

CS-2851 Lab 2: DoublyLinkedListQueue Class

Objective

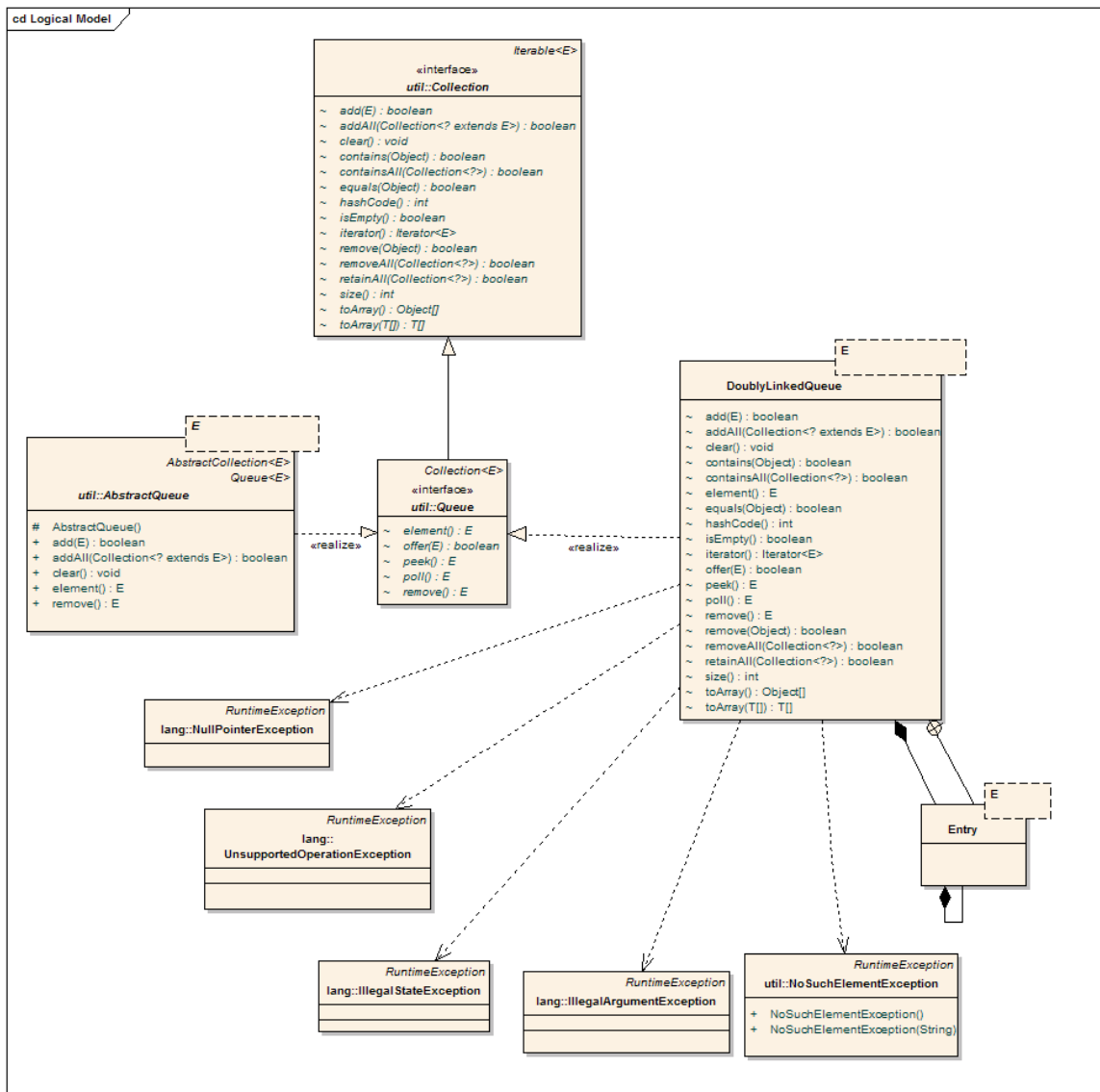
- Gain experience designing a collection, understand linked list data structure.

Overview

This is a 2-week lab (not including Spring break), but you must have the final design completed and submitted (along with a preliminary implementation) before the second lab, i.e., week 5.

Except for a very basic **AbstractQueue<E>** class that provides a minimal implementation of the **element()** and **remove()** methods, the Java Collections Framework only provides a **Queue<E>** interface, i.e., no classes that provide an implementation.

In this lab, you will create an implementation of the **Queue<E>** interface in a generic class called **DoublyLinkedListQueue<E>**. The UML class diagram for the related classes is shown below:



Assignment

Create a class called **DoublyLinkedList<E>** that implements the **Queue<E>** interface. Your class may not make use of any collection classes in the JCF such as **ArrayList<E>** or **LinkedList<E>**, i.e., you must implement the internal data structure yourself. Internally, your class must store elements of the collection in a doubly-linked list that you design and implement. Recall from lecture and reading assignments that the elements within the doubly-linked list are objects of the nested class **Entry<E>**. Re-read Chapter 7 in the textbook, and study the **SingleLinkedList.java** sample code.

You are more than welcome to implement additional methods for extra credit.

Finally write a basic application of your choosing that utilizes your **DoublyLinkedList** class.

Final Lab report due prior to lab week 5

You must submit:

- your completed **DoublyLinkedListQueue<E>** class
- Sample test application that calls each method at least once.
- Your final report, including:
 - A completed Enterprise Architect-generated UML class diagram containing all attributes you will implement in your **DoublyLinkedListQueue<E>** class, as well as the complete nested **Entry<E>** class (with all methods and attributes). Use EA template file, JCFTemplate.EAP, which contains the reverse-engineered JCF classes taken from the java.util package of the JDK.
 - the **DoubleLinkedList.java** file containing the attributes and method declarations that you'll implement - *including javadoc-style documentation for each method*. You needn't have the real implementation within the methods - just the declaration (which essentially documents your detailed design) and a call to throw an UnsupportedOperationException. Note that this implies that your class should be able to be incorporated into another project and compile without errors!
 - Your reaction (a couple of paragraphs on what you learned and what you thought about while working on this assignment, including at least one thing that you discovered as a result of executing your test plan)
 - An analysis of the asymptotic time complexity for each method in the **DoublyLinkedListQueue<E>** class.
 - Results from a test program that calls each method at least once.

Email your lab a zip archive file containing your UML class diagram, DoubleLinkedList.java file, and your report (PDF) to: urbain@msode.edu . Make sure you include cs2851, lab2, and your name in the archive filename and in the subject line of your email.

Indicate how much time you spend on this assignment (during week 6) in the [FAST system](#).

Thanks, Jay Urbain

This assignment was originally developed by [Dr. Chris Taylor](#).