Quiz Submissions - Linked Lists, Stacks, and Queues Homework Quiz

Chork Hieng (username: gt9182iu)

Attempt 1

Written: Feb 23, 2022 9:13 PM - Feb 23, 2022 9:20 PM

Submission View

Your quiz has been submitted successfully.

Question 1 **2 / 2 points**

What will be the output of this function:

```
public static void demo1(Scanner sc) {
    Queue<String> q = new ArrayQueue<>(10);
    while (sc.hasNext()) {
        String s = sc.next();
        q.enqueue(s);
    }
    while (!q.isEmpty()) {
        String s = q.dequeue();
        System.out.printf("%s ", s);
    }
}
```

If the input is:

```
one two three four
```

- ○ `4 3 2 1`
- ○ `1 2 3 4`
- ✓ ○ `one two three four`
- ○ `four three two one`

Question 2 — 2 / 2 points

What will be the output of this function:

```
public static void demo2(Scanner sc) {
    Stack<String> stk = new LinkedStack<>();
    while (sc.hasNext()) {
        String s = sc.next();
        stk.push(s);
    }
    while (!stk.isEmpty()) {
        String s = stk.pop();
        System.out.printf("%s ", s);
    }
}
```

If the input is:

```
one two three four
```

- ◯ `one two three four`
- ✔◯ `four three two one`
- ◯ `1 2 3 4`
- ◯ `4 3 2 1`

Question 3 — 2 / 2 points

In this drawing, the arrows (`-->`) represent the links between **`Node`**s in a **`LinkedList<String>`**, and the strings represent the data in the **`Node`** objects. If the original list contents are:

```
Curly --> Larry --> Moe
```

What is the state of the list after **`addFront("Shep")`**?

- ✔◯ `Shep --> Curly --> Larry --> Moe`

- `Shep --> Larry --> Moe`
- `Curly --> Larry --> Moe --> Shep`
- `Curly --> Larry --> Shep`

Question 4 **2 / 2 points**

In this drawing, the arrows (`-->`) represent the links between `Node`s in a `LinkedList<String>`, and the strings represent the data in the `Node` objects. If the original list contents are:

```
Curly --> Larry --> Moe
```

What is the state of the list after `addBack("Shep")`?

- `Curly --> Larry --> Shep`
- `Shep --> Larry --> Moe`
- ✓ `Curly --> Larry --> Moe --> Shep`
- `Shep --> Curly --> Larry --> Moe`

Question 5 **2 / 2 points**

In this drawing, the arrows (`-->`) represent the links between `Node`s in a `LinkedList<String>`, and the strings represent the data in the `Node` objects. If the original list contents are:

```
Curly --> Larry --> Moe
```

What is the state of the list after removeFront `()`?

- Shep --> Larry --> Moe
- Curly --> Larry --> Shep
- Curly --> Larry
- ✓ Larry --> Moe

Question 6 **2 / 2 points**

Which of these functions will properly swap the first two elements of a stack? (Assume the stack has at least two elements.)

- ✓
```
public static <T> void swapFirstTwo(Stack<T> stk) {
    T t1 = stk.pop();
    T t2 = stk.pop();
    stk.push(t1);
    stk.push(t2);
}
```

-
```
public static <T> void swapFirstTwo(Stack<T> stk) {
    T t1 = stk.pop();
    T t2 = stk.pop();
    stk.push(t2);
    stk.push(t1);
}
```

-
```
public static <T> void swapFirstTwo(Stack<T> stk) {
    T t1 = stk.pop();
    stk.push(t1);
    T t2 = stk.pop();
    stk.push(t2);
}
```

-
```
public static <T> void swapFirstTwo(Stack<T> stk) {
    T t1 = stk.pop();
    stk.push(t2);
    T t2 = stk.pop();
    stk.push(t1);
}
```

Question 7 **2 / 2 points**

Which of these functions will move the front-most item of a queue to the back of the queue? Assume the queue has at least one element, and that it is implemented using the API described in the text.

```
public static <T> void rotateToBack(Queue<T> q) {
    T t = q.pop();
    q.push(t);
}
```

```
public static <T> void rotateToBack(Queue<T> q) {
    T t = q.enqueue();
    q.dequeue(t);
}
```

✔
```
public static <T> void rotateToBack(Queue<T> q) {
 T t = q.dequeue();
 q.enqueue(t);
}
```

Attempt Score: 14 / 14 - 100 %

Overall Grade (highest attempt)**:** 14 / 14 - 100 %

Done