Michael Evans Program 2 Documentation

The purpose of the program is to allow the user to interact with stacks using multiple methods that are outside of the original stack class. To build the program, you only need the classes “FStack” and “Driver.”

Program interaction is simple; just type the desired menu option into the console, and, for methods with parameters, follow the input up with said parameter when prompted.

Algorithms

* drop()
  + Pop the top item of the stack.
* swap()
  + Top and copy the first element.
  + Pop.
  + Top and copy the second element.
  + Pop.
  + Push the first element copy onto the stack.
  + Push the second element copy onto the stack.
* rot()
  + Top and copy the first element.
  + Pop.
  + Top and copy the second element.
  + Pop.
  + Top and copy the third element.
  + Pop.
  + Push the second element copy.
  + Push the first element copy.
  + Push the third element copy.
* dup()
  + Top and copy the first element.
  + Push the first element copy.
* over()
  + Top and copy first element.
  + Pop.
  + Top and copy second element.
  + Push first element copy.
  + Push second element copy.
* pick(int i)
  + Copy the item at position “i” in the stack.
  + Push the copy.
* roll(int i)
  + Copy the stack.
  + Find the item at index “i.”
  + Push it onto the top of the copy.
  + Change the original stack into a copy of the copy.

Test Cases

I would test this program with empty and full stacks, to make sure that my exception handling works correctly.