

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

Отчет о лабораторной работе №14 по дисциплине
«Основы программной инженерии»

Выполнил студент
2 курса, группы ПИЖ-б-о-20-1
Тотубалина С.С.

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

Ход работы

```
>>> def add_two(a):  
...     x = 2  
...     return a + x  
...  
>>> add_two(3)  
5  
>>> x  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'x' is not defined
```

Рис. 1 – использование локальной переменной

```
>>> 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
```

Рис. 2 – переменная область видимости enclosing

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

Рис. 3 – использование глобальных переменных

```

>>> def mul(a, b):
...     return a * b
...
>>> mul(3, 4)
12
>>> def mul5(a):
...     return mul(5, a)
...
>>> mul5(2)
10
>>> def mul(a):
...     def helper(b):
...         return a * b
...     return helper
...
>>> mul(5)(2)
10
>>> new_mul5 = mul(5)
>>> new_mul5
<function mul.<locals>.helper at 0x0000025A24803E20>
>>> new_mul5(2)
10

```

Рис. 4 – использование замыканий

```

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

```

Рис. 5 – использование функции nonlocal

```

1 ▶ 1 #!/usr/bin/env python3
2   2 # -*- coding: utf-8 -*-
3
4
5   5 def sample(string):
6       6 def name_surname(n, s):
7           7 sample_data = string.replace("%N%", n)
8           8 sample_data = sample_data.replace("%F%", s)
9           9 return sample_data
10
11      11 return name_surname
12
13
14 ▶ 14 if __name__ == "__main__":
15     15 sample_string = (
16         16 "Уважаемый %F% %N%! Вы делаете работу по замыканиям функции."
17     )
18     18 name, surname = input("Введите имя и фамилию: ").split()
19     19 print(sample(sample_string)(name, surname))

```

Рис. 6 – код программы individual_14.py (Вариант №22)

```

Введите имя и фамилию: Светлана Тотубалина
Уважаемый Тотубалина Светлана! Вы делаете работу по замыканиям функции.

Process finished with exit code 0

```

Рис. 7 – результат работы программы individual_14.py (Вариант №22)

Ответы на вопросы:

1. Что такое замыкание?

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?

- У нас должна быть вложенная функция (функция внутри функции).
- Вложенная функция должна ссылаться на значение, определенное в объемлющей функции.
- Объемлющая функция должна возвращать вложенную функцию.

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.

7. Как использовать замыкания в языке программирования Python?

```
def mul(a):  
    def helper(b):  
        return a * b  
    return helper
```

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