

Priority Queues

1. On pp. 397-399, the author gives a description of the Priority Queue ADT. Create a new project called "Priority Queue", and in it create an interface called **PriorityQueue** that is based on the ADT. Do **not** make it parameterized. Instead, assume that it will contain elements of type **Object**. In parts 2 and 3 below, you will have to do some writing. It might be better for you to put those off until after you finish the rest of the lab, since 2 and 3 do not involve writing any code.
2. On page 399 the author discusses the idea of possibly implementing a priority queue by "composition" (using a class that has a queue in it). Although we will not do this, it is helpful to answer the question, what's so bad about it? What would you have to do to make a priority queue work using the Adapter Pattern with a **ListQueue**? What would be bad about that? Write your answers for this on a separate sheet of paper to be turned in tomorrow.
3. Another possible way to implement a priority queue is by using inheritance, as the author outlines in #4 on page 399. Would this be better or worse than #2? Why? Give a detailed answer. Provide your answer on the same page(s) as what you did for #2.
4. If each priority number is a nonnegative integer, and there are not a lot of priorities, a good way to implement a priority queue is to use an array of queues. (An **ArrayList** of queues would also work) As the author indicates at the bottom of p. 397, lower numbers represent higher priorities. So you would have a queue for all items with priority 1, another queue for all items with priority 2, and so on. Assume that there are no more than 10 integer priorities, numbered from 1 to 10. Create a class called **PQ** that will implement the **PriorityQueue** interface you created in step 1 as an array (or **ArrayList**) of **ListQueue** objects. Create a **Job** class that contains a **String** called **name** and an integer called **priority**. Provide a complete set of accessor methods for this class. Then in the **PQ** class implement stubs for each required method (just enough to allow the class to compile). Then create a JUnit test class that tests each method. Provide a good, complete set of test methods. Then implement the methods so that the tests pass. Copy all of your java files to the shared drive in a new folder called "Priority Queue" by Monday of next week.