

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.11

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Примаков В. Д. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

ВЫПОЛНЕНИЕ

Пример 1

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

```
C:\Users\vadym\AppData\Local\Programs\Python\Python39\python.exe  
19  
  
Process finished with exit code 0
```

Индивидуальное задание

8. 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).

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
def func1(a, b):  
    x = a  
    y = b  
  
    def func2():  
        nonlocal x, y  
        return f"Для значений {x}, {y} функция f({x},{y}) = {x + y}"  
    return func2  
  
a = int(input("a? "))  
b = int(input("b? "))  
test = func1(a, b)  
print(test())
```

```
a? 8  
b? 7  
Для значений 8, 7 функция f(8,7) = 15  
  
Process finished with exit code 0
```

Ссылки на репозитории

GitHub - https://github.com/surai5a/laba_2_11

Ответы на контрольные вопросы

1. 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. В python замыкание реализуется с помощью внутренних функций, например:

```
def mul(a):  
    def helper(b):  
        return a * b  
    return helper
```

3. Область видимости Local имеют переменные, которые создаются и используются внутри функций.
4. Суть области видимости Enclosing 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. Переменные области видимости global – это глобальные переменные уровня модуля.
6. Built-in – это максимально широкая область видимости. 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.
7. [Ссылка](#)
8. «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.