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
Плешенко Данила Георгиевич
« »_____2022г.
Подпись студента_____
Работа защищена « »_____2022г.
Проверил к.т.н., доцент
Кафедры инфокоммуникаций
Воронкин Роман Александрович
_____
(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 - danilaple/lab.rab.2.11 (github.com)

Ход работы:

Проработка примеров лабораторной работы

Пример 1

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

Рисунок 1. Результат выполнения программы из примера 1

Пример 2

Рисунок 2. Результат выполнения программы из примера 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
```

Пример 3

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

Рисунок 3. Результат выполнения программы из примера 3

Пример 4

```
>>> def mul(a, b):
...     return a * b
...
>>> mul(3, 4)
12
>>> mul(5, 2)
10
>>> mul(5, 7)
35
```

Рисунок 4. Результат выполнения программы из примера 4

Пример 5

```
>>> def mul5(a):
...     return mul(5, a)
...
>>> mul5(2)
10
>>> mul5(7)
35
```

Рисунок 5. Результат выполнения программы из примера 5

Пример 6

```
>>> def mul(a):
...     def helper(b):
...             return a * b
...     return helper
...
>>> mul(5)(2)
10
```

Рисунок 6. Результат выполнения программы из примера 6

Пример 7

```
>>> new_mul5 = mul(5)
>>> new_mul5
<function mul.<locals>.helper at 0x0000025FF368FC70>
>>> new_mul5(2)
10
>>> new_mul5(7)
35
```

Рисунок 7. Результат выполнения программы из примера 7

Пример 8

```
>>> 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 8. Результат выполнения программы из примера 8

Индивидуальное задание (3 вариант)

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.

```
>>> def fun1(type):
...     def fun2(spisok):
...             if type == list:
...                     return list(map(int, spisok.split()))
...             return tuple(map(int, spisok.split()))
...     return fun2
...
>>> if __name__ == '__main__':
...     print(fun1(list)('1 2 3 4 5 6 7 8 9 10'))
...     print(fun1(tuple)('1 2 3 4 5 6 7 8 9 10'))
...
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

Рисунок 9. Результат выполнения индивидуального задания

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.

7. 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.

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