P10 Order Up! v2.0

Overview

As with P08's Order Up!, you'll be creating a queue of Orders (in fact, almost exactly the same Order class as before). This time, though, we're going to impose a **priority** on them - anything that takes *longer* to prep gets to move to the front of the queue and start first.

To implement this priority queue, you'll be using an array-based heap. Everything else is pretty straightforward!

Grading Rubric

5 points	**Pre-assignment Quiz**: accessible through Canvas until 11:59PM on **04/26**.
15 points	**Immediate Automated Tests**: accessible by submission to Gradescope. You will receive feedback from these tests *before* the submission deadline and may make changes to your code in order to pass these tests. Passing all immediate automated tests does **not** guarantee full credit for the assignment.
30 points	**Additional Automated Tests**: these will also run on submission to Gradescope, but you will not receive feedback from these tests until after the submission deadline.

Learning Objectives

The goals of this assignment are:

- Gain experience with test-driven development
- Familiarize yourself with the implementation of an array-based priority queue
- Review the Comparable interface
- Finish the semester on a high note?

Additional Assignment Requirements and Notes

As you work on your code and before you submit your assignment, verify that:

- ▢ You have NOT added any instance or class constants, or instance or class variables, or public methods (including constructors) beyond what is defined or provided in this writeup.
    - **NOTE**: You are allowed to add *private* helper methods to any class, and any additional **public static boolean** test methods you like *to your tester class only*.
- ▢ You have defined any *local* variables you need to implement the methods in this specification.
- ▢ All methods, public or private, must have their own Javadoc-style method header comments in accordance with the CS 300 Course Style Guide. We've provided most of them, but if you add any helper methods or additional tester methods, you must comment those too.
- ▢ You have NOT imported anything into your files EXCEPT java.util.**Arrays** and java.util.**NoSuchElementException**.
- ▢ You have adhered to the Academic Conduct Expectations and Advice. If you're feeling the strain of this incredibly difficult Spring 2021 semester and panicking, *please* don't hesitate to contact your instructor. We understand, we're feeling it too. We want you to succeed in this course, but not at the expense of your mental or physical health, or your academic integrity. We can always help you work something out.

CS 300 Assignment Requirements

You are responsible for following the requirements listed on both of these pages on all CS 300 assignments, whether you've read them recently or not. Take a moment to review them if it's been a while:

- Academic Conduct Expectations and Advice, which addresses such questions as:
    - How much can you talk to your classmates?
    - How much can you look up on the internet?
    - What do I do about hardware problems?
    - and more!
- Course Style Guide, which addresses such questions as:
    - What should my source code look like?
    - How much should I comment?
    - and more!

Getting Started

1. Create a new project in Eclipse, called something like **P10 OrderUp2**.
    a. Ensure this project uses Java 11. Select "JavaSE-11" under "Use an execution environment JRE" in the New Java Project dialog box.
    b. Do **not** create a project-specific package; use the default package.

2. Add the following four (4) Java source files within that project's src folder:
    a. PriorityQueueADT.java (interface, will not be modified)
    b. Order.java (redownload! **small modifications needed**)
    c. OrderPriorityQueue.java (array-based max-heap, code skeleton only)
    d. OrderPriorityQueueTester.java (code skeleton only, main method)

First, make your Order class Comparable – you'll need to add the Comparable<Order> interface and add the associated method.

Next, continue coding with the OrderPriorityQueueTester class. Everything is laid out for you there, and it will direct you step-by-step on what to do next.

In a move that will infuriate many of you, the immediate tests for the autograder this week will be almost entirely focused on running your tester class against various implementations - pretend we're running your tests on your classmates' code, and you're trying to help them figure out where they've gone wrong. We'll try to help you determine whether your tests are sufficiently robust to help you test your OWN code, too, so that if you're passing all of our tests AND all your own tests, you should get nearly a perfect score.

Assignment Submission

Once you're satisfied with your work, both in terms of adherence to this specification and the academic conduct and style guide requirements, submit ALL of your source code through Gradescope.

Your score for this assignment will be based on the submission marked "**active**" prior to the deadline. You may select which submission to mark active at any time, but by default this will be your most recent submission.

Copyright Notice