

Assignment 2

The goal of this assignment is to improve on your assignment one and implement Iterator, Strategy and Command pattern.

Due Sept 25

1. Make sure that you have a copy of your unit tests from assignment 1. When you are done with this assignment determine how good the tests were. That is after making the changes required in this assignment how confident are you that your code works after running the tests. Did you have to write new tests as you refactored your code?
2. Refactor your priority queue code to use standard names for methods, remove helper methods on the priority queue that deal with nodes, and any other clean up you feel is needed in your code. You might find the refactorings rename and move useful here.
3. A priority queue is a collection. Determine the correct location in your language's collection class hierarchy. Find **all** methods that you need to implement in-order to add your class in the language's collection class hierarchy.
4. Make the parent class of your priority queue class the parent determined in problem 3. Rename your existing methods to conform to the collection classes standards. You may need to stub some methods to satisfy the parent class's constraints. Note we will only be interested in implementing a few of these methods. You do not have to implement all the methods in the parent class. In addition to the methods from assignment 1 add toArray and the toString methods. These are Java names use the appropriate names for your language.
5. Use the strategy pattern to allow you determine how the priority queue will be ordered when a priority queue object is created.
6. Implement an iterator for your strategy pattern class. Don't convert your priority queue to an array or other collection to implement your iterator. You can implement either an internal or external iterator.
7. Support undo using the Command pattern on your priority queue. That is you should be able to create a priority queue add n elements, remove k elements, add L elements then undo those operations on at a time.
8. Write unit tests for the code you write for this assignment.

Grading

Item	Points
Working Code	10
Unit Tests	10
Proper implementation of Patterns	10 per Pattern (30 points total)
Quality of Code	10
Proper Parent Class & Method names	10
Written Answers to question 1	5

Turning in your Assignment

The assignment is given as a series of steps. Turn in the code as it is when you have finished all steps. Do not turn in multiple copies of the same class.