Lab 4

Using Loop Constructs

Objectives

Upon completion of this lab, you should be able to:

- Complete review questions
- Use `while` and `do/while` loops in a Java program
- Use `for` loops in a Java program

Lab Overview

In this lab, you complete the review questions and three exercises.

- In the first exercise, you create `while` loop constructs.
- In the second exercise, you create `for` loop constructs.
- In the third exercise, you create `do/while` loop constructs.

Completing Review Questions

Complete the following questions:

1. Fill in the blank of the following sentence with one of the options given below. _____ enables you to check and recheck a decision to execute and re-execute a block of code.
   a. Classes
   b. Objects
   c. Loops
   d. Methods
2. Which type of loop allows you to declare a variable as part of its construct:
   a. The `do/while` loop
   b. The `while` loop
   c. The nested `while` loop
   d. The `for` loop
3. Which of the following types of loops is a one/many iterative loop:
   a. The `while` loop
   b. The nested `while` loop
   c. The `do/while` loop
   d. The `for` loop
4. State whether the following statement is true or false:
   You should prefer to use the for loop to step through statements a pre-defined number of times.

Exercise 1: Using the while Loop

The objective of this exercise is to create classes that use `while` loop constructs.

Preparation

Ensure that `CounterTest.java` file and `SequenceTest.java` file exists in the `SL110/exercises/07_loops/exercise1` directory. This is your working directory.

This exercise has two tasks. In each task you create a class and use the while loops wherever applicable. The tasks are:

- "Task 1 – Writing a Class That Uses a `while` Loop"
- "Task 2 – Writing Another Class That Uses a `while` Loop"

Task 1 – Writing a Class That Uses a while Loop

In this task, you write a class called `Counter` and use a `while` loop to display the value of a count variable that is incremental in steps of one. Follow these steps to create your class:

1. Open the terminal window and go to your working directory. Using an editor, create a class called `Counter` with three member variables called `MAX_COUNT`, `step`, and `count`. Assign the value 50 to `MAX_COUNT` and the value 1 to count and `step`. Ensure that `MAX_COUNT` is declared as a `final` variable.
2. Create a `public displayCount` method in the class that does not accept any argument and returns `void`. For example:
   ```
   public void displayCount(){
   ```
3. Create a `while` loop in the method with the following characteristics:
   a. *Boolean expression*: Repeat if the value of `count` is less than or equal to the value of `MAX_COUNT`
   b. *Code block*:
      - Print the value of the count variable
      - Increment the value of count by step value. For example:
        ```
        count=count + step;
        ```
4. Save and compile the `Counter.java` file.

5. Compile the CounterTest.java file that is provided in your working directory. Execute the CounterTest class file.
6. Verify if your class prints all the numbers between 1 and 50.
7. You can modify the value of the step variable to modify the output of the Counter class. Perform the following steps:
   a. Open the CounterTest.java file for modification.
   b. Before the displayCount method is invoked, assign a value of 2 to the step member variable of the Counter class. Use the object of the Counter class to access the step member variable.
   c. Complete steps 3 - 4 and verify the output. The output should be: 1 3 5 7.....49
8. Repeat Step 6, for the step value of 3. step is the member variable of Counter class.

Task 2 – Writing Another Class That Uses a while Loop

In this task, you write a class named Sequence1 that displays a sequence starting with the numbers 0 and 1. Successive numbers in the sequence are the sum of the previous two numbers. For example: 0 1 1 2 3 5 8 13 21..... This sequence is also called the Fibonacci series.

Follow these steps to write your class:

1. Go to your working directory and open an editor.
2. Create a class called Sequence1 with three variables called firstNumber, secondNumber, and nextNumber. Assign the values of 0 and 1 to the firstNumber and secondNumber variables, respectively.
3. Create a public method called displaySequence with no return type and no argument, that will perform the following actions:
   a. Print the value of firstNumber, secondNumber to start with the sequence.
   b. Calculate the sum of firstNumber and secondNumber and assign the sum to the nextNumber variable.

4. Create a `while` loop in the method with the following characteristics:
   - *Boolean expression*: Repeat if the value of `nextNumber` is less than or equal to 100
   - *Code block*:
     - Print the value of the `nextNumber` variable.
     - Assign the value of `secondNumber` to `firstNumber` and the value of `nextNumber` to `secondNumber` variables.
     - Recalculate the value of `nextNumber` to be the sum of `firstNumber` and `secondNumber`.
5. Save and compile the `Sequence1.java` file.
6. Compile the `SequenceTest`.java file provided in your working directory and execute the `SequenceTest` class file.
7. Verify the output. The result should look like: 0 1 1 2 3 5 8 13 21 34 55 89

Exercise 2: Using the for loop

The objective of this exercise is to use the `for` loop in a class.

Preparation

Ensure that the `CounterTwoTest.java` file and the `Sequence2Test.java` file exist in the `SL110/exercises/07_loops/exercise2` directory. This is your working directory.

This exercise has two tasks. In each task you create a class and use the for loop where ever applicable.

- "Task 1 – Writing a Class That Uses the `for` Loop"
- "Task 2 – Modifying a `while` Loop to a `for` Loop"

Task 1 – Writing a Class That Uses the for Loop

In this task, you write a class that counts from 1 to a constant number and displays all the numbers divisible by 12 between them.

Complete the following steps to write your class:

1. Go to the working directory and open an editor.
2. Write a class called `CounterTwo` containing a member variable called `MAX_COUNT`. Assign the values of 100 to the variable.
3. Create a `public` method called `displayCount` with no argument and no return type, which:
   - Counts from 1 to the value of the `MAX_COUNT` constant, using a `for` loop. Increment the value of the loop variable by 1.
   - Displays the value of the loop variable if it is divisible by 12.
4. Save your class as `CounterTwo.java` file and compile the class.
5. Compile `CounterTwoTest.java` file and execute it to verify the output. The result should look like: 12 24 36 48 60 72 84 96

Task 2 – Modifying a while Loop to a for Loop

In this task, you modify the class that you created in Exercise 1, Task 2 and replace the `while` loop with a `for` loop.

Complete the following steps:

1. Create a class `Sequence2` similar to `Sequence1` as in Exercise 1, Task 2 and save it as `Sequence2.java` in your working directory.
2. Create an additional final member variable, `SEQUENCE_COUNT`, and assign 10 to it. Ensure that the other member variables remain unchanged.
3. In the `displaySequence` method, modify the `while` loop to a `for` loop such that only the first 10 values of the fibonacci series are displayed. The result should look like: 0 1 1 2 3 5 8 13 21 34.
4. Save and compile the `Sequence2.java` file.
5. Compile the provided `Sequence2Test.java`, execute it, and verify the output.

Exercise 3: Using the do/while Loop

The objective of this exercise is to write a class that uses a do/while loop.

Preparation

Ensure that the DiceTest.java file exists in the SL110/exercises/07_loops/exercise3 directory. This directory is your working directory.

Task – Writing a Class Using the do/while Loop

In this task, you write a class that simulates a simple game of throwing two dice. Follow these steps to write your class:

1. Go to the working directory.
2. Write a class called Dice containing two member variables diceNumber1 and diceNumber2 to hold the dice numbers.
3. Create a public method called throwDice with no argument and no return type and add a do/while loop with the following characteristics:
   - *Code block*: Generate a random number between 1 and 6 and assign it to diceNumber1 variable. Repeat the same for diceNumber2 variable. For example:
     diceNumber1=(int)(Math.random()*6) +1;
     Print the value of diceNumber1 and diceNumber2 variables.
   - *Boolean expression*: Repeat if diceNumber1 is equal to diceNumber2.
4. Save and compile the Dice class.
5. Use the DiceTest.java file to test the result of the Dice class.
6. Verify that the values of diceNumber1 and diceNumber2 are displayed.

Exercise Summary

Take a few minutes to identify what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications