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 №1

«Алгоритмы и структуры данных: Полустатические структуры данных. Стек.»

Проверил:	Выполнил:
Сентерев Ю.А	Студент группы Р3255
«» 201_г.	Кабардинов Д. В
Опенка	

## Цель работы:

- исследовать и изучить полустатические структуры данных (на примере стеков, реализованных с помощью массивов);
- овладеть навыками разработки алгоритмов и написания программ по исследованию стеков на языке программирования Python;

### Задание

Ввести символы, формируя из них стек.

5. Вставить символ '\*' в середину стека, если число элементов четное, или после среднего элемента, если число элементов нечетное.

return self.stack[len(self.stack) - 1]

#### Ход выполнения работы

Реализация стека:

```
class Stack:
        def __init__(self):
                self.stack = []
        def push(self, elem):
                self.stack.append(elem)
        def pop(self):
                if len(self.stack) == 0:
                        return None
                else:
                        return self.stack.pop()
        def empty(self):
                return len(self.stack) == 0
        def stackTop(self):
                if self.empty():
                        return None
                else:
```

Создание стека и добавление в него элементов:

```
Функция, добавляющая элемент в середину стека:
# works only for stacks with length > 1
def insertInTheMiddle(stack, elementToInsert):
        helperStack = Stack()
        size = 0
        while stack.empty() == False:
                helperStack.push(stack.pop())
                size += 1
        if (size % 2) == 0:
                #size is even
                position = size / 2
        else:
                #size is odd
                position = (size // 2) + 1
        for i in range(size + 1):
                if i == position:
                        stack.push(elementToInsert)
                else:
                        stack.push(helperStack.pop())
        return stack
Проверка результатов выполнения программы:
insertInTheMiddle(charactersStack, '*')
# Checking the result:
while charactersStack.empty() == False:
        print(charactersStack.pop())
#expected result:
                          2
#
                          1
#test the function on odd sized stack:
charactersStack2 = Stack()
for i in range(5):
        charactersStack2.push(i)
insertInTheMiddle(charactersStack2, '*')
while charactersStack2.empty() == False:
        print(charactersStack2.pop())
#expected result
                          4
                          3
#
#
                          2
#
                          1
```

0

#

#### 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://www.pythoncentral.io/the-difference-between-a-list-and-an-array/- Python Central: Python Programming Guides and Tutorials
- 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.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html Carnegie Mellon University Web Site Stacks and Queues