CSC6301OM Module 5: Assignment 1 – Project 5 – Planning and Executing a Maintenance Task
17 February 2024
Donald Franks II

Table of Contents

Click the links to go to specific sections.
Section titles are links back to Table of Contents.

Purpose of Program

Franks_CSC6301_M5A1P5.java prompts users for a string of integers to be placed into a Stack. Users can opt to have a list auto-generated with the number of how many integers they choose, or manually enter their own list of integers to be sorted from least to greatest. User can then choose if they want to push or pop numbers from the Stack. Error handling is in place for all inputs. The Stack is printed after each user input to display the indexes of the values.

Command Line Interface

These are the commands I used to navigate to the folder that contained my file, compile, run, and generate the Javadocs for my file in the command line:

1) Navigate to the folder containing the .java file: cd <pathway to the folder containing .java file>

a) You can go to the file containing the folder and click on the file explorer search bar to get the file pathway to copy and paste into the command line.
b) If you have Java packages, make sure you are in the correct directory.

2) Compile the .java file:

a) Without packages: javac <file name WITH .java extension>
b) With packages: javac franks_csc6301_m5a1p5\Franks_CSC6301_M5A1P5.java
c) Note that with packages the first part is the file directory name. These need to be lower case so not to conflict with class names. Note the backslash (\).

3) Run the .java file:

a) Without packages: java <file name WITHOUT .java extension>
b) With packages: java franks_csc6301_m5a1p5.Franks_CSC6301_M5A1P5
c) Note the period (.) separating the directory and file name this time.
d) Note: Make sure to run the java command from the directory where the compiled .class file is located.

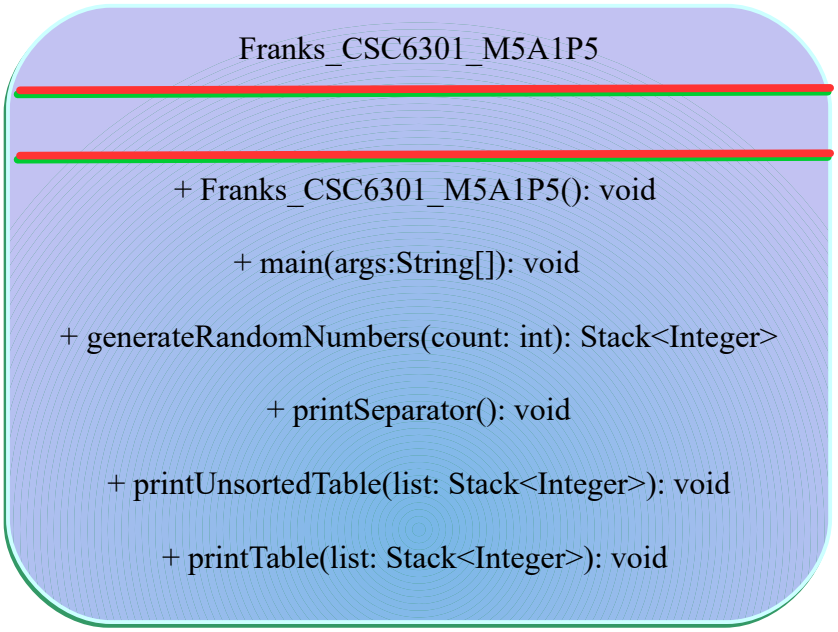
4) Generate the Javadocs:

a) Without packages: javadoc -author -version -d docs <file name WITH .java extension>
b) With packages: javadoc -author -version -d docs franks_csc6301_m5a1p5\Franks_CSC6301_M5A1P5.java
c) Note the backslash (\) again.
d) Note: The -author and -version options are optional. If included, they will add author and version information to the generated Javadocs.

Remember to replace <pathway to the folder containing the .java file> and <file name WITH .java extension> with the actual path and filename.

Class Diagram

Franks_CSC6301_M4A1P4.java has one class with six methods and no attributes:



Changes Made

Franks_CSC6301_M5A1P5.java is a modification of Franks_CSC6301_M4A1P4.java. Franks_CSC6301_M4A1P4.java executes the same output as Franks_CSC6301_M5A1P5.java, but utilizes a Linked List rather than a Stack. Franks_CSC6301_M4A1P4.java is also a single file and has less methods than Franks_CSC6301_M5A1P5.java. An argument could be made to separate the print and separator methods into other class files, but I still felt like with this small, single use program, one class with multiple methods still works. If this program got any bigger or I wanted to use say the print methods in other programs, then I could see reusing the code in separate class files.

The changes from Franks_CSC6301_M4A1P4.java to Franks_CSC6301_M5A1P5.java include:

1) Added package name at the top of the code.
2) Changed the main data structure from a Linked List to a Stack.
3) This also means that the built in functions of add and remove for a Linked List were changed to push and pop for a Stack.
4) In order to demonstrate the Last-In-First-Out (LIFO) principle, a second, unsorted list was instantiated at the beginning of the code to differentiate from the sorted Stack.
5) The printUnsortedTable() method was added to display the unsorted Stack.
6) The printSeparator() method was added as a pseudo decorator to make output readability easier.
7) Output spacing issues from Franks_CSC6301_M4A1P4.java have been fixed.