

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

Институт цифрового развития Кафедра
инфокоммуникаций

«Замыкания в языке Python»

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

Выполнил:

Логвинов Иван васьильевич
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.

Проработка примера из лабораторной работы:

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

```
C:\Users\Иван\AppData\Local\Programs\  
19  
  
Process finished with exit code 0
```

Рисунок 1 – Код и результат работы примера

Индивидуальное задание: Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента коллекцию (список или кортеж) и возвращает или минимальное значение, или максимальное, в зависимости от значения параметра type внешней функции. Если type равен «max», то возвращается максимальное значение, иначе – минимальное. По умолчанию type должно принимать значение «max». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
def func_1(type_1='max'):  
    def func_2(b):  
        if type_1 == 'max':  
            return max(b)  
        else:  
            return min(b)  
    return func_2  
  
if __name__ == "__main__":  
    test_list = [1, 2, 3, 5, 0, -13, 100]  
    test_tuple = [1.5, 2, 123, 5, 0, -167, 100]  
  
    print(f'Max of entered list: {func_1()(test_list)}')  
    print(f'Min of entered tuple: {
```

```
C:\Users\Иван\AppData\Local\Programs\  
Max of entered list: 100  
Min of entered tuple: -167  
  
Process finished with exit code 0
```

Рисунок 2 – Код и результат работы индивидуального задания

Контрольные вопросы

1. Что такое замыкание?

Для начала обратимся к википедии: “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.” Перед тем как перейти к рассмотрению примеров реализации замыканий на Python, для начал вспомним тему “область видимости переменных”. Обычно, по области видимости, переменные делят на глобальные и локальные. Глобальные существуют в течении всего времени выполнения программы, а локальные создаются внутри методов, функций и прочих блоках кода, при этом, после выхода из такого блока переменная удаляется из памяти.

2. Как реализованы замыкания в языке программирования Python?

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

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 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?

```
>>> def fun1(a):  
    x = a * 3  
    def fun2(b):  
        nonlocal x  
        return b + x  
    return fun2  
  
>>> test_fun = fun1(4)  
  
>>> test_fun(7)  
19
```

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

Теперь перейдем с уровня математики на уровень функционального программирования. Вот как определяется “свойство замыкания” в книге “Структура и интерпретация компьютерных программ” Айбельсона Х., Сассмана Д.Д.: “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. Покажем это на примере кортежей в Python.