

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №14 по  
дисциплине: основы программной инженерии**

Выполнил:

студент группы ПИЖ-б-о-21-1

Прокопов Дмитрий Владиславович

Проверил:

доцент кафедры инфокоммуникаций

Воронкин Р.А.

Ставрополь, 2022 г.

## ВЫПОЛНЕНИЕ

### Пример

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

1 x

C:\Users\Vova\AppData\Local\Progra  
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

пробел. С помощью реализованного замыкания преобразовать эту строку в соответствующую коллекцию. Результат работы замыкания выведите на экран.

```
im.py x ind.py x
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def con(type: str):
    def act(nums: str):
        if type == 'list':
            return list(map(int, nums.split(' ')))
        if type == 'tuple':
            return tuple(map(int, nums.split(' ')))

    return act

if __name__ == "__main__":
    print(f"List: {con('list')('3 7 5 9 1')}")
    print(f"Tuple: {con('tuple')('5 7 1 3 6 8')}")
```

```
ind x
C:\Users\Dmitriy\PycharmProjects\lb14\venv\Scripts\python
List: [3, 7, 5, 9, 1]
Tuple: (5, 7, 1, 3, 6, 8)
```

### 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.