Assignment 3 - Queues

Comp310 Object oriented data structures in Java

# TopicS

Queue applications and implementations

# Readings

Carrano: Chapter’s 10 & 11

# Objectives

Implement a queue algorithm, and apply your implementation to a First-In first-Out (FIFO) project.

# Instructions

1. **(Optional) Select a Partner.** Please let instructor know **BEFORE YOU START** who you will work with.
2. Read over the assignment and ASK QUESTIONS about anything that you don’t understand (before you start).
3. Do the [Problems](#_problems) listed in the next section. .
4. Be sure to follow [Good Programming Practices](#_Good_programming_practices), also listed below.
5. Individually answer the [Analysis / Summary](#_Analysis_/_Summary) questions as a Journal entry on blackboard.
6. Follow instructions for [submitting your work](#_Submitting_your_work:).
7. **Demo your project to instructor** (required for grading).

# problems

1. **Implementation:** Write your own **circular array** implementation of the **Queue** Abstract Data Type (ADT) described in chapter 11 on page 288.
   1. **Test all methods** before you proceed to use your class in the application *(below)*
2. **Application**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ch | Project | | Page | Notes |
| 10 | **Select one** | **#7** | 271 | Use the queue that you implemented in #1. |
| **#8** | 271 |

# Good programming practices

1. **Design your solution using UML diagrams before you begin coding.**

**Please ask if you need help with this**

* 1. Provides understanding of the big picture and how classes relate to one another.
  2. Some IDE’s will then create the framework for your classes right from the diagrams.

1. **For all classes**

* Make your instance variables private
* Include constructors to initialize your instance variables.
* Derived class constructors should leave initialization of super class instance variables to the super classes’ constructors:
  + Remember the call to the super classes constructor is: super( <init1>, <init2>,..)
* Include Accessor and Mutator methods for all instance variables *(please ask if you’re not sure what these are)*

“use” = Before its’ methods are called by a driver or by another class.

* Include a **main** method for testing (unless it’s an abstract class) and test all methods before you “use” this class.
* Add comments to your code, not just so it’s easier for other readers, but also so it’s easier for you to remember your logic.

# Analysis / Summary

**In your Journal on Blackboard, please answer the following questions:**

1. Analysis
   1. Was this the best application of a queue? Why or why not?
   2. Describe other types of problems for which you think it would be appropriate to apply a queue.
2. Summary
   1. If you worked in pairs
      1. How did you “divide up” the work so that each student still met the objectives for the assignment (i.e., learned, understood and applied the concepts).
      2. What was your contribution?
      3. How did you coordinate code changes/testing?
      4. Other observations about working with a partner?
   2. Where did you have trouble with this assignment? How did you move forward? What topics still confuse you?
   3. What did you learn from this assignment?  *(Please be specific)*

# Submitting your work

1. Make sure that your name(s) are in all of your project files
2. If you have more than one file for your solution, make a compressed file for your project
3. In Blackboard, attach your solution file to the submission for this assignment.