Assignment 4

Stacks

Introduction

This assignment requires you to implement a generic `Stack` class in Part 1, and then use this stack to solve some problems in Part 2.

Although there is a tester provided for this assignment, it does not include a comprehensive set of sets for each method. You should add your own tests to test cases not considered.

Note: The automated grading of your assignment will include different and additional tests to those found in the `A4Tester.java` file. You are expected to write additional tests until you are convinced each method has full test coverage. The `displayResults` and test coverage videos provide more information about code testing.

Objectives

Upon finishing this assignment, you should be able to:

- Implement a reference-based (node) implementation of a stack;
- Refactor a Java class to that it supports generics;
- Solve problems using only the methods available in the Stack ADT (interface).

Submission and Grading

Submit `A4Exercises.java` and `A4Stack.java` to the BrightSpace assignment page. Remember to click **submit** afterward. You should receive a notification that your assignment was successfully submitted.

If you chose not to complete some of the methods required, you **must** provide a stub for the incomplete method(s) in order for our tester to compile. If you submit files that do not compile with our tester, you will receive a zero grade for the assignment. It is your responsibility to ensure you follow the specification and submit the correct files. Additionally, your code must not be written to specifically pass the test cases in the tester, instead, it must work on all valid inputs. We may change the input values during grading and we will inspect your code for hard-coded solutions. This video explains stubs.

Be sure you submit your assignment, not just save a draft. ALL late and incorrect submissions will be given a ZERO grade. A reminder that it is OK to talk about your assignment with your classmates, but not to share code electronically or visually (on a display screen or paper). We will be using plagiarism detection software.

Instructions

Part 1:

1. Download all of the `.java` files found in the **Assignments -> Assignment 4** page on BrightSpace.
2. Read through the documentation provided in the `Stack.java` interface. There is a lot of information there that will help you set up your generic types when implementing the `Stack` interface.
3. Compile and run `A4Tester.java`. Work through implementing each stack method one at a time. Debug the method until all of the tests pass for that method before proceeding to the next method.

Part 2:

1. Read the documentation for the `stackedCorrectly` and `insertPlate` methods found in `A4Exercises.java`.
2. The `stackedCorrectly` takes a stack of plates and determines whether all of the plates in the stack are ordered correctly. A stack of plates is ordered correctly if the plate with the largest diameter is at the bottom, and no plate is stacked on top of a larger plate;
3. The `insertPlate` method inserts a plate into the correct place in the given stack. Remember: assume the elements in the given stack are placed correctly before inserting the new plate.

CRITICAL: Any compile or runtime errors will result in a **zero grade** (if the tester crashes, it will not be able to award you any points for any previous tests that may have passed). Make sure to compile and run your program **before** submitting it!