# Lab 1 Introduction

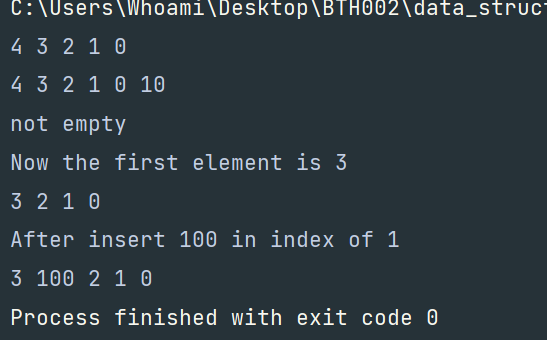
This lab work is about the data structure List, linked list, stack and queue.

I create 7 classes, and they are

* Alist
* LinkedList
* Node
* Stack
* Queue
* PeopleNode
* Circle

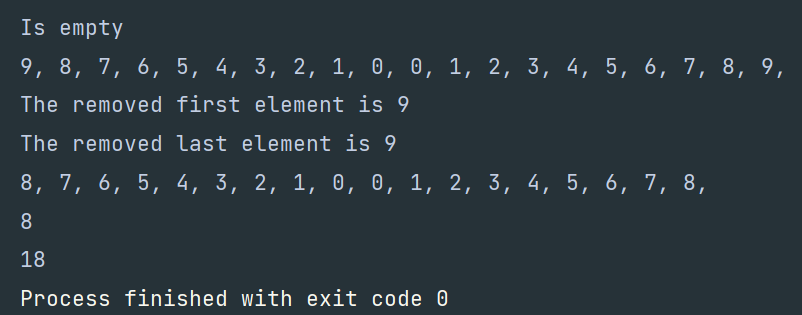
AList

The class AList is to store data like a array linearly, but the element in it can be added or removed in the first or last, get the Nth data and so on. The base implement method is using array.



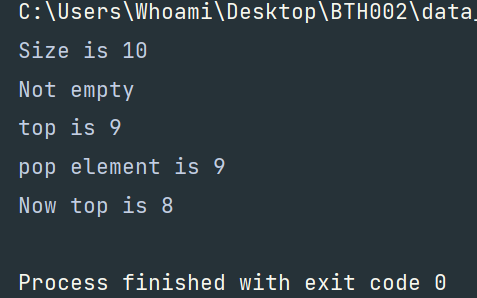
Node and LinkedList

These two classes are about doubly linked list. It is more convenient to do some operations such as add an element in the first or the last of this list vice versa the remove operation. It is used mostly the pointers to get data in the memory because they are not in succession.



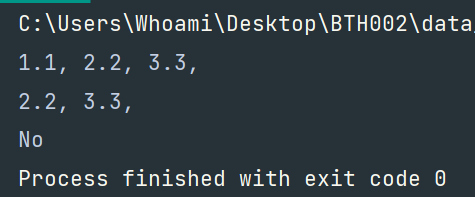
Stack

This class is about the classical data structure, stack. Each data in the stack should be last out if it is first in. Like a box. It is implemented by array, too. But, it has some different method like pop and peek. We pop the top element in the stack and also we peek it, but don’t pop it.



Queue

It is implemented by array, too. It is different from stack, element first in, must first out, like a real queue we are waiting for lunch in dining hall. Specifically, it is circle queue, once the array is not full, back and head pointer can cross and change position when next data is being added.



PeopleNode and Circle

It is about the Joseph Circle problem and the best match data structure is circle linked list. Circle is like the list and PeopleNode is like the Node.



Q&A

A pointer points an object, if the object is deleted by the programmer, the pointer will not become the nullptr, we need to set it to null instead. And the address cell in memory back to the resource pool. But the data is still in there.