

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития

Кафедра инфокоммуникаций

дисциплина Языки программирования

Отчет по лабораторной работе №2.11

Замыкания в  
языке Python

Выполнил: студент группы ИТС-б-о-21-1  
Пушкин Максим Алексеевич

---

(подпись)

Проверил: кандидат технических наук, доцент кафедры  
инфокоммуникаций,  
Роман Александрович Воронкин

---

(подпись)

Ставрополь, 2022

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

## **2. Теоретическое сведения**

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.

Что касается Python, то тут выделяют четыре области видимости для переменных (с вашего позволения я буду использовать английские термины).

### **Область видимости *Local***

Эту область видимости имеют переменные, которые создаются и используются внутри функций.

### **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.

### **Область видимости *Global***

Переменные области видимости *global* – это глобальные переменные уровня модуля (модуль – это файл с расширением *.py*).

### **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.

Как уже было сказано выше, каждый раз, когда мы вызываем функцию, у нее создаются локальные переменные (если они у нее есть), а после завершения – уничтожаются, при очередном вызове эта процедура повторяется. Можно ли сделать так, чтобы после завершения работы функции, часть локальных переменных не

### **Метод и порядок выполнения**

4. Используя замыкания функций, объявите внутреннюю функцию, которая из переданного ей списка строк формирует многострочную строку вида: и возвращает ее. Где строка1, строка2,

<ol>

<li>строка\_1</li>

...

<li>строка\_N</li>

</ol>

... - это строки из переданного функции списка. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

Код:

```

8  ▶ if __name__ == '__main__':
9
10     def get_func(tag):
11         tag1 = "<" + tag + ">"
12         tag2 = "</" + tag + ">"
13         print("<ol>")
14         def func(string):
15             nonlocal tag1
16             nonlocal tag2
17             for i in range(len(string)):
18                 print(tag1 + " " + string[i] + " " + tag2)
19             print("</ol>")
20         return func
21
22     a = input().split()
23     new_tag = get_func("li")
24     new_tag(a)

```

Результат:

```

строка1 ... строкаN
<ol>
<li> строка1 </li>
<li> ... </li>
<li> строкаN </li>
</ol>

Process finished with exit code 0

```

Git:

[https://github.com/manulmax21/lab\\_rab1.git](https://github.com/manulmax21/lab_rab1.git)

## Вопросы для защиты работы

### 1. Что такое замыкание?

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?

### 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
```

*mul*

В функции *fun1()* объявлена локальная переменная *x*, значение которой определяется аргументом *a*. В функции *fun2()* используются эта же переменная *x*, *nonlocal* указывает на то, что эта переменная не является локальной, следовательно, ее значение будет взято из

ближайшей области видимости, в которой существует переменная с таким же именем. В нашем случае – это область `enclosing`, в которой этой переменной `x` присваивается значение `a * 3`. Также как и в предыдущем случае, на переменную `x` после вызова `fun1(4)`, сохраняется ссылка, поэтому она не уничтожается.

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

“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.

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