

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 Р3255
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 property = 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
# element 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

```

```

dmitrii@kdv:~/projects/TeX/algs/lab2$ python3 single-linked-list.py

```

```

a
c
d

```

Кольцевые списки

```
class CircularList:
    # receives ListNode to be the first element in the list
    def __init__(self, node):
        self.lst = node
        node.setPointerTo(node)
    # Inserts node which info property is set to "value" param
    # after the node to which pointer p refers
    def insertAfter(self, value, p):
        node = ListNode(value)
        node.setPointerTo(p.ptr)
        p.setPointerTo(node)
        return node

    # Removes list node to which pointer of p node refers
    # Returns removed node's 'info' property
    def removeAfter(self, p):
        node = p.ptr
        if node == self.lst:
            self.lst = p
        p.setPointerTo(node.ptr)
        return node.info
```

Решение задачи Джозефуса:

```
# In a circle of n elements
# execute (remove) every m-th node (m > 1)
# returns k - original position of last remaining node (position is 0 - based)
def josephus(n, m):
    # create circular list with n elements
    for i in range(n):
        if i == 0:
            firstNode = ListNode(i)
            circle = CircularList(firstNode)
            last = firstNode
        else:
            last = circle.insertAfter(i, last)

    #start deleting every m-th item
    counter = n
    current = circle.lst
    while counter != 1:
        for j in range(m - 2):
            current = current.ptr
        circle.removeAfter(current)
        current = current.ptr
        counter -= 1
    return current.info
```

Проверка правильности работы программы:

```
print(josephus(10, 3)) # expected result 3
```

```
dmitrii@kdv:~/projects/TeX/algs/lab2$ python3 lab2-circular-list.py  
3
```

Выводы

В ходе выполнения работы мной были реализованы такие структуры данных как односвязный и кольцевой списки, а также функция удаляющая n-ый элемент односвязного списка и функция, решающая задачу Джозефуса с использованием кольцевого списка. Т.о. работа выполнена в полном объёме в соответствии с заданием. В результате получены навыки программирования на языке Python и изучены списковые структуры данных.

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