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:
Чернова Софья Андреевна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

1. Ход работы:

1.1 Пример 1 (рис. 1)

```
>>> 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
```

Рисунок 1 – использование локальной переменной

1.2 Пример 2 (рис. 2)

```
>>> 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 2 – переменная области видимости enclosing

1.3 Пример 3 (рис. 3)

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

Рисунок 3 – использование глобальных переменных

1.4 Пример 4 (рис. 4)

```
>>> 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
```

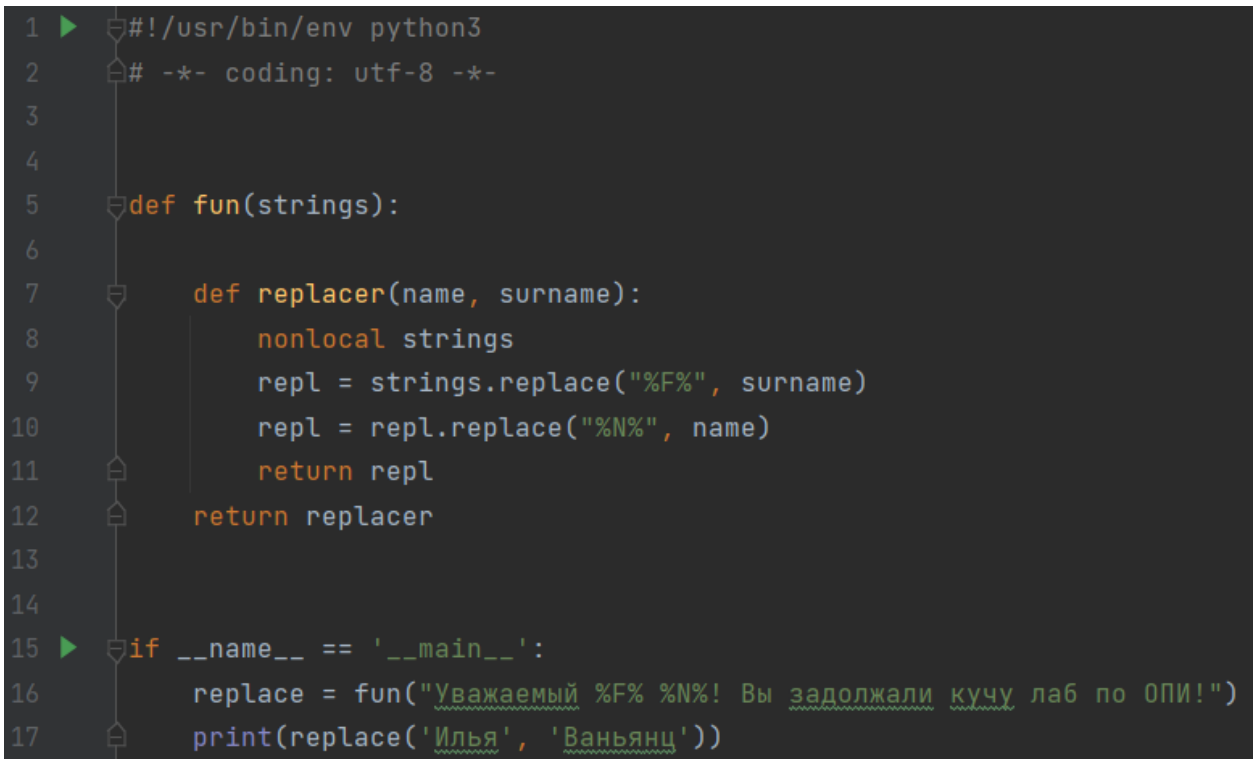
Рисунок 4 – использование замыканий

1.5 Пример 5 (рис. 5)

```
>>> 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 5 – использование функции nonlocal

1.6 Индивидуальное задание (рис. 6, 7)



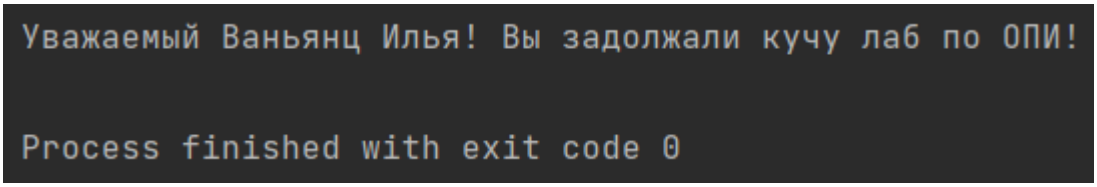
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


def fun(strings):

    def replacer(name, surname):
        nonlocal strings
        repl = strings.replace("%F%", surname)
        repl = repl.replace("%N%", name)
        return repl
    return replacer


if __name__ == '__main__':
    replace = fun("Уважаемый %F% %N%! Вы задолжали кучу лаб по ОПИ!")
    print(replace('Илья', 'Ваньянц'))
```

Рисунок 6 – код программы



```
Уважаемый Ваньянц Илья! Вы задолжали кучу лаб по ОПИ!

Process finished with exit code 0
```

Рисунок 7 – результат работы программы

2. Ответы на контрольные 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