

Chapter 4

ADT - Part 01

4.1 Bag ADT Review

In the first part of the lab, you will review the Bag ADT.

4.2 Stack ADT

In this part, you are given the Stack description, and you are required to provide your Stack ADT implementation using Python List.

4.3 Stack

A stack (sometimes called a “push-down stack”) is an ordered collection of items where the addition of new items and the removal of existing items always takes place at the same end. This end is commonly referred to as the “top.” The end opposite the top is known as the “base.”

4.3.1 Stack Description

The base of the stack is significant since items stored in the stack that are closer to the base represent those that have been in the stack the longest. The most recently added item is the one that is in position to be removed first. This ordering principle is sometimes called LIFO, last-in first-out. It provides an ordering based on length of time in the collection. Newer items are near the top, while older items are near the base.

Many examples of stacks occur in everyday situations. Almost any cafeteria has a stack of trays or plates where you take the one at the top, uncovering a new tray or plate for the next customer in line. Imagine a stack of books on a desk. The only book whose cover is visible is the one on top. To access others in the stack, we need to remove the ones that are sitting on top of them.

4.3.2 Lab Requirements

Given the Previous Stack description. Identify the following:

1. Stack ADT Specifications
2. Most suitable data structure to implemnt the Stack

3. Stack implementation

Chapter 5

ADT - Part 02

5.1 Programming Projects

5.1.1 Click Counter

A click counter is a small hand-held device that contains a push button and a count display. To increment the counter, the button is pushed and the new count shows in the display. Clicker counters also contain a button that can be pressed to reset the counter to zero. Design and implement the Counter ADT that functions as a hand-held clicker.

5.1.2 Grab Bag

A Grab Bag ADT is similar to the Bag ADT with one difference. A grab bag does not have a `remove()` operation, but in place of it has a `grab_item()` operation, which allows for the random removal of an item from the bag. Implement the Grab Bag ADT.

5.1.3 Counting Bag

Counting Bag ADT is just like the Bag ADT but includes the `num_of(item)` operation, which returns the number of occurrences of the given item in the bag. Implement the Counting Bag ADT and defend your selection of data structure.

5.2 Queue ADT

In this part, you are given the Queue description, and you are required to provide your Queue ADT implementation using Python List.

5.2.1 Queue Description

A queue is an ordered collection of items where the addition of new items happens at one end, called the “rear,” and the removal of existing items occurs at the other end, commonly called the “front.” As an element enters the queue it starts at the rear and makes its way toward the front, waiting until that time when it is the next element to be removed.

The most recently added item in the queue must wait at the end of the collection. The item that has been in the collection the longest is at the front. This ordering principle is sometimes called FIFO, first-in first-out. It is also known as “first-come first-served.”

The simplest example of a queue is the typical line that we all participate in from time to time. We wait in the check-out line at a grocery store, and we wait in the cafeteria line (so that we can pop the tray stack). Well-behaved lines, or queues, are very restrictive in that they have only one way in and only one way out. There is no jumping in the middle and no leaving before you have waited the necessary amount of time to get to the front.

5.2.2 Lab Requirements

Given the Previous Queue description. Identify the following:

1. Queue ADT Specifications
2. Most suitable data structure to implement the Queue
3. Queue implementation