

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

«Замыкания в Python»

**Отчет по лабораторной работе № 2.11
по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Сумин Никита Сергеевич.

«31 » октября 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.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

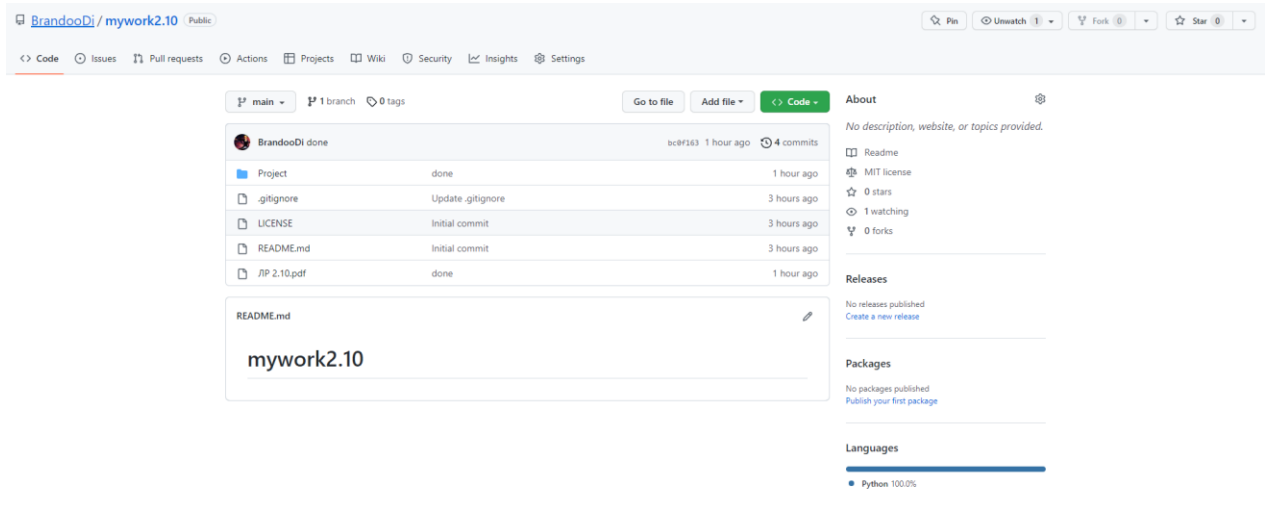


Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

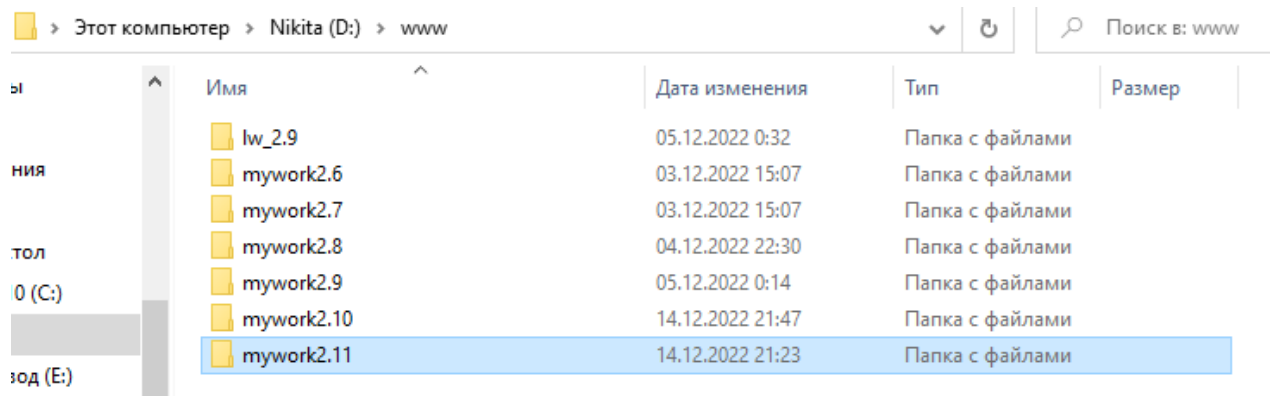


Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

Создайте проект PyCharm в папке репозитория.

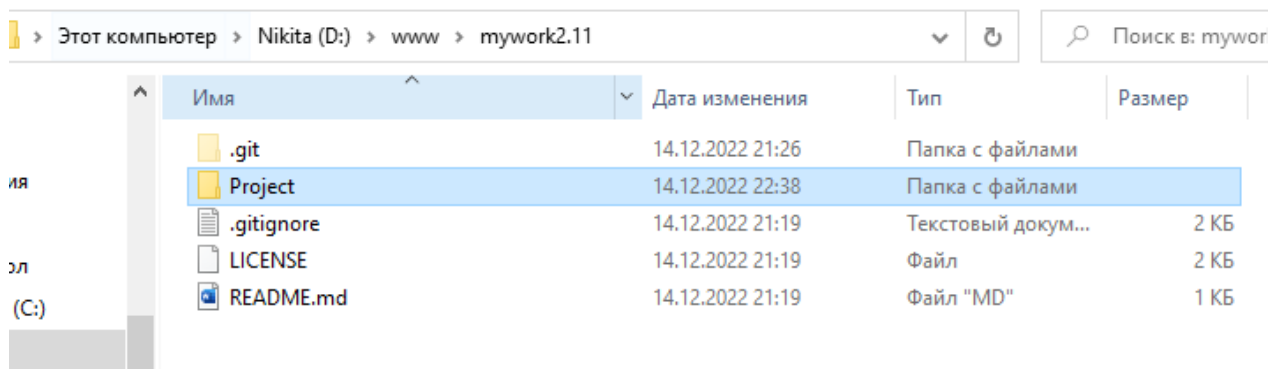


Рисунок 4 - Проект PyCharm

5. Проработайте примеры лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

Функция `mul()` умножает два числа и возвращает полученный результат.

```
D:\www\mywork2.11\Project\venv\Scripts\python.exe D:/www/mywork2.11/Project/pr1.py
10

Process finished with exit code 0
```

Рисунок 5 - Результат выполнения примера 1

В функции `fun1()` объявлена локальная переменная `x`, значение которой определяется аргументом `a`. В функции `fun2()` используются эта же переменная `x`, `nonlocal` указывает на то, что эта переменная не является локальной, следовательно, ее значение будет взято из ближайшей области видимости, в которой существует переменная с таким же именем. В нашем случае – это область `enclosing`, в которой этой переменной `x` присваивается значение `a * 3`. Также как и в предыдущем случае, на переменную `x` после вызова `fun1(4)`, сохраняется ссылка, поэтому она не уничтожается.

```
D:\www\mywork2.11\Project\venv\Scripts\python.exe D:/www/mywork2.11/Project/pr2.py
19

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения примера 2

6. Решите индивидуальное задание. (Вариант 19/9)

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.

```
D:\www\mywork2.11\Project\venv\Scripts\python.exe D:/www/mywork2.11/Project/ind1.py
65
1

Process finished with exit code 0
```

Рисунок 7 - 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?

Замыкания в Python реализованы посредством вложенных 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?

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 `Built-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 использовать замыкания в языке программирования Python?

Мы имеем дело с замыканием в Python, когда вложенная функция ссылается на значение из локальной области видимости объемлющей функции.

Критерии, которые должны быть выполнены для создания замыкания в Python, изложены в следующих пунктах:

1. У нас должна быть вложенная функция (функция внутри функции).
2. Вложенная функция должна ссылаться на значение, определенное в объемлющей функции.
3. Объемлющая функция должна возвращать вложенную функцию.

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