

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:

Зиёдуллаев Жавохир Эркин угли
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.

Пример 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def fun1(a):
    x = a * 3
    def fun2(b):
        nonlocal x
        return b + x
    return fun2

test_fun = fun1(4)

print(test_fun(7))
```

zamik1 x

C:\Users\work\PycharmProjects\pythonProject5\venv\S
19

Индивидуальное задание

. Используя замыкания функций, объявите внутреннюю функцию, которая преобразует строку из списка целых чисел, записанных через пробел, либо в список, либо в кортеж. Тип коллекции определяется параметром `type` внешней функции. Если `type = 'list'`, то используется список, иначе – кортеж. Далее, на вход программы поступает две строки: первая – это значение для параметра `type`; вторая – список целых чисел, записанных через

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def convert(type: str):
    def activate(nums: str):
        if type == 'list':
            return list(map(int, nums.split(' ')))
        if type == 'tuple':
            return tuple(map(int, nums.split(' ')))
    return activate

if __name__ == '__main__':
    print(f"List: {convert('list')('1 2 3 4 5 6')}\n"
          f"Tuple: {convert('tuple')('1 2 3 4 5 6')}")
```

zamikindiv x

C:\Users\work\PycharmProjects\pythonProject5\venv\Scripts\python.exe C

List: [1, 2, 3, 4, 5, 6]
Tuple: (1, 2, 3, 4, 5, 6)

GitHub:

<https://github.com/javoxir21/laba-s14.git>

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.

Функция `outer` определяется с функцией `inner` внутри, а функция `outer` возвращает функцию `inner`; именно она – возвращаемое значение `outer`.

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.