**ИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

## **«Замыкания в языке Python»**

**Отчет по лабораторной работе № 2.11**

**по дисциплине «Основы программной инженерии»**

|  |
| --- |
| Выполнил студент группы ПИЖ-б-о-21-1 |
| Коновалова В.Н. « » 2022г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил Воронкин Р.А. \_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Ставрополь 2022

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

Ход работы:

1. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.

3. Выполните клонирование созданного репозитория.

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

6. Создайте проект PyCharm в папке репозитория.

7. Проработайте пример лабораторной работы.

Код:

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
def fun1(a):  
 x = a \* 3  
  
 def fun2(b):  
 nonlocal x  
 return b + x  
  
 return fun2  
  
  
test\_fun = fun1(4)  
print(test\_fun(7))

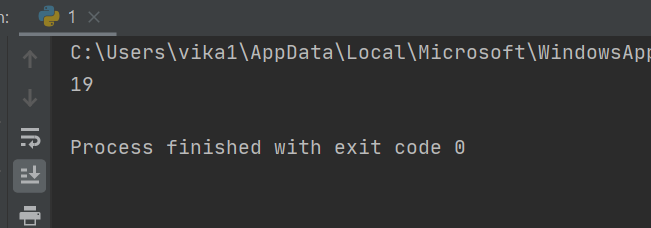
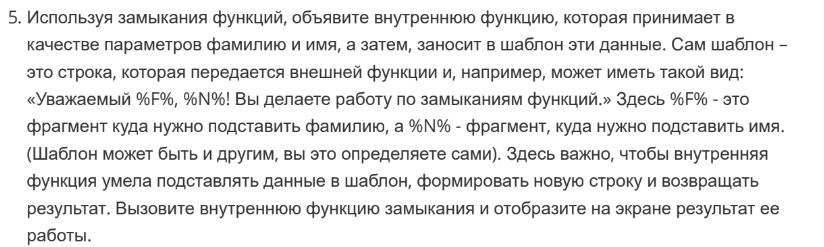


Рисунок 6 – Результат работы программы

8. Решите индивидуальное задание согласно своему варианту.



#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
*"""  
Используя замыкания функций, объявите внутреннюю функцию, которая принимает в  
качестве параметров фамилию и имя, а затем, заносит в шаблон эти данные. Сам шаблон –  
это строка, которая передается внешней функции и, например, может иметь такой вид:  
«Уважаемый %F%, %N%! Вы делаете работу по замыканиям функций.» Здесь %F% - это  
фрагмент куда нужно подставить фамилию, а %N% - фрагмент, куда нужно подставить имя.  
(Шаблон может быть и другим, вы это определяете сами). Здесь важно, чтобы внутренняя  
функция умела подставлять данные в шаблон, формировать новую строку и возвращать  
результат. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее  
работы.  
"""*def sample(string):  
 def name\_surname(n, s):  
 sample\_data = string.replace("%N%", n)  
 sample\_data = sample\_data.replace("%F%", s)  
 return sample\_data  
  
 return name\_surname  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 sample\_string = (  
 "Уважаемый(ая) %F% %N%! Вы делаете работу по замыканиям функций."  
 )  
 name, surname = input("Введите имя и фамилию: ").split()  
 print(sample(sample\_string)(name, surname))

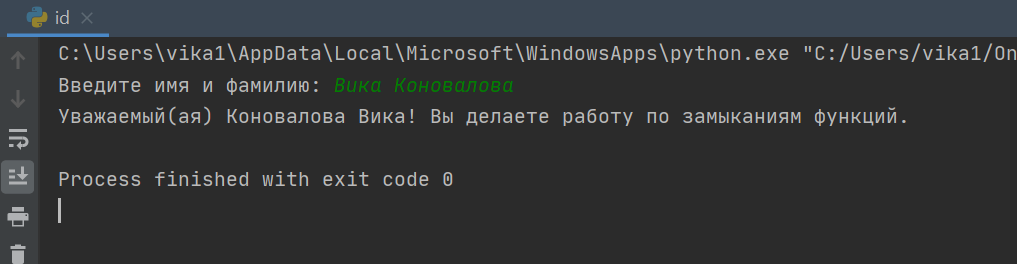


Рисунок 9 – Результат работы программы

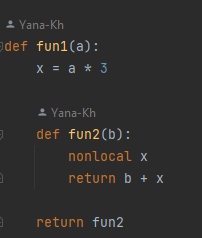
12. Зафиксируйте сделанные изменения в репозитории.

Вопросы для защиты работы

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 её параметрами.

1. Как реализованы замыкания в языке программирования Python?



1. Что подразумевает под собой область видимости Local?

Эту область видимости имеют переменные, которые создаются и используются внутри функций.

1. 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 её вложенной функции находится в enclosing области видимости.

1. Что подразумевает под собой область видимости Global?

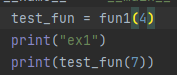
Переменные области видимости global – это глобальные переменные уровня модуля (модуль – это файл с расширением .py)

1. Что подразумевает под собой область видимости Build-in?

Эти сущности доступны в любом модуле Python и не требуют предварительного импорта. Built-in – это максимально широкая область видимости.

1. Как использовать замыкания в языке программирования Python?

В случае с реализацией выше:





8. Как замыкания могут быть использованы для построения иерархических данных?

