

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»

Выполнил:
Евдаков Евгений Владимирович
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 _____

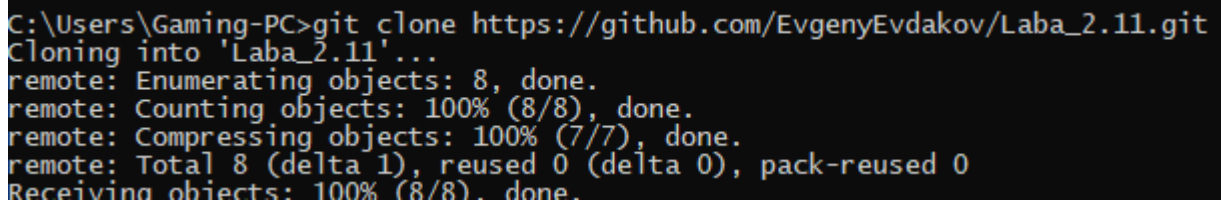
Ставрополь, 2023 г.

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:

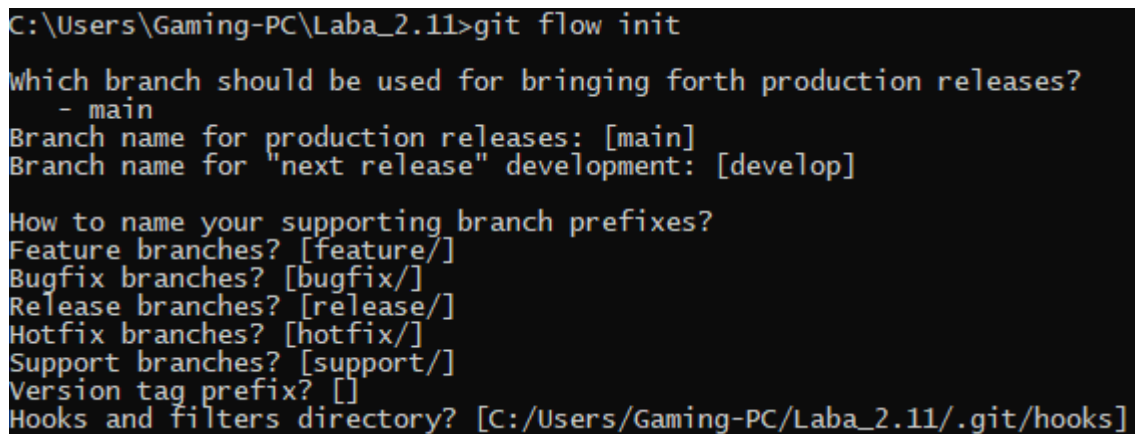
Задание 1. Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер.



```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/Laba_2.11.git
Cloning into 'Laba_2.11'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
```

Рисунок 1. Клонирование репозитория

Задание 2. Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.



```
C:\Users\Gaming-PC\Laba_2.11>git flow init
which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Gaming-PC/Laba_2.11/.git/hooks]
```

Рисунок 2. Модель ветвления git-flow

Задание 3. Создал проект PyCharm в папке репозитория. Приступил к работе с примером 1. Добавил новый файл primer1.py.

Условие примера: функция mul() умножает два числа и возвращает полученный результат. Если мы ходим на базе нее решить задачу: “умножить число на пять”, то в самом простом случае, можно вызывать mul(), передавая в качестве первого аргумента пятерку.

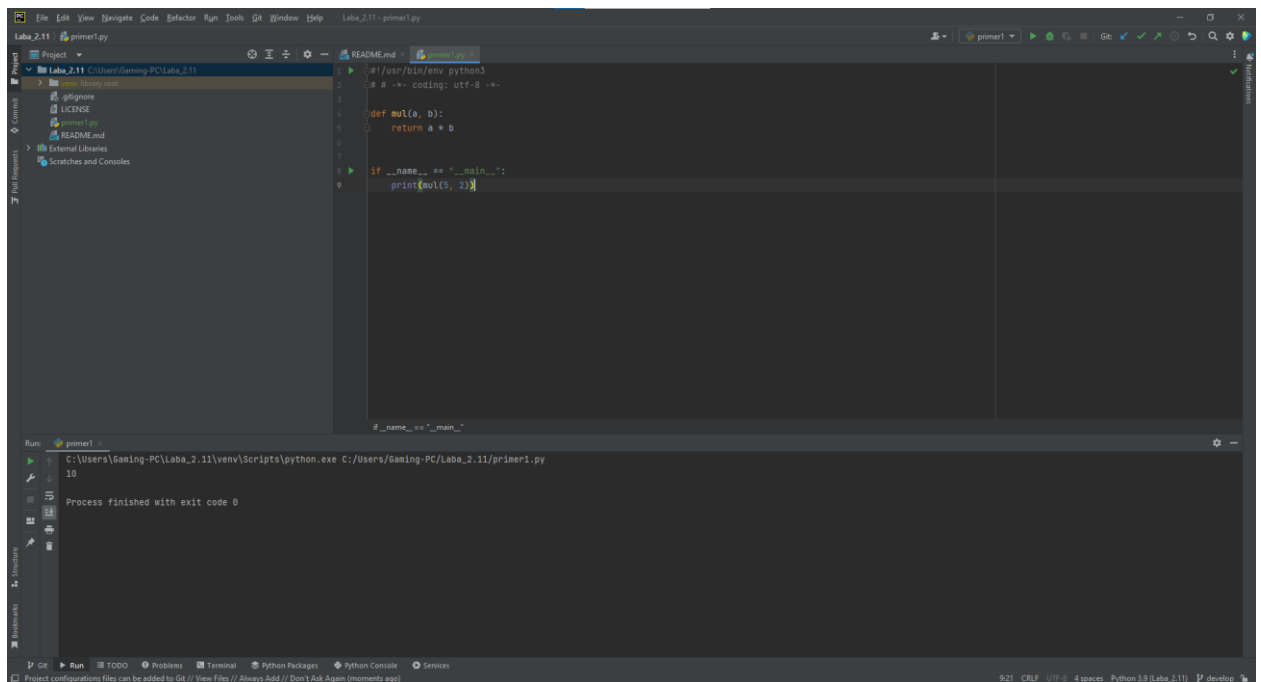


Рисунок 3. Реализация примера 1

Задание 4. Создал новый файл под названием `primer2.py`. Приступил к работе с примером №2.

Условие примера: на самом деле мы можем создать новую функцию, которая будет вызывать `mul()`, с пятеркой и ещё одним числом, которое она будет получать в качестве своего единственного аргумента.

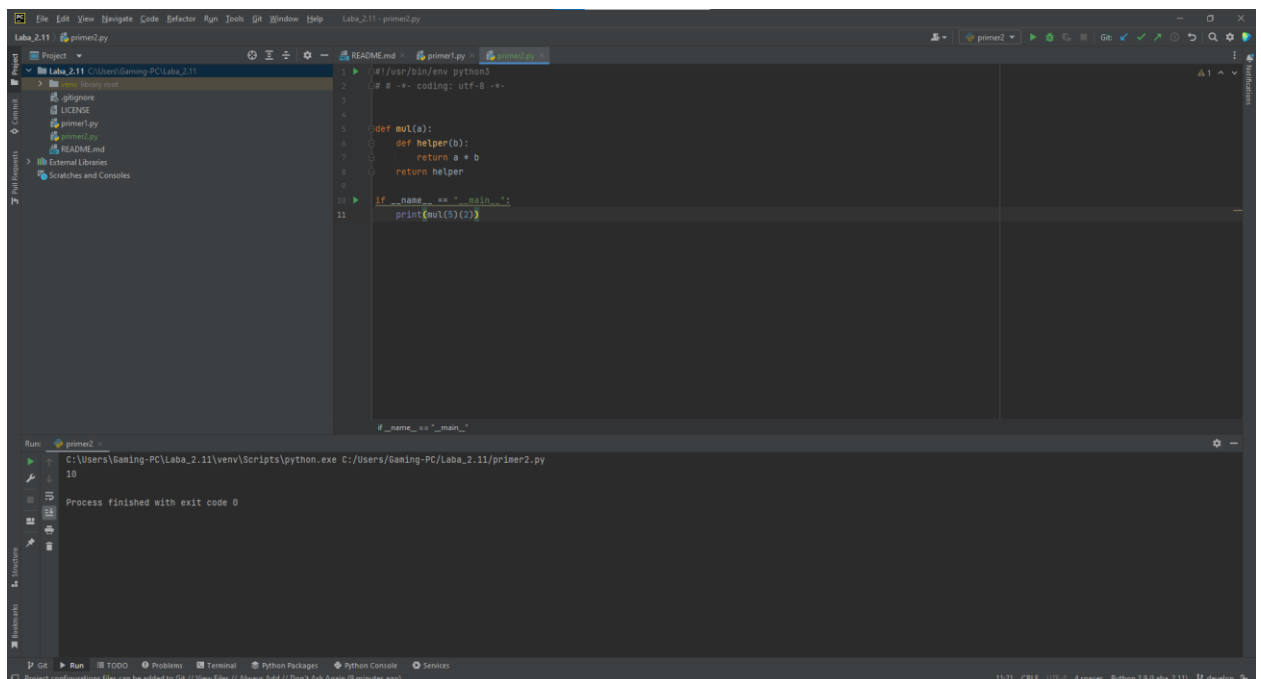


Рисунок 4. Реализация примера 2

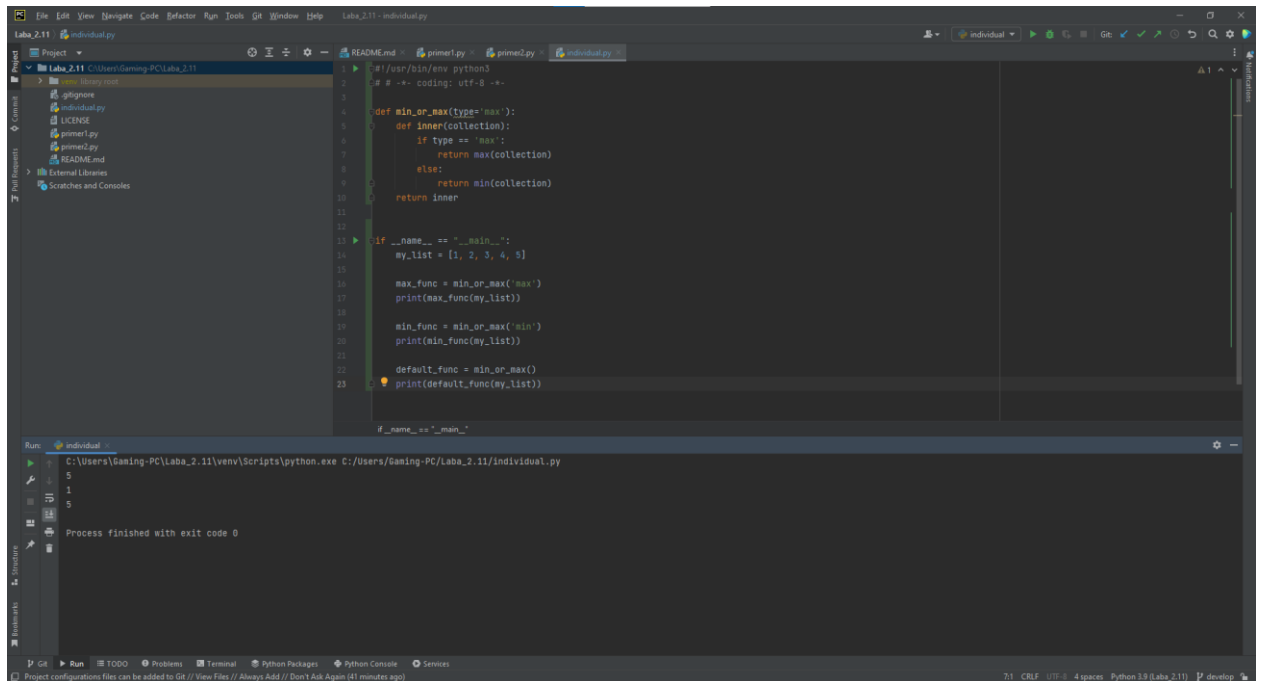
Задание 5.

Индивидуальное задание

Вариант 9 (по списку группы)

Создал новый файл под названием individual.py.

Условие задания: Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента коллекцию (список или кортеж) и возвращает или минимальное значение, или максимальное, в зависимости от значения параметра type внешней функции. Если type равен «max», то возвращается максимальное значение, иначе – минимальное. По умолчанию type должно принимать значение «max». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def min_or_max(type='max'):
5     def inner(collection):
6         if type == 'max':
7             return max(collection)
8         else:
9             return min(collection)
10    return inner
11
12
13 if __name__ == '__main__':
14     my_list = [1, 2, 3, 4, 5]
15
16     max_func = min_or_max('max')
17     print(max_func(my_list))
18
19     min_func = min_or_max('min')
20     print(min_func(my_list))
21
22     default_func = min_or_max()
23     print(default_func(my_list))
24
25 if __name__ == '__main__':
```

Run individual

C:\Users\Gaming-PC\Laba_2.11\venv\Scripts\python.exe C:/Users/Gaming-PC/Laba_2.11/individual.py

5
1
5

Process finished with exit code 0

Рисунок 5. Программа индивидуального задания

Задание 6.

После выполнения работы на ветке develop, слил ее с веткой main и отправил изменения на удаленный сервер.

```

C:\Users\Gaming-PC\Laba_2.11>git merge develop
Updating b60a878..1f7b04e
Fast-forward
 individual.py | 23 ++++++
 primer1.py    | 9 +++++
 primer2.py    | 11 +++++
 3 files changed, 43 insertions(+)
 create mode 100644 individual.py
 create mode 100644 primer1.py
 create mode 100644 primer2.py

```

Рисунок 10. Слияние ветки develop и main

Ссылка: https://github.com/EvgenyEvdakov/Laba_2.11

Ответы на контрольные вопросы:

1. Что такое замыкание?

Замыкание - это функция, которая имеет доступ к переменным из внешней функции, в которой она была определена, даже после того, как внешняя функция завершила свое выполнение. Это позволяет создавать более гибкие и мощные функции в Python.

2. Как реализованы замыкания в языке программирования Python?

В языке Python замыкания реализуются путем определения внутренней функции внутри внешней функции, и возвращения этой внутренней функции в качестве результата выполнения внешней функции. Это позволяет сохранять контекст вызова, чтобы его можно было использовать при последующих вызовах внутренней функции.

3. Что подразумевает под собой область видимости Local?

Область видимости Local - это область видимости, ограниченная внутри текущей функции, где переменные могут быть определены и использованы только внутри этой функции.

4. Что подразумевает под собой область видимости Enclosing?

Область видимости Enclosing - это область видимости, в которой определена текущая функция, а также область видимости любых внешних функций, в которых эта функция была определена, но не в глобальной области видимости.

5. Что подразумевает под собой область видимости Global?

Область видимости Global предполагает, что переменная объявлена за пределами функции или класса и доступна везде.

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

Область видимости Build-in - это область видимости, содержащая функции и объекты, встроенные в Python, которые могут быть использованы в любом месте в программе без импорта. В ней находятся все стандартные имена модулей и функций, например: `print ()`, `len ()`, `str ()`, `list ()`.

7. Как использовать замыкания в языке программирования Python?

Для использования замыканий в Python нужно определить внутреннюю функцию внутри внешней функции и вернуть эту внутреннюю функцию в качестве результата выполнения внешней функции. Затем можно вызвать возвращенную функцию, и она будет иметь доступ к переменным из внешней функции, даже после того, как внешняя функция завершила свое выполнение.

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

Замыкания могут быть использованы для построения иерархических данных в языке программирования Python, например, для создания древовидных структур данных, таких как деревья решений. Один из способов использования замыканий для построения иерархических данных - это создание функций-генераторов.

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