

CS401 Lab 6: Queue Data Structure Implementation

Overview

- Focus: Understanding and implementing Queue data structures using Arrays.
- Objective: Implement two types of Queue designs and perform operations on Employee objects.

Requirements

1. Queue Implementations

Implement two types of Queue data structures using Arrays:

1. Fixed Front Queue
2. Floating Front Queue

2. Queue Operations

For each Queue implementation, include the following methods:

1. *public boolean enqueue(Employee e)*
 - Adds an Employee object to the rear of the queue
 - Returns true if successful, false if the queue is full
2. *public Employee dequeue()*
 - Removes and returns the Employee object from the front of the queue
 - Returns null if the queue is empty

3. Employee Class

Each Employee object should have at least:

- Name (String)
- ID (int or String)

4. Input Processing

- Read Employee data from "emp.txt" (same format as Lab 4)
- Each line format: "Name ID" (Name and ID separated by a space)

5. Main Program

For each Queue implementation (Fixed Front and Floating Front):

1. Read the first 5 Employee objects from "emp.txt" and enqueue them
2. Print all elements currently stored in the queue
3. Dequeue twice
4. Print all elements currently stored in the queue again

6. Code Documentation

- Include inline comments for all methods and complex code sections
- Provide a README file with:
 - Description of the program
 - Instructions on how to compile and run the program
 - Explanation of both Queue implementations

- Command to run the JAR file

Input

- A single text file named "emp.txt"
- Format: One employee per line, "Name ID"

Output

Your program should demonstrate the following for both Fixed Front and Floating Front Queues:

Fixed Front Queue:

Initial Queue Contents:

1. [Employee Name: John Doe, ID: 1001]
2. [Employee Name: Jane Smith, ID: 1002]
3. [Employee Name: Bob Johnson, ID: 1003]
4. [Employee Name: Alice Brown, ID: 1004]
5. [Employee Name: Charlie Davis, ID: 1005]

After dequeuing twice:

1. [Employee Name: Bob Johnson, ID: 1003]
2. [Employee Name: Alice Brown, ID: 1004]
3. [Employee Name: Charlie Davis, ID: 1005]

Floating Front Queue:

[Similar output structure as Fixed Front Queue]

Submission Requirements

1. Source Code:
 - FixedFrontQueue.java
 - FloatingFrontQueue.java
 - Employee.java
 - Main.java (or appropriate name for your main class)
2. Compiled Bytecode:
 - All corresponding .class files
3. Output:
 - PDF file containing program output for both queue implementations
4. Executable JAR file
5. README file
6. emp.txt (the input file)

Important Notes

- Ensure proper error handling for queue overflow and underflow situations
- Clearly demonstrate the difference between Fixed Front and Floating Front implementations in your code
- Test your program thoroughly with various scenarios (e.g., enqueueing when full, dequeueing when empty)