

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе № 2.11

Замыкания в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.


Проверил(а) _____

Ставрополь 2021

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

Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Выполнены примеры из лабораторной, рис. 1-2



```
1.py x 2.py x ind1.py x
5 def fun1(a):
6     x = a * 3
7
8     def fun2(b):
9         nonlocal x
10        return b + x
11    return fun2
12
13
14 if __name__ == '__main__':
15     test_fun = fun1(4)
16     print(test_fun(7))
```

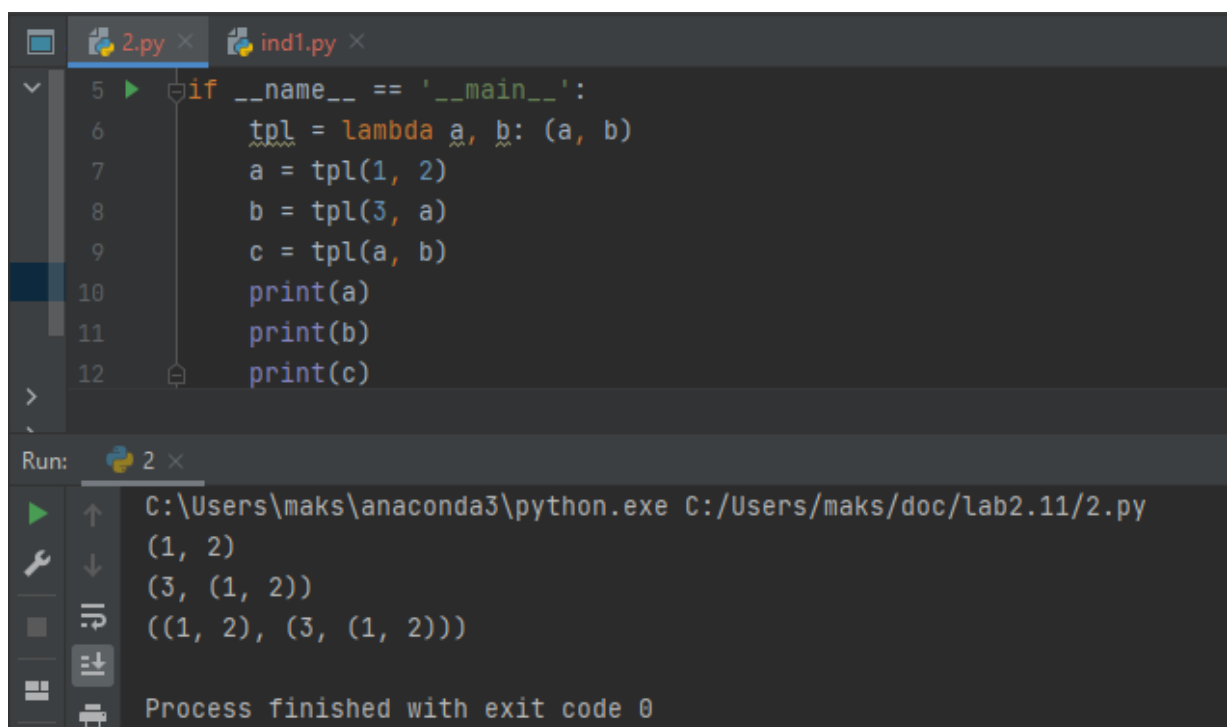
1 x

C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.11/1.py

19

Process finished with exit code 0

Рисунок 1 – Пример 1



```
2.py x ind1.py x
5 if __name__ == '__main__':
6     tpl = lambda a, b: (a, b)
7     a = tpl(1, 2)
8     b = tpl(3, a)
9     c = tpl(a, b)
10    print(a)
11    print(b)
12    print(c)
```

Run: 2 x

C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.11/2.py

(1, 2)

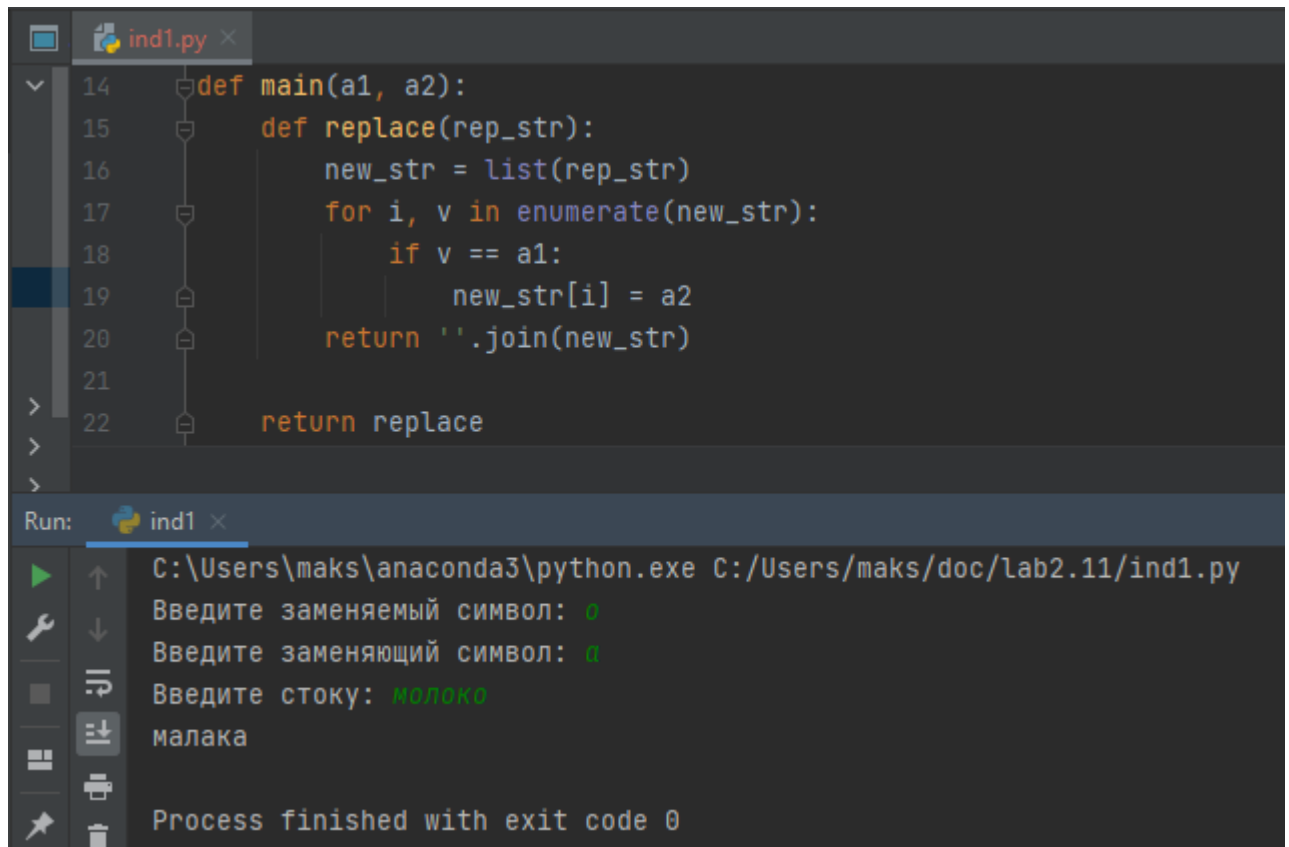
(3, (1, 2))

((1, 2), (3, (1, 2)))

Process finished with exit code 0

Рисунок 2 – Пример 2

Выполнено индивидуальное задание варианта 6, рис. 4



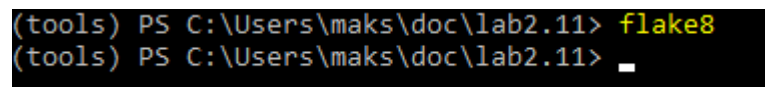
```
14 def main(a1, a2):
15     def replace(rep_str):
16         new_str = list(rep_str)
17         for i, v in enumerate(new_str):
18             if v == a1:
19                 new_str[i] = a2
20         return ''.join(new_str)
21
22     return replace
```

Run: ind1 x

```
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.11/ind1.py
Введите заменяемый символ: o
Введите заменяющий символ: a
Введите сток: молоко
малака
Process finished with exit code 0
```

Рисунок 3 – Индивидуальное задание

Программы проверены на flake8, рис. 5



```
(tools) PS C:\Users\maks\doc\lab2.11> flake8
(tools) PS C:\Users\maks\doc\lab2.11> _
```

Рисунок 4 – Проверка заданий

Контрольные вопросы:

1. Что такое замыкание?

Замыкание 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?

Каждый раз, когда мы вызываем функцию, у нее создаются локальные переменные, а после завершения – уничтожаются. Можно сделать так, чтобы после завершения работы функции, часть локальных переменных не уничтожалась, а сохраняла свои значения до следующего запуска.

Локальная переменная не будет уничтожена, если на нее где-то останется “живая” ссылка, после завершения работы функции. Эту ссылку может сохранять вложенная функция. Функции, построенные по такому принципу могут использоваться для построения специализированных функций, т.е. являются фабриками функций. Замыкания как раз и использую эту идею.

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?

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?

Вызывая функцию, мы фактически обращаемся к вложенной функции. Переменная внешней функции, является в ней локальной и имеет область enclosing во вложенной функции. Несмотря на то, что внешняя функция завершает свою работу, переменная 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 – это свойство позволяет строить иерархические структуры данных.

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