

**МИНИСТЕРСТВО 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

Кучеренко С. Ю. « » 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.

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.

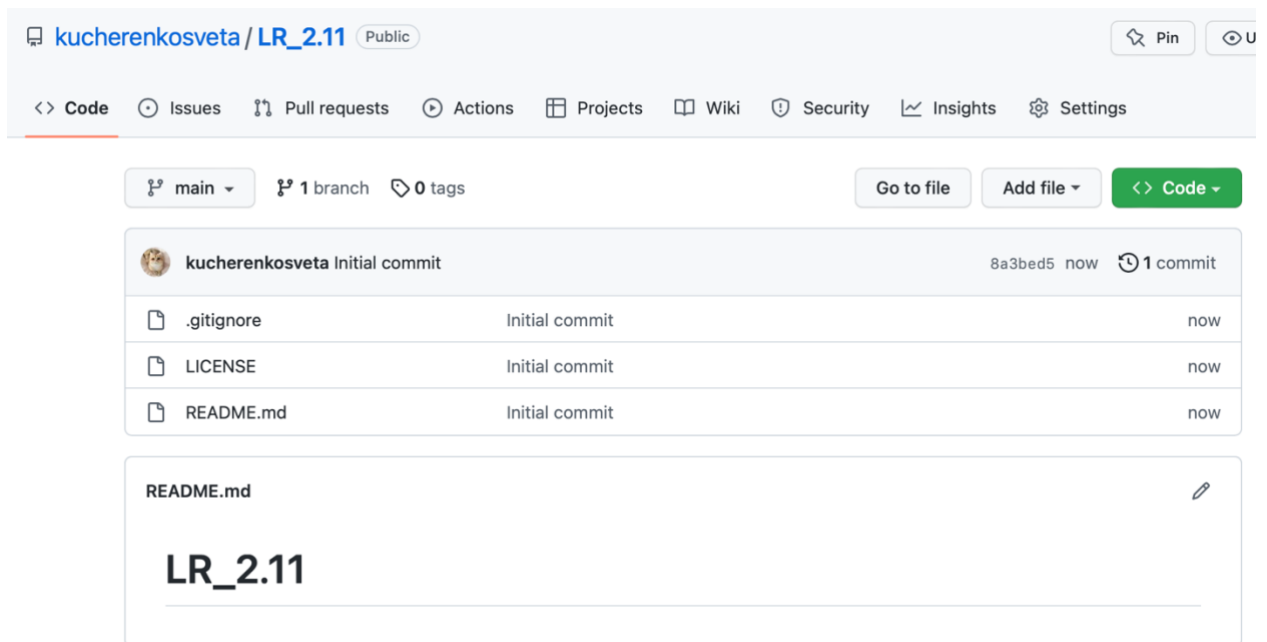


Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
[(base) svetik@MacBook-Air-Svetik Laba14 % git clone https://github.com/kucherenkosveta/LR_2.11.git
Cloning into 'LR_2.11'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```

[(base) svetik@MacBook-Air-Svetik LR_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? [/Users/svetik/Desktop/Laba14/LR_2.11/.git/hooks]
(base) svetik@MacBook-Air-Svetik LR_2.11 % █

```

Рисунок 3 – Организация репозитория в соответствии с моделью git-flow

6. Создайте проект PyCharm в папке репозитория.

7. Проработайте пример лабораторной работы.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def fun1(a):
    x = a * 3

    def fun2(b):
        nonlocal x
        return b + x

    return fun2

if __name__ == '__main__':
    test_fun = fun1(4)
    print(test_fun(7))

```

19

Process finished with exit code 0

Рисунок 4 – Результат работы программы

8. Решите индивидуальное задание согласно своему варианту.

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.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def max_min(type='max'):
    def helper(collection):
        if type == 'max':
            return max(collection)
        else:
            return min(collection)
    return helper

if __name__ == '__main__':
    print(max_min(input("Введите max или min: "))
          (map(int, input("Введите список: ").split(' '))))
```

```
Введите max или min: max
Введите список: 1 2 7 33 -5 0 5
33
```

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.

2. Как реализованы замыкания в языке программирования Python?

```
new *  
def fun2(b):  
    nonlocal x  
    return b + x  
  
return fun2
```

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 её 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?

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?

В случае с реализацией выше:

```
test_fun = fun1(4)
print("ex1")
print(test_fun(7))
```

19

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

```
tpl = lambda d, e: (d, e)
s = tpl(1, 2)
print(s)
f = tpl(3, s)
print(f)
c = tpl(s, f)
print(c)
```

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