

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.11 Замыкания в
языке Python»**

ОТЧЕТ
по лабораторной работе №14
дисциплины
«Основы программной инженерии»

Выполнил:
Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 2.11 Замыкания в языке Python

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

Ход работы:

Индивидуальное задание

Используя замыкания функций, объявите внутреннюю функцию, которая принимает два параметра `a`, `b`, а затем, возвращает строку в формате: «Для значений `a`, `b` функция `f(a,b)` = `<число>`» где `число` – это вычисленное значение функции `f`. Ссылка на `f` передается как аргумент внешней функции. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы. Функцию `f` придумайте самостоятельно (она должна что то делать с двумя параметрами `a`, `b` и возвращать результат).

```
1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def fun1(a):
8      def fun2(b):
9          f = a * math.pow(b, a)
10         l = f"Для значений {a},{b} функция f(a,b) = {f}"
11         return l
12
13     return fun2
14
15
16  ▶ if __name__ == "__main__":
17      test_fun = fun1(2)
18      print(test_fun(3))
```

Рисунок 1 – Код программы для индивидуального задания

```
E:\GitHub\laba14\user\Scripts\python.exe
Для значений 2,3 функция f(a,b) = 18.0

Process finished with exit code 0
```

Рисунок 2 – Результат работы программы

Вывод: приобретены навыки по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.

Ответы на 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?

```
>>> def fun1(a):  
    x = a * 3  
    def fun2(b):  
        nonlocal x  
        return b + x  
    return fun2  
  
>>> test_fun = fun1(4)  
  
>>> test_fun(7)
```

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 (модуль – это файл с расширением .py).

6. 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.

7. Как использовать замыкания в языке программирования Python?

```
>>> def fun1(a):  
    x = a * 3  
    def fun2(b):  
        nonlocal x  
        return b + x  
    return fun2  
  
>>> test_fun = fun1(4)  
  
>>> test_fun(7)
```

8. Как замыкания могут быть использованы для построения иерархических данных? 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. Следовательно, с помощью замыканий объекты одного типа могут содержать объекты такого же типа.