Report of the Coursework 1 Assignment

Name: Lee Boon Kah

Student ID: 20297564

Module: Programming Paradigms (COMP1029)

School: University of Nottingham Malaysia

Lecturer : Dr. Mohammad Aazam

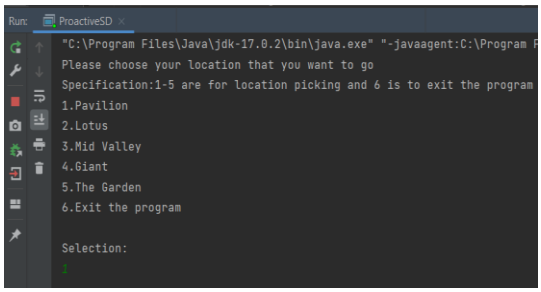
Report

First and foremost, I had learned a lot of advanced knowledge about the function of java and have some idea about the use of java in my future when building the program in this coursework. The first thing that I learned is the java inner class that is used in the class LocationDatabase. The java inner class is also a nest class and there are three types of the inner classes which are the private inner class, static inner class, and a normal inner class. In fact, I used the normal class in the class LocationDatabase but the IntelliJ application had given me a suggestion to change the normal inner class to the static class. Therefore, I have done some research on it and I understand the difference between the normal inner class and the static inner class. The difference between the normal inner class and static inner class is the normal inner class still can access other members of the enclosing class while the static inner class do not have access to other members of other classes. Besides, we need to declare one more line in the normal inner class compared to the static inner class.

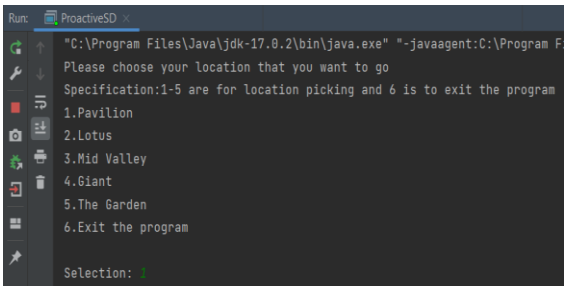
Secondly, I also found something that is interesting which is related to the print function. When using the scanner after the System.out.println() statrement, the user will be forced to insert their input at the next line of the printed argument. The solution to solve this is to use the System.out.print() statement. By using System.out.print() statement, the user is allowed to insert the input beside the printed argument. In my opinion, it looks cleaner when inserting the input beside the printed argument.

Example:

i. System.out.println() statement



ii. System.out.print() statement



Moreover, I had used the java try and catch keywords that come in pairs to solve the invalid input of the user at the menu. The problem of the try and catch statements are they only check the error of the input one time.Then, the try and catch statement will unable to use once the user inserted the invalid input again and it will lead the program to terminate immediatedly. By applying the do-while loop, I had solve the problem and the loop will keep looping and terminate the loop once the user inserted the valid input. In the prompting for the user to enter the distance, I had use the do-while loop only and if statement to prevent the user input the negative number and string. In this case, I did not use the try and catch statement and I use the do while loop.

Method of error handling in the menu:

```
    do {
        System.out.println(menu); //print the menu
        System.out.println("\nSelection: ");  //prompt for the user to input the selection
        //error handling once the user had inserted the string or alphabet
        try {
            choice_input = Integer.parseInt(sc.nextLine());
            choice = choice_input; //assign the choice_input's value into the choice which is the global variable
        } catch (Exception e){
            choice = 0;    //assign the choice to zero which is a default value
        }
```

Condition of the do-while loop:

```
    //condition of the do-while loop
    } while ((choice < 1) || (choice > 6) ||
```

Method of error handling in distance:

```
do{//error handling in North
    System.out.println("North:");
    while (!sc.hasNextDouble()) {//check the input is a number or not
        System.out.println("That's not a number! Please enter a positive number!\nNorth:");
        sc.next();
    }
    distance_north = sc.nextDouble();
    if(distance_north<=0){ //make sure that the input is a positive number
        System.out.println("Please reenter a positive number!");
    }
}while (distance_north <= 0);
```

Moreover, there are two ways to generate the random boolean value that I learn from this coursework. The first way is a common method that uses the nextBoolean() method of the java.util.Random class. It will randomly generate the boolean value which is either true or false. The second method is the way that I found it from the online resource through the website called Stack Overflow. In this method, the boolean will still randomly generate but this way includes the concept of probability.This method will return a pseudorandom double greater than or equal to 0.0 and less than 1.0 which means that true and false will randomly generate too.

First method:

```
import java.util.Random;

Random random = new Random();
random.nextBoolean();
```

Second method:

```
public static boolean getRandomBoolean() {
    return Math.random() < 0.5;
}
```

Last but not least, there are many alternative ways to do the java program. In my program, I can change the do-while loop to the while loop and perform the same function but the while loops are controlled by entrance and do while loops are controlled by exit. In other words, while loop is an entry-controlled loop and the do-while is an exit-controlled loop. In addition, I also can change the switch case statement to the else if statement. In my opinion, switch case statements look more clearer that the else if statement. There are many ways to do it but we can choose the way that we are confident in.