CSC933 Tutorial 3

Overview

This tutorial aims to teach you how we use stacks and queues implicitly to organise information and solve everyday problems.

The Problems

Warm up Question

Come up with as many real world examples of stacks and queues as you can. For example eating Pringles (a stack: LIFO) and a queue of people at a checkout (a queue: FIFO).

Stacks

In relation to stacks, answer the following questions:

1. What does push mean?
2. What does pop mean?
3. What does peek mean?

Draw a simple diagram to illustrate this.

Queues

In relation to queues, answer the following questions:

1. What does dequeue mean?
2. What does enqueue mean?

Again, draw a simple diagram to illustrate this.

Examples

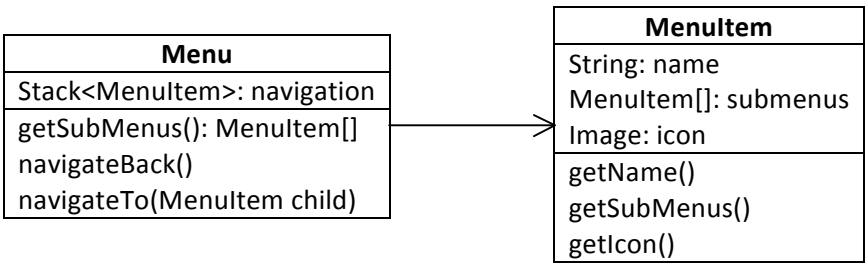
1. Push(6), pop, push(7), push(1), push(8), pop, pop, push(2), pop, pop, push(4), push(6), pop.
    a. Show the numbers removed from the stack in the order they are removed.
    b. Now perform the same task, but assume that instead of push and pop we use dequeue and enqueue on a queue.
2. Someone gives you a set of directions to navigate to their house. Would you ask for directions to get home? Why?

(Last question on page 2…)

The Bigger Problem

Find and inspect any settings menu based system (for example on a house phone, mobile phone, PC, camera, etc) where you have to go down into different settings options. What do you notice about the navigation?

Let's look at how we could code our own menu system. Take a look at the following two classes:



Write the method outline and content for the methods `getSubMenus`, `navigateBack` and `navigateTo` in the `Menu` class using Java code which will compile.

Hints

To get a `MenuItem` from the top of the stack we would simply say:

```
MenuItem topMenuItem = navigation.pop();
```

To add a `MenuItem` (called newMenuItem) to the top of the stack we would simply say:

```
navigation.push(newMenuItem)
```

Example Usage

For those of you who would find it useful to see example code, here is some rough skeleton code to show you the overview of how the menu class could get used.

```
Menu settingsMenu = new Menu();
while (true)
{
      display(settingsMenu.getSubMenus());
      if (pressedBack())
            settingsMenu.navigateBack();
      else
            settingsMenu.navigateTo(getUserSelection());
}
```