

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования  
Отчет по лабораторной работе №2.11**

**Замыкания в языке Python**

Выполнил студент группы  
ИТС-б-о-21-1 (2)

Якупов Э.А «        » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена «    » \_\_\_\_\_ 20\_\_ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.

**Ссылка на репозиторий** - <https://github.com/Blekroyt/Fox1.git>

**Ход работы:**

### Пример 1.

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> #!/usr/bin/env python3
>>> # -*- кодировка: utf-8 -*-
>>> import sys
>>>
>>> def add_four(a):
...     x = 2
...     def add_some():
...         print("x = " + str(x))
...         return a + x
...     return add_some()
...
>>> add_four(5)
x = 2
7
>>>
```

Рисунок 1. Результат выполнения программы

### Пример 2.

```
>>> x=4
>>> def fun():
...     print(x+3)
...
>>> fun()
7
```

Рисунок 2. Результат выполнения программы

### Пример 3.

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

Рисунок 3. Результат выполнения программы

**Индивидуальное задание. 9 вариант.** 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.

```
1  #!/usr/bin/env python3
2  # -*- кодировка: utf-8 -*-
3  import sys
4
5  def fun1(type_='max'):
6      def fun2(lst):
7          return eval(f'{type_}(lst)')
8      return fun2
9
10 a = [1, 2, 3, 4, 5, 65, 6,]
11 max_fun = fun1()
12 min_fun = fun1('min')
13 print(max_fun(a))
14 print(min_fun(a))
```

```
65
1
```

Рисунок 4. Результат выполнения программы

### Контрольные вопросы:

1. Что такое замыкание?

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. Как реализованы замыкания в языке программирования Python? В

Python выделяют четыре области видимости для переменных: `local`, `enclosing`, `global`, `build-in`.

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.

**Вывод:** в ходе лабораторной работы были приобретены навыки по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.