

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:

Борсуков Владислав Олегович
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:

```
1  def fun1(a):
2      x = a * 3
3
4      def fun2(b):
5          nonlocal x
6          return b + x
7      return fun2
8
9
10 test_fun = fun1(4)
11 print(test_fun(7))
```

Run: Example x

C:\Users\Borsukov\Desktop\LR14\PyCharm\example.py
19

Process finished with exit code 0

Рисунок 1 – Код и результат работы примера

Индивидуальное задание: 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 `snt`. Затем, вызовите внутреннюю функцию через переменную `snt` со значением `k`, введенным с клавиатуры.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 """
5     Используя замыкания функций, определите вложенную функцию, которая бы увеличивала
6     значение переданного параметра на 3 и возвращала бы вычисленный результат. Вызовите
7     внешнюю функцию для получения ссылки на внутреннюю функцию и присвойте ее
8     переменной с именем cnt. Затем, вызовите внутреннюю функцию через переменную cnt со
9     значением k, введенным с клавиатуры.
10 """
11
12 def fun1():
13     def fun2(k):
14         return k + 3
15     return fun2
16
17
18 ▶ if __name__ == "__main__":
19     cnt = fun1()
20     kk = int(input("Введите k: "))
21     print(cnt(kk))
22
23 if __name__ == "__main__":
24     cnt = fun1()
25     kk = int(input("Введите k: "))
26     print(cnt(kk))
```

Run: Individual ×

C:\Users\Borsukov\Desktop\LR14\PyCharm\venv\Scripts\python.exe C:\Users\Borsukov\Des
Введите k: 3
6

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 существует 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 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)  
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.