

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №6**

**Замыкания в языке Python**

**По дисциплине «Технологии программирования и алгоритмизация»**

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р. А. \_\_\_\_\_

(подпись)

Ставрополь 2021

**Цель работы:** приобретение навыков по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.

**Ход работы:**

**Ссылка на репозиторий:** <https://github.com/DIMITRY-GALYAS1/laba-2.11.git>

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Выполнил первый пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def fun1(a):
    x = a * 3

    def fun2(b):
        nonlocal x
        return b + x
    return fun2

if __name__ == '__main__':
    test_fun = fun1(4)
    print(test_fun(7))
```

Рисунок 1. Пример 1

3. Выполнил второй пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

tpl = lambda a, b: (a, b)
a = tpl(1, 2)
print(a)
b = tpl(3, a)
print(b)
c = tpl(a, b)
print(c)
```

primer\_2 x

C:\Users\lizeq\anaconda3\envs\pythonProject6\py  
(1, 2)  
(3, (1, 2))  
((1, 2), (3, (1, 2)))

Process finished with exit code 0

Рисунок 2. Пример 2

4. Выполнил индивидуальное задание.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
Используя замыкания функций, объявите внутреннюю функцию, которая заключает
строку s ( s – строка, параметр внутренней функции) в произвольный тег,
содержащийся в переменной tag – параметре внешней функции. Далее, на вход
программы поступает две строки: первая с тегом, вторая с некоторым содержимым.
Вторую строку нужно поместить в тег из первой строки с помощью реализованного
замыкания. Результат выведите на экран.
"""

def fun1():
    def fun2(tag, s):
        tag1 = '<' + tag + '>'
        tag2 = tag1[0] + '/' + tag1[1::]
        return "".join([tag1, s, tag2])
    return fun2

if __name__ == '__main__':
    print(fun1()(input('Введите тег: '), input('Введите строку: ')))
```

Рисунок 6. Код индивидуального задания

```
C:\Users\lizeq\anaconda3\envs\pythonProject6\python.exe C:/Users/lizeq/PycharmProjects/pythonProject6/individual_1.py
Введите тег: <p>
Введите строку: Python
<p>Python</p>

Process finished with exit code 0
```

Рисунок 7. Выполнение индивидуального задания

### Контрольные вопросы:

1. Что такое замыкание?

Замыкание (closure) в программировании — это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции в окружающем коде и не являющиеся ее параметрами.

2. Как реализованы замыкания в языке программирования Python?

В Python замыкания реализованы путём вложенных функций, где аргумент отправляется только во внешнюю функцию, а внутренняя функция, несмотря на отсутствие параметров успешно выполняет операции с этим аргументом.

3. Что подразумевает под собой область видимости Local?

Эту область видимости имеют переменные, которые создаются и используются внутри функций.

4. Что подразумевает под собой область видимости Enclosing?

Суть данной области видимости в том, что внутри функции могут быть вложенные функции и локальные переменные, так вот локальная переменная функции для ее вложенной функции находится в enclosing области видимости.

5. Что подразумевает под собой область видимости Global?

Переменные области видимости global – это глобальные переменные уровня модуля

6. Что подразумевает под собой область видимости Build-in?

Уровень Python интерпретатора. В рамках этой области видимости находятся функции open, len и т. п., также туда входят исключения. Эти сущности доступны в любом модуле Python и не требуют предварительного импорта. Built-in – это максимально широкая область видимости.

7. Как использовать замыкания в языке программирования Python?

Рассмотрим на примере:

```
>>> def mul(a):  
def helper(b):  
return a * b  
return helper  
>>> mul(5)(2)
```

Программа перемножает переданные в неё числа путём замыкания в функции.

8. Как замыкания могут быть использованы для построения иерархических данных?

В книге “Структура и интерпретация компьютерных программ” Айбельсона Х., Сассмана Д.Д. свойство замыкания определяется так: “В общем случае, операция комбинирования объектов данных обладает свойством замыкания в том случае, если результаты соединения объектов с помощью этой операции сами могут соединяться этой же операцией”. Это свойство позволяет строить иерархические структуры данных.

**Вывод:** в ходе выполнения лабораторной работы приобрел навыки по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.