OBJECT-ORIENTED PROGRAMMING MINI-PROJECT REPORT

Group 15 – Project 01

Demonstration of basic operations on List, Stack and Queue

Instructor: Prof. Nguyen Thi Thu Trang

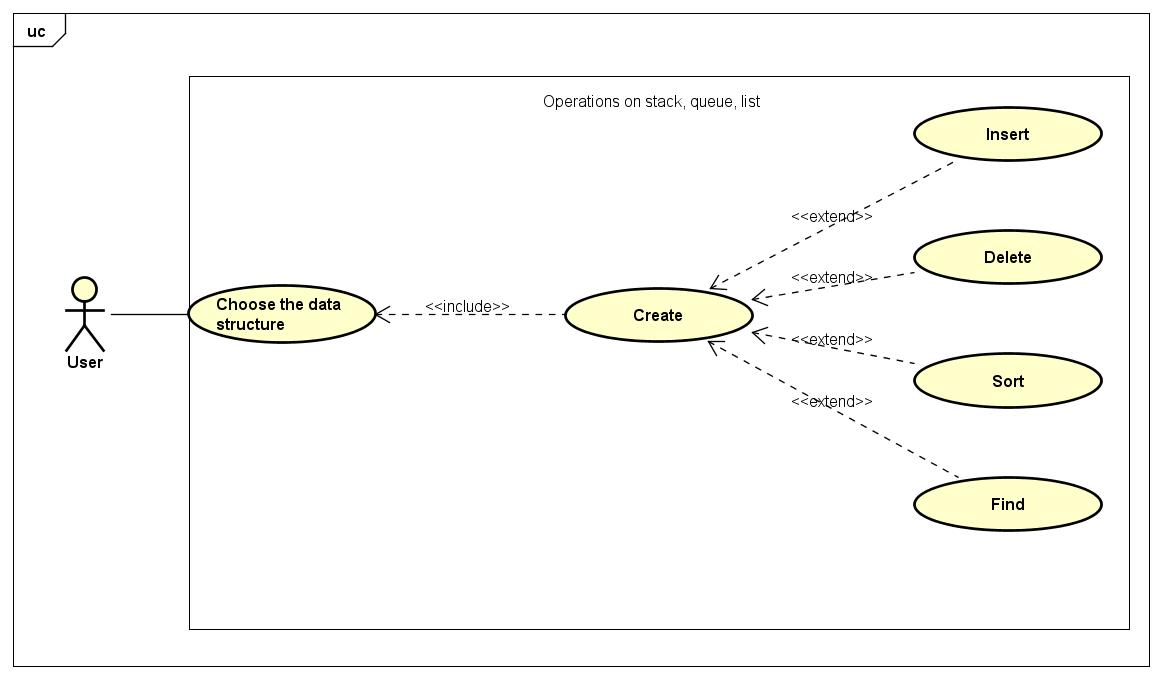
I. Mini-project description

In the world of computer science, List, Stack, and Queue are three foundational and widely used types of data structures. These structures provide essential tools for organizing and manipulating data efficiently. Our group has developed an application specifically designed to showcase the fundamental features of these data structures: creating, inserting, sorting, finding, and deleting elements.

II. Requirement

The main menu of our application showcases three buttons together with the general description of the three fundamental data structures: List, Stack, and Queue. By selecting any of these buttons, users can create a data structure of their chosen type and proceed to explore its basic operations, including insertion, sorting, finding, and deletion.

For assistance, a Help menu is available, accessible by clicking the Help button. User can also close the application at any time by using the Exit button.



*Use case diagram*

III. Design

We structure our application into two main packages: datastructure and gui.

1. Package datastructure

The datastructure package contains the abstract class DataStructure which the three other classes List, Stack and Queue inherited from. All the basic operations are implemented in this package.

A screenshot of a computer

Description automatically generated

datastructure package

2. Package gui

The gui package contains the classes used for demonstration. The main menu will be called from the GUIMain() in the GUIMain class, it contains three panels from the GUIPanel classes. Each GUIPanel class contains a button that will call upon a GUIFrame of the corresponding datastructure. There are also two helper classes, GUISlideshow used for demonstration purposed and TextFieldDemo used to implement the Help menu.

A yellow post-it note with black text

Description automatically generated

gui package

3. Relationship between two packages

Each GUIFrame class in the gui package aggregate its corresponding class in the datastructure package

A diagram of a computer

Description automatically generated

IV. Implementation

1. Encapsulation

We make sure to try to declare most of our instance variables as private, and we also minimize the use of setter and getter methods to a bare minimum.

2. Abstraction

We implement abstraction using the abstract class DataStructure and requires its subclasses to provide implementations for the abstract methods.

3. Inheritance

- List, Stack, Queue inherits from the abstract class DataStructure.

- GUIMain, GUIListFrame, GUIStackFrame, GUIQueueFrame inherits from JFrame.

- GUIListPanel, GUIStackPanel, GUIQueuePanel inherits from JPanel.

4. Polymorphism

All three classes List, Stack, Queue provide an implementation of several abstract methods from the DataStructure classes. These methods implemented by the child and the parent classes share the same name and parameters but they have different functionality.

V. Demo video

A demo video of our application can be found [here](https://youtu.be/ygdMFH8gqZ0).

VI. List of tasks

1. Nguyen Huu Nam 20210630

- Implement GUIMain, GUIQueue, Queue

- Create Use case diagram, General class diagram

- Writing README.md

- Writing report

2. Vu Tuan Minh 20210597

- Implement GUIList, List

- Create Class diagrams

- Implemented LinkedList (but it was ultimately excluded from the final version)

3. Vu Nhat Minh 20214919

- Implement GUIStack, Stack

- Record demo video

- Prepare slides for presentation

VII. References

1. Idea for the general structure of the program: OOP Lab excercises – AIMS Project.

2. Idea for the queue implementation: [Introduction and Array Implementation of Queue | geeksforgeeks](https://www.geeksforgeeks.org/introduction-and-array-implementation-of-queue/)

3. Idea for the Help menu: [Java Tutorial | Oracle](https://docs.oracle.com/javase/tutorial/uiswing/components/textfield.html)