

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

Погорелов Д.Н. « » \_\_\_\_\_ 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:**

1. Изучил теоретический материал для выполнения лабораторной работы.
2. Создал общедоступный репозиторий
3. Проработал примеры, приведённые в методическом материале.

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def fun1(a):
5          x = a * 3
6
7          def fun2(b):
8              nonlocal x
9              return b + x
10         return fun2
11
12
13     test_fun = fun1(4)
14     print(test_fun(7))
15     |
```

Рисунок 1 – Код первого примера

```
19
|
Process finished with exit code 0
```

Рисунок 2 – Результат первого примера

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5      tpl = lambda a, b: (a, b)
6      a = tpl(1, 2)
7      print(a)
8      b = tpl(3, a)
9      print(b)
10     c = tpl(a, b)
11     print(c)
12

```

Рисунок 3 – Код второго примера

```

(1, 2)
(3, (1, 2))
((1, 2), (3, (1, 2)))

Process finished with exit code 0

```

Рисунок 4 – Результат выполнения кода второго примера

### Индивидуальное задание. Вариант 1.

1. Изучив теоретический материал и проработав примеры, приступил к выполнению индивидуального задания.

Условие: используя замыкания функций, определите вложенную функцию, которая бы увеличивала значение переданного параметра на 3 и возвращала бы вычисленный результат. Вызовите внешнюю функцию для получения ссылки на внутреннюю функцию и присвойте ее переменной с именем `cnt`. Затем, вызовите внутреннюю функцию через переменную `cnt` со значением `k`, введенным с клавиатуры.

2. Написал код для реализации задачи.

```

13 def fun1(a):
14     def fun2():
15         return a + 3
16     return fun2()
17
18
19 if __name__ == '__main__':
20     # Запрашиваем переменную с клавиатуры
21     k = float(input("Введите число: "))
22
23     # Присваиваем внешней функции новую переменную
24     cnt = fun1(k)
25
26     # Вызов внутренней функции через переменную cnt
27     print(cnt)
28

```

Рисунок 5 – Код решения индивидуального задания

```

Введите число: 7
10.0

Process finished with exit code 0

```

Рисунок 6 – Результат работы кода

### 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.

#### 4. Что подразумевает под собой область видимости Local?

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

## 5. 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.

## 6. Что подразумевает под собой область видимости Global?

Переменные области видимости global – это глобальные переменные уровня модуля

## 7. Что подразумевает под собой область видимости Built-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.

## 8. 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.

## 9. 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.