

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.11
дисциплины «Основы кроссплатформенного программирования»

Выполнила:
Мурашко Анастасия Юрьевна
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Замыкания в языке Python

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

Ход работы:

Вариант №10

Пример №1:

```
1  def add_four(a):
2      x = 2
3
4      def add_some():
5          print("x = " + str(x))
6          return a + x
7
8      return add_some()
9
10
11 print(add_four(5))
```

Рисунок 1. Код

```
↑ C:\Users\Monstr\PycharmProjects\pythonProject\venv
↓ x = 2
  7
⏮
⏭ Process finished with exit code 0
```

Рисунок 2. Результат выполнения

Пример №2:

```
1  def mul(a):
2      def helper(b):
3          return a * b
4
5      return helper
6
7
8  new_mul5 = mul(5)
9  print(new_mul5(2))
10 print(new_mul5(7))
```

Рисунок 3. Код

```
↑ C:\Users\MonstR\PycharmProjects\pythonProject\venv\Script
↓ 10
:U 35
⇅ Process finished with exit code 0
```

Рисунок 4. Результат выполнения

Пример №3:

```
1  def fun1(a):
2      x = a * 3
3      print(a)
4
5      def fun2(b):
6          nonlocal x
7          return b + x
8
9      return fun2
10
11
12 test_fun = fun1(4)
13 print(test_fun(7))
```

Рисунок 5. Код

```
↑ C:\Users\MonstR\PycharmProjects\pythonProject\venv\Scripts\python.exe
↓ 4
:U 19
⇅ Process finished with exit code 0
```

Рисунок 6. Результат выполнения

Индивидуальное задание:

10. Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента список целых чисел и удаляет из него все четные или нечетные значения в зависимости от значения параметра `type`. Если `type` равен «even», то удаляются четные значения, иначе – нечетные. По умолчанию `type` должно принимать значение «even». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

Рисунок 7. Условия задания

```

10 def main():
11     def delchis(list1):
12         def delchetnechet():
13             type = input('Введите even или noteven:\n')
14             if type == 'even':
15                 for x in range(len(list1)):
16                     if x % 2 == 0:
17                         list1.pop(list1.index(x))
18                         print(list1)
19             elif type == 'noteven':
20                 for x in range(len(list1)):
21                     if x % 2 != 0:
22                         list1.pop(list1.index(x))
23                         print(list1)
24             else:
25                 print('Введена не правильная команда')
26
27         return delchetnechet
28
29     f = delchis(list1=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
30     f()
31
32
33 if __name__ == '__main__':
34     main()

```

Рисунок 8. Код

```

C:\Users\MonstR\PycharmProjects\pythonProject\venv\Scripts\python.exe "D:/U
Введите even или noteven:
even
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 6, 7, 8, 9, 10]
[1, 3, 5, 7, 8, 9, 10]
[1, 3, 5, 7, 9, 10]
[1, 3, 5, 7, 9]

```

Рисунок 9. Результат выполнения

Ответы на 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?

С помощью функций.

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?

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). Но если мы этот модуль импортируем в каком-то другом модуле, то глобальная переменная для него уже не будет переменной уровня global.

6. Что подразумевает под собой область видимости Built-in?

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?

```
>>> new_mul5 = mul(5)

>>> new_mul5
<function mul.<locals>.helper at 0x000001A7548C1158>

>>> new_mul5(2)
10

>>> new_mul5(7)
35
```

Вызывая new_mul5(2), мы фактически обращаемся к функции helper(), которая находится внутри mul(). Переменная a, является локальной для mul(),

имеет область `enclosing` в `helper()`. Несмотря на то, что `mul()` завершила свою работу, переменная `a` не уничтожается, т.к. на нее сохраняется ссылка во внутренней функции, которая была возвращена в качестве результата.

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.

Вывод: Приобретены навыки по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.