CHC4010 DevOps - Coursework Assignment (Resit)

2023-2024

Learning Outcomes

Students will be able to:

- analyze Java source code using an IDE
- write JavaDoc description clarifying the input, output, and the functionality of specified methods
- identify and fix bugs in the program utilizing debugger and code analysis tools
- design and implement automated tests for specified methods using JUnit5
- use Git to track progress and manage changes

Introduction

For this coursework you will be doing work understanding, debugging, documenting, and testing a program that has been provided to you.

You have been provided with the java source code that implements a game of Tower of Hanoi. (The game is described in more detail at the end of this document.) Unfortunately, a rather strange and oddly specific glitch has carried the names of all the variables and functions in the program away – literally.

Use a Java IDE to analyse the program with the debugger and code analysis tools, and work out what each variable, parameter, and method does. You should also find and fix the bugs in the program and write automated tests for relevant methods with JUnit5. **All your work should be tracked with a version control tool.**

You must submit your work **before August 22nd**.

Part of the coursework include a quiz related to the program, and will be taken in class on August 23rd. Note that you should complete tracing and learning about the program before submitting it at the end of August 22nd, and doing the quiz on August 23rd.

Feedback will be provided online once the work has been completed. You may also ask for feedback from your class tutor in any module session, although tutors will not directly tell you the answers to questions.

This is individual work and is subject to the University's regulations on assessment. Students who plagiarize work from other students, the Internet, or any other source, will be subject to the Academic Conduct Process. Work submitted late without an appropriate Mitigating

Circumstances claim or Equality Memo will not be marked. Posting parts of the source code, or of your answer, on the Internet (to solicit answers or for any other reason) is not permitted and will be treated as an attempt to cheat or facilitate others cheating.

Tasks

- Change the name of the variables, parameters and methods to more meaningful names
- Add JavaDoc description to each *public* method, describing the input, output of the functions and what they do
- Find and fix the bugs
- Write automated tests for methods `frock()`, `muffs()`, and `norfolk()` with the testing library JUnit 5.
- Use Git to track your progress and make different commits for the four tasks mentioned above with clear commit messages. Your local Git repository should be synced with a remote repository on gitee.com.
- In the quiz, you will answer questions about variable and method naming, documentation, bug fixing, and testing. A printout of the original java source code will be supplied.

  Once you have finished refactoring the program, you must submit ONLY the source code files and the test file on the student website before the deadline.

Marking Scheme

The coursework accounts for 50% of your total score for this module, divided into 3 parts:

- Submitted code: 10%
  - Variable, method, and parameter naming - 3%
  - Javadoc - 2%
  - Bug fixing - 2%
  - Testing - 3%
- Git repository: 10%
  - Commit messages - 3%
  - Isolation of issues - 3%
  - Clean Git history - 2%
  - Organized and methodical work - 2%
- Quiz: 30%
  - Variable, method, and parameter naming - 16%
  - Testing - 10%
  - Bug fixing - 4%

More details are included in the rubrics.

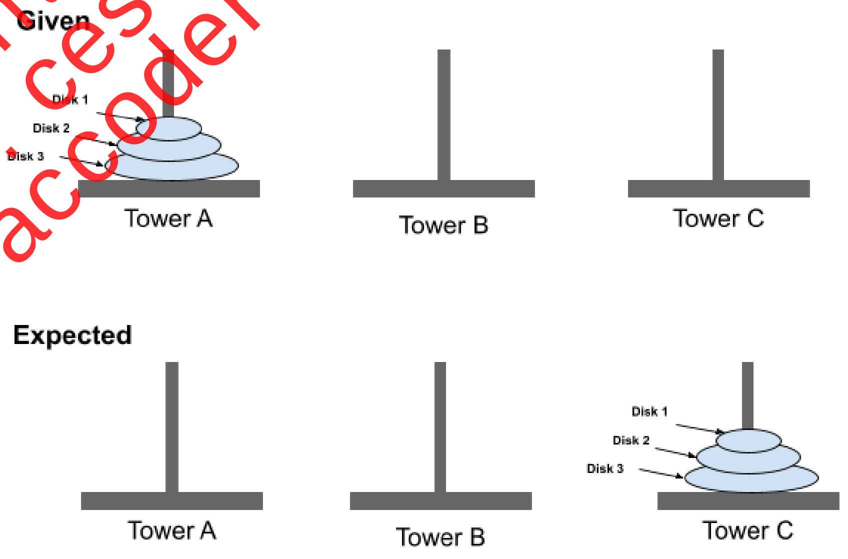
Tips

- You may use either Netbeans or IntelliJ for this exercise, whichever you are most comfortable with.
- The files are provided as bare Java source files. **You should create a *Project* in an IDE and then load the files into it.** Do not just open the files raw, as they will not be associated with a JDK and analysis tools will not be available.
- You should rename the variables in the source code first as it will make the work much easier. Start by identifying the obvious variables based on the user prompts and the string outputs, rename these, and then see how they are used elsewhere in the program. Do make notes of what you have renamed each variable as the quiz will ask you about this.
- The program may use some Java constructions you have not seen before. Tracing the code should help you learn what these do.

Tower of Hanoi

The Tower of Hanoi (also called The problem of Benares Temple or Tower of Brahma or Lucas' Tower and sometimes pluralized as Towers, or simply pyramid puzzle) is a mathematical game or puzzle consisting of three rods and a number of disks of various sizes, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a tower shape. The objective of the puzzle is to move the entire stack of disks to one of the other towers, obeying the following rules:

1. Only one disk may be moved at a time.
2. Each move consists of taking the upper disk from one of the towers (popping) and placing it on top of another tower (pushing).
3. No disk may be placed on top of a disk that is smaller than it.



You can play this game here.

Our program represents the disks on the towers with blocks of black squares, with the length of the block indicating the size of the disk. User selects disk they want to move and the tower they want to move it to by entering two numbers separated by a white space. The first number represents the tower whose upper disk they want to move and the second number represents the tower that the disk will be moved to. For example, if the user entered “2 1”, the upper disk on tower 2 is moved onto the top of tower 1, or at least such an attempt is made.

```
---██--- -------- --------
--████-- -------- --------
-██████- -------- --------
████████ -------- --------
   1        2        3
Enter two numbers x and y to move a disk from tower x to tower y (e.g. 1 2)
1 2
==================================================
-------- -------- --------
--████-- -------- --------
-██████- -------- --------
████████ ---██--- --------
   1        2        3
Enter two numbers x and y to move a disk from tower x to tower y (e.g. 1 2)
1 3
==================================================
-------- -------- --------
-------- -------- --------
-██████- -------- --------
████████ ---██--- --████--
   1        2        3
Enter two numbers x and y to move a disk from tower x to tower y (e.g. 1 2)
```