UUM

Universiti Utara Malaysia

UNIVERSITI UTARA MALAYSIA

SECOND SEMESTER SESSION A242

STIWK 3014 REAL TIME PROGRAMMING

(GROUP A)

Assignment 1

Lecturer: Dr. Ruzita binti Ahmad

Name: Chong Mun Kei

Matric No: 298767

Github Link: https://github.com/Chong0508/RealTime.git

Source Code

```
package Assignment1;


import java.io.*;

import java.util.Scanner;

import java.util.concurrent.atomic.AtomicInteger;


public class Assignment1 {

   public static void main(String[] args) {

       Scanner scan = new Scanner(System.in);


       // Testing path:
src/main/java/Assignment1/Exercise4/src/main/java/org/example

       System.out.println("(Testing path:
src/main/java/Assignment1/Exercise4/src/main/java/org/example)")
;

       System.out.print("Enter the directory path to check for
Java files: ");

       String path = scan.nextLine();

       path = path.replace('"',' ').trim();

       path = path.replace('\\','/');

       System.out.println("Path detected: "+ path);


       scan.close();


       File folder = new File(path);

       File[] listOfFiles = folder.listFiles((dir, name) ->
name.endsWith(".java"));
```

```
        if (listOfFiles == null) {
            System.out.println("Invalid directory path or no Java
files found!");
            return;
        }

        int javaFileCount = listOfFiles.length;
        AtomicInteger issuesCount = new AtomicInteger(0);
        Thread[] threads = new Thread[javaFileCount];

        System.out.println("Java File Names: ");
        for (int i = 0; i < javaFileCount; i++) {
            final File file = listOfFiles[i];
            System.out.println((i+1)+". "+file.getName());

            threads[i] = new Thread(() -> {
                try (Scanner scanner = new Scanner(file)) {
                    while (scanner.hasNextLine()) {
                        String line =
scanner.nextLine().toUpperCase();
                        if (line.contains("// SOLVED")) {
                            issuesCount.incrementAndGet();
                        }
                    }
                } catch (FileNotFoundException e) {
                    System.out.println("File not found: " +
file.getName());
                }
```

```
            });
            threads[i].start();
        }

        // Wait for all threads to finish
        for (Thread thread : threads) {
            try {
                thread.join();
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted.");
            }
        }

        System.out.println("Number of Java Files = " +
javaFileCount);
        System.out.println("Number of Issues = " +
issuesCount.get());
    }
}
```

Output

a. Invalid directory path/no Java files found

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.
(Testing path: src/main/java/Assignment1/Exercise4/src/main/java/org/example)
Enter the directory path to check for Java files: java/Assignment1/Exercise4/src/main/java/org/example
Path detected: java/Assignment1/Exercise4/src/main/java/org/example
Invalid directory path or no Java files found!

Process finished with exit code 0
```

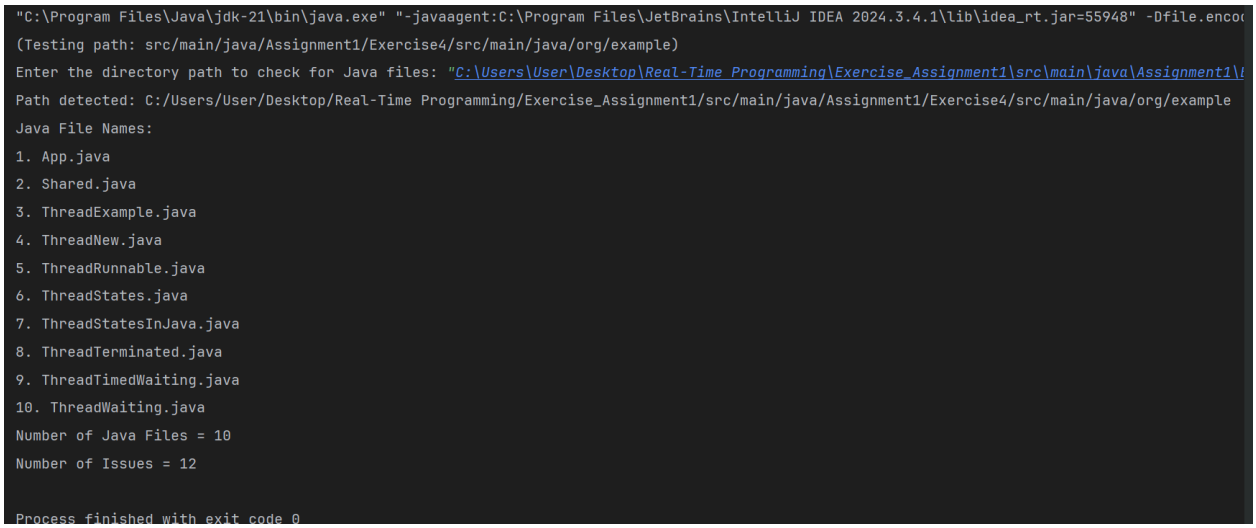
b. Correct testing path

- If using subfolder as testing path that contains “//SOLVED” to calculate number of issues

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.4.1\lib\ide
(Testing path: src/main/java/Assignment1/Exercise4/src/main/java/org/example)
Enter the directory path to check for Java files: src/main/java/Assignment1/Exercise4/src/main/java/org/example
Path detected: src/main/java/Assignment1/Exercise4/src/main/java/org/example
Java File Names:
1. App.java
2. Shared.java
3. ThreadExample.java
4. ThreadNew.java
5. ThreadRunnable.java
6. ThreadStates.java
7. ThreadStatesInJava.java
8. ThreadTerminated.java
9. ThreadTimedWaiting.java
10. ThreadWaiting.java
Number of Java Files = 10
Number of Issues = 12

Process finished with exit code 0
```

- If using folder path in computer as testing path that contains “//SOLVED” to calculate number of issues



- If testing for the file that contains java files but without “// SOLVED” as comment to calculate the number of issues