilder();
        for (int i : this.data)
            joiner.add(String.valueOf(i));
        return String.format("The average of the array is %f.\nThe descending sorted array is: %
s",
                                this.mean, joiner.toString());
    }

    public void selectionSort() {
        for (int i = 0; i < this.data.length; i++) {
            int val = this.data[i], pos = i;
            for (int j = i; j < this.data.length; j++) {
                if (this.data[j] > val) {
                    pos = j;
                    val = this.data[j];
                }
            }
            this.data[pos] = this.data[i];
            this.data[i] = val;
        }
    }
}

public class AverageDriver {
    public static void main(String[] args) {
        Average avg = new Average();
        System.out.println(avg);
    }
}
```

CompactDisc.java:

```
/*This program creates a list of songs for a CD by reading from a file*/
import java.io.*;

public class CompactDisc {
    public static void main(String[] args) throws IOException {
        FileReader file = new FileReader("Classics.txt");
        BufferedReader input = new BufferedReader(file);
        String title;
        String artist;

        // Declare an array of songs, called cd, of size 6
        Song cd[] = new Song[6];

        for (int i = 0; i < cd.length; i++) {
            title = input.readLine();
            artist = input.readLine();
            // fill the array by creating a new song with
            // the title and artist and storing it in the
            // appropriate position in the array
            cd[i] = new Song(title, artist);
        }

        System.out.println("Contents of Classics:");
        for (int i = 0; i < cd.length; i++) {
            // print the contents of the array to the console
            System.out.print(cd[i]);
        }
    }
}
```