Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«Алгоритмы и структуры данных: Списковые структуры данных.»

Проверил:
Сентерев Ю.А. _____
«_____» _____ 201_г.

Выполнил:
Студент группы P3255
Кабардинов Д. В. _____

Оценка _____

Санкт-Петербург

2019

Цель работы:

- исследовать и изучить списковые структуры данных и их основные процедуры;
- овладеть умениями и навыками написания программ по исследованию списковых структур данных и их основных процедур на языке программирования Python;

Задание

5. Удалить n-ый элемент из списка.
15. Задача Джозефуса: n воинов из одного войска убивают каждого m-го из другого. Требуется определить номер k начальной позиции воина, который должен будет остаться последним.

Ход выполнения работы

Реализация односвязного списка:

```
class SingleLinkedList:
        def __init__(self):
                self.lst = None
        # Inserts node with info propery = value to the beginning of the list
        def unshift(self, value):
                node = ListNode(value)
                if self.lst != None:
                        node.setPointerTo(self.lst)
                self.lst = node
        # Removes first node of the list
        # Returns removed node's 'info' property
        def shift(self):
                if self.lst == None:
                        return None
                else:
                        data = self.lst.info
                        self.lst = self.lst.ptr
                        return data

        # Inserts node which info property is set to "value" param
        # after the node to which pointer p refers
        def insertAfter(self, p, value):
                node = ListNode(value)
                node.setPointerTo(p.ptr)
                p.setPointerTo(node)
        # Removes list node to which pointer of p node refers
        # Returns removed node's 'info' property
        def removeAfter(self, p):
                if p.ptr == None:
                        return None
                else:
                        node = p.ptr
                        p.setPointerTo(node.ptr)
                        return node.info
```

```
class ListNode:
        ptr = None

        def __init__(self, info):
                self.info = info

        def setPointerTo(self, node):
                self.ptr = node
```

Функция, решающая задачу:

```
# Removes n-th element from the list
# n is 1 based. So if it was an array, n = 1 would mean that
# elemenent at index 0 would be deleted.
# Returns info property of removed node
def removeNth(list, n):
        #find a node before the one that we need to remove, i.e. n - 1
        node = list.lst
        if n > 1:
                for i in range(n - 2):
                        node = node.ptr
                        if node == None:
                                return
                return list.removeAfter(node)
        else: # n == 1 - removing first element
                return list.shift()
```

Проверка правильности работы функции:

```
# Testing:
testList = SingleLinkedList()
testList.unshift('d')
testList.unshift('c')
testList.unshift('b')
testList.unshift('a')
# list now have structure: abcd
removeNth(testList, 2) # removing "b" letter
#printing to check
letter = testList.lst
while letter.ptr != None:
        print(letter.info)
        letter = letter.ptr
print(letter.info)
# expected result: acd
```

Выводы

В ходе выполнения работы мной была реализована структура данных Стек, а также функция добавляющая элемент в середину стека. Т.о. работа выполнена в полном объёме в соответствии с заданием. В результате получены навыки программирования на языке Python и изучена широко применяемая на практике структура данных - Стек.

Список используемой литературы:

1. `https://tinyurl.com/y3aj6pku`
   - Linked Lists in Detail with Python Examples: Single Linked Lists
2. `https://docs.python.org/3.5/tutorial/index.html` - The Python Tutorial
3. Стивен Скиена - Алгоритмы. Руководство по разработке
4. Никлаус Вирт - Алгоритмы и структуры данных
5. Томас Кормен - Алгоритмы. Построение и анализ
6. `https://www.pythoncentral.io/singly-linked-list-insert-node/` - Python Data Structures Tutorial