Lab 5

Stacks and Queues

1. Questions. (10 points)

1.1 What does LIFO stand for?  
  
Last In Last Out  
  
Last In First Out  
  
First In Last Out  
  
First In First Out

1.2. How many items can you put on a stack?  
  
As many as you allocate when you create the stack  
  
As many as will fit in the computer's memory  
  
As many as will fit in a single string  
  
Ten

1.3. Which operations does a stack support? (Choose two)  
  
Delete Push Append Sort Pop

1.4. Which kind of structure is a stack?  
  
FIFO LIFO FILO LILO

1.5. Suppose variable LIFO is defined as a stack. How do you check to see if the stack is empty? (Choose two)  
  
if (LIFO.empty()) if (LIFO.size() > 0) ***<-Tough one!!***

1.6. What happens if you call peek() on an empty stack?  
(Hint: <http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html> )

peek() returns zero  
  
peek() returns NULL  
  
peek() throws an exception  
  
You can't call peek() on an empty stack

1.7. Please complete the following table.

|  |  |  |
| --- | --- | --- |
| Operation | Stack | Output |
| push(“Mandarin”) | “Mandarin” | - |
| pop() | - | “Mandarin” |
| push(“Navel”) | “Navel” | - |
| push(“Clementine”) | “Clementine”  “Navel” | - |
| pop() | “Navel” | “Clementine” |
| push(“Tangelo”) | “Tangelo”  “Navel” | - |
| push(“Valencia”) | “Valencia”  “Tangelo”  “Navel” | - |
| pop() | “Tangelo”  “Navel” | “Valencia” |

1.8. Please complete the following table.

|  |  |  |
| --- | --- | --- |
| Operation | Stack | Output |
| push(15); | 15 |  |
| push(25); | 25  15 |  |
| push(10); | 10  25  15 |  |

Rubric:

1.1 - 1.6: 0.5 points per question

1.7, 1.8: 3 points per table

2. Stacks in Java. (10 points)

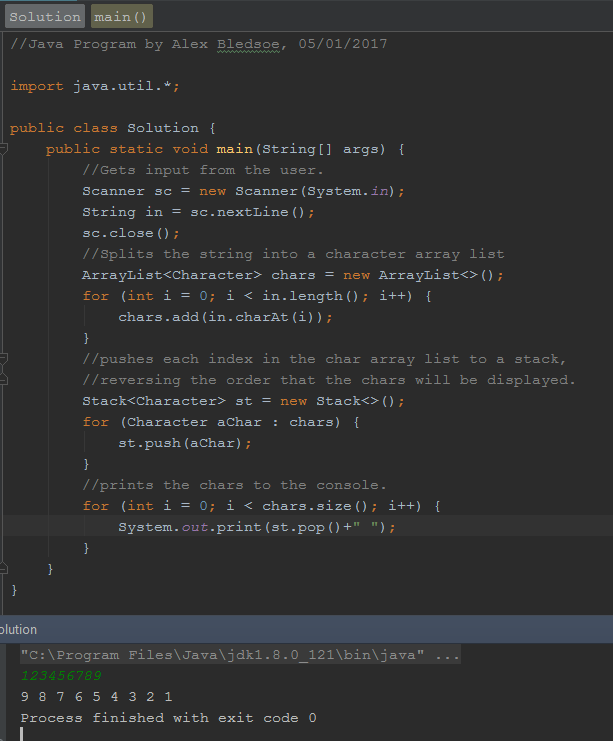
Please review the reference material provided on how to use java.util.Stack.  
  
Create a Java program to do the following using console input/output:

2.1. Input a string from the user  
 2.2. Use a stack to reverse the letters of the string  
 2.3. Output the reversed string  
  
Rubric:  
Student name and today’s date is a comment in the first line of the programs: -5 points if fails  
Screenshot and program code: -5 points if fails   
Input from the user: 2 points  
String -> Character conversion: 2 points  
Stack holds all characters: 2 points  
Reverse string and output: 4 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your main program's .java file, here

**Solution.java:**

//Java Program by Alex Bledsoe, 05/01/2017  
  
import java.util.\*;  
  
public class Solution {  
 public static void main(String[] args) {  
 //Gets input from the user.  
 Scanner sc = new Scanner(System.*in*);  
 String in = sc.nextLine();  
 sc.close();  
 //Splits the string into a character array list  
 ArrayList<Character> chars = new ArrayList<>();  
 for (int i = 0; i < in.length(); i++) {  
 chars.add(in.charAt(i));  
 }  
 //pushes each index in the char array list to a stack,  
 //reversing the order that the chars will be displayed.  
 Stack<Character> st = new Stack<>();  
 for (Character aChar : chars) {  
 st.push(aChar);  
 }  
 //prints the chars to the console.  
 for (int i = 0; i < chars.size(); i++) {  
 System.*out*.print(st.pop()+" ");  
 }  
 }  
}



3. Queues (5 points)

3.1. What does FIFO stand for?

Last In Last Out  
  
Last In First Out  
  
First In Last Out  
  
First In First Out

3.2. How many items can you put on a queue?  
  
As many as you allocate when you create the queue  
  
As many as will fit in the computer's memory  
  
As many as will fit in a single string  
  
Ten

3.3. Which operations does a queue support? (Choose two)  
  
Add Push Remove Sort Pop

3.4. Which kind of structure is a queue?  
  
FIFO LIFO FILO LILO

3.5. Suppose variable FIFO is defined as a queue in Java. How do you check to see if the queue is empty? (Choose two)  
  
if (FIFO.size() > 0) if (FIFO.peek() != null)

3.6. What happens if you call remove() on an empty queue?  
(Hint: <https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html> )

remove() returns zero  
  
remove() returns NULL  
  
remove() throws an exception  
  
You can't call remove() on an empty queue

3.7. Please complete the following table.

|  |  |  |
| --- | --- | --- |
| Operation | Queue | Output |
| Add(15); | 15 | - |
| Add(25); | 15, 25 | - |
| Add(10); | 15,25,10 | - |
| Remove(); | 25, 10 | 15 |
| Remove(); | 10 | 25 |
| Add(7); | 10, 7 |  |
| Remove(); | 7 | 10 |
| Remove(); |  | 7 |

3.8. Please complete the following table.

|  |  |  |
| --- | --- | --- |
| Operation | Queue | Output |
| add(“Putty”) | “Putty” | - |
| remove() | - | “Putty” |
| add(“Beige”) | “Beige” | - |
| remove() | - | “Beige” |
| add(“Bisque”) | “Bisque” | - |
| add(“Buff”) | “Bisque”  “Buff” | - |
| remove() | “Buff” | “Bisque” |

Rubric:

3.1 - 3.6: 0.5 points per question

3.7, 3.8: 1 points per table

4. Queues in Java (10 points) Thanks to Reges Building Java Programs 4th edition.

Write a method called doubleQueue that accepts a queue and replaces every element of the queue with two copies of that element.

Please program three test cases into your program:

[ 1, 2, 3 ] becomes [1,1,2,2,3,3,]

[ “a”, “b”, “c” ] becomes [ “a”, “a”, “b”, “b”, “c”, “c” ]

[ 5, 6, 7, 8, 9] becomes [5, 5, 6, 6, 7, 7, 8, 8, 9, 9]

However, your doubleQueue routine must work for *any* queue, not just the test cases.

Rubric:

Student name and today’s date are a comment on the first line of the program: (-5 if fails)

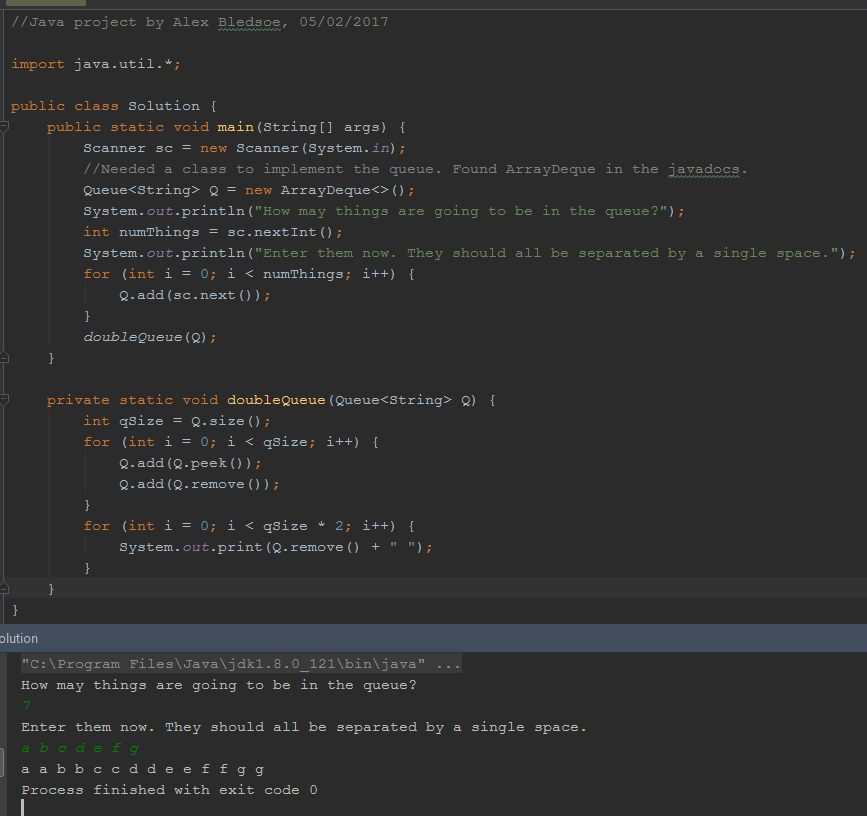
Screenshot and program code: (-5 points if fails)

doubleQueue algorithm: 5 points  
Main program with test cases: 5 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your main program's .java file, here.

**Solution.java:**

//Java project by Alex Bledsoe, 05/02/2017  
  
import java.util.\*;  
  
public class Solution {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 //Needed a class to implement the queue. Found ArrayDeque in the javadocs.  
 Queue<String> Q = new ArrayDeque<>();  
 System.*out*.println("How may things are going to be in the queue?");  
 int numThings = sc.nextInt();  
 System.*out*.println("Enter them now. They should all be separated by a single space.");  
 for (int i = 0; i < numThings; i++) {  
 Q.add(sc.next());  
 }  
 *doubleQueue*(Q);  
 }  
  
 private static void doubleQueue(Queue<String> Q) {  
 int qSize = Q.size();  
 for (int i = 0; i < qSize; i++) {  
 Q.add(Q.peek());  
 Q.add(Q.remove());  
 }  
 for (int i = 0; i < qSize \* 2; i++) {  
 System.*out*.print(Q.remove() + " ");  
 }  
 }  
}



5. Comprehensive lab. (10 points)

Create a Java program to do the following using console input/output:   
Thanks to Augenstein's Data Structures Using Java

* 1. Read in a string from the user
  2. Output "Valid" if the string contains a valid parenthesis pattern.
  3. Output "Invalid" if the string contains an invalid parenthesis pattern.

In addition to use input, please code the test cases into the main program so you don’t have to input them manually each time. The program must still work on arbitrary input strings, not just the test cases.

Examples:

|  |  |
| --- | --- |
| Input | Output |
| (We all (love LISP)) | Valid |
| Life is easy | Valid |
| Life is (fun) | Valid |
| (We never)(forget) | Valid |
| Call\_Me() | Valid |
| (1 (2 (3 | Invalid |
| (We all (hate APL) | Invalid |
| This is terrible. ((()(()())(())() | Invalid |
| )( | Invalid |

Rubric:

Student name and today’s date are a comment on the first line of the program: (-5 if fails)  
Screenshot and program code: (-5 points if fails)

Test cases included in program: (-5 points if fails)

Works for user input, not just test cases: (-5 points if fails)

String to character breakdown: 2 points  
Parentheses check using a stack or queue: 8 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your main program's .java file, here.

**Solution.java:**

//Java program by Alex Bledsoe, 5/2/2017.  
  
import java.util.\*;  
  
public class Solution {  
 public static void main(String[] args) {  
 //Create scanner object and get string to test from user.  
 Scanner sc = new Scanner(System.*in*);  
 String in = sc.nextLine();  
 //Passes the string into isValid method.  
 if (*isValid*(in)) {  
 System.*out*.println("Valid");  
 } else {  
 System.*out*.println("Invalid");  
 }  
  
 }  
  
 //Method checks for valid parenthesis pattern in a string.  
 private static boolean isValid(String in) {  
 //Turns input string into a character array.  
 char[] chars = new char[in.length()];  
 for (int i = 0; i < chars.length; i++) {  
 chars[i] = in.charAt(i);  
 }  
 /\*New Stack to hold '(' characters.  
 \*Loop checks for ')' characters when the Stack is empty (invalid pattern, returns false).\*/  
 Stack<Character> st = new Stack<>();  
 for (char aChar : chars) {  
 if (aChar == ')' && st.isEmpty()) {  
 return false;  
 } else if (aChar == '(') {  
 st.push(aChar);  
 } else if (aChar == ')') {  
 st.pop();  
 }  
 }  
 /\*If uneven # of each parenthesis (Stack isn't empty after loop),  
 \*then there is an invalid pattern, returns false.  
 \*Else it will return true.\*/  
 return st.isEmpty();  
 }  
}

