

CIS2168 F18 005-006 Assignment 3

Linked Stack and Queue

1. Objectives

This assignment will

- Enhance your understanding of linked list data structures
- Enhance your understanding of stack and queue
- Develop your skill of writing java programs implementing and using stack and queue

2. Description

Part I. Implement Stack and Queue

Modify the class `LinkedStack<E>` given in class in the following ways.

- Add the following methods:
 - A constructor that loads the stack from an array parameter The last array element should be at the top of the stack
 - A method named **peekNextTop** that allows you to retrieve the element just below the one at the top of the stack without removing it. It should return null if the stack has just one element, and it should throw an exception if the stack is empty.
- **NOTE: You CAN NOT add or remove data fields to `LinkedStack<E>` class. You CAN NOT change the definition of `Node` class.**

Modify the class `ListQueue<E>` given in class (to be covered in Lec#12) in the following ways.

- Add the following methods:
 - `public int size()`
Return the size of the current queue (i.e. this `ListQueue` object, also the calling `ListQueue` object).
 - `public boolean empty()`
Return true if the queue is empty, false otherwise.
- **NOTE: You CAN NOT add or remove data fields to `ListQueue<E>` class. You CAN NOT change the definition of `Node` class.**

Part II. Implement Stack and Queue

Write a java application called `HiringApp` that you will use to hire and fire workers for a company. **The application `HiringApp` MUST use EITHER of the following:**

- **The predefined implementation of Stack and Queue data structures in Java API in the `java.util` package**
 - **Interfaces: `Deque` and `Queue`**
 - **And classes: `ArrayDeque` or `Stack` for stack and `LinkedList` for queues**
- **The `LinkedStack<E>` and `LinkedQueue<E>` classes that are implemented in the book and given in the lectures.**

The rules of hiring and firing are described as follows:

- 1) If you are asked to fire somebody at a time when the firm has no employees, you should notify your supervisor (i.e. print a related message).
- 2) If you are asked to fire somebody when the firm has 1 or more employees, you must fire the most recently hired.
- 3) You are to keep a list of applicants and the order in which they applied.
- 4) When you are asked to hire someone, if anybody has been fired, the most recently fired must be re-hired.
- 5) If there is nobody who has been fired, then the person who applied earliest is to be hired.
- 6) If there is nobody available for hiring, then you must notify your supervisor (i.e. print a related message).

The program should have a simple menu that allows you to specify the three actions:

- Accept application
- Hire
- Fire
- Quit

"Accept application" should prompt you for the name of an applicant and add that person's information to an appropriate data structure.

"Hire" should choose the appropriate person to hire, print his or her name, and appropriately update the internal data structures.

"Fire" should choose the appropriate person to fire, print his or her name, and appropriately update the internal data structures.

"Quit" should exit the program.

The minimum requirement for coding an applicant or an employee is using a String object to represent his/her name.

Hints:

1. **The code that are related to this assignment and covered in the lecture section is in the attached Assign3RelatedCode.zip.**
2. **Before you start programming, you need to decide which data structures are appropriate for storing the people involved in this HiringApp.**

Bonus:

- **NOTES:**
 - If you have **NOT COMPLETED** the basic programming skills listed in the **NON-BONUS** part, please do **NOT attempt at the bonus part.**
 - If you have completed all **NON-BONUS** part, please verify with the TA that you are done correctly **before attempting at the bonus part.**
- Define a class Person with the following data fields and methods, and use an object of Person class to represent an applicant or employee in your HiringApp.
 - Data fields: Name, age, skills (like Java, C++, etc.)
 - A constructor
 - Getters and setters for all data fields
 - ToString() method – returns a string containing all information about a person
- **Hints: consider the following implementation for skills:**
 - A string of words separated by coma: "Java, C++"
 - An array or list of Strings: {"Java", "C++"}

3. Sample Output: (data shown in NON-BONUS format)

Action (1 to accept, 2 to hire, 3 to fire):3
Memo to supervisor: There is nobody to fire

Action (1 to accept, 2 to hire, 3 to fire):1
What is the applicant's name: Fred

Action (1 to accept, 2 to hire, 3 to fire):1
What is the applicant's name: Barney

Action (1 to accept, 2 to hire, 3 to fire):1
What is the applicant's name: Bambam

Action (1 to accept, 2 to hire, 3 to fire):2
Fred hired
Action (1 to accept, 2 to hire, 3 to fire):2
Barney hired

Action (1 to accept, 2 to hire, 3 to fire):3
Barney fired

Action (1 to accept, 2 to hire, 3 to fire):2
Barney hired

Action (1 to accept, 2 to hire, 3 to fire):2
Bambam hired

Action (1 to accept, 2 to hire, 3 to fire):2
Memo to Supervisor: There is nobody to hire

4. Submission Requirements & Grading

This assignment (both parts) is **due**

For Section 005: by 11:50PM, Friday, Oct. 12, 2018.

For Section 006: by 11:50PM, Wednesday, Oct. 10, 2018.

Please see the file CIS2168 F18 005-006 Assign3 Submission Requirements.pdf for more details.