

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего
образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

**Кафедра
инфокоммуникаций
Институт цифрового
развития**

**ОТЧЁТ
по лабораторной работе №14
Дисциплина: «Основы программной инженерии»
Тема: «Замыкания в языке Python»**

Выполнил: студент 2
курса группы Пиж-б-о-
21-1
Рязанцев Матвей
Денисович

Ставрополь 2022

Цель работы: приобретение навыков по работе с функциями с переменным числом

Выполнение работы

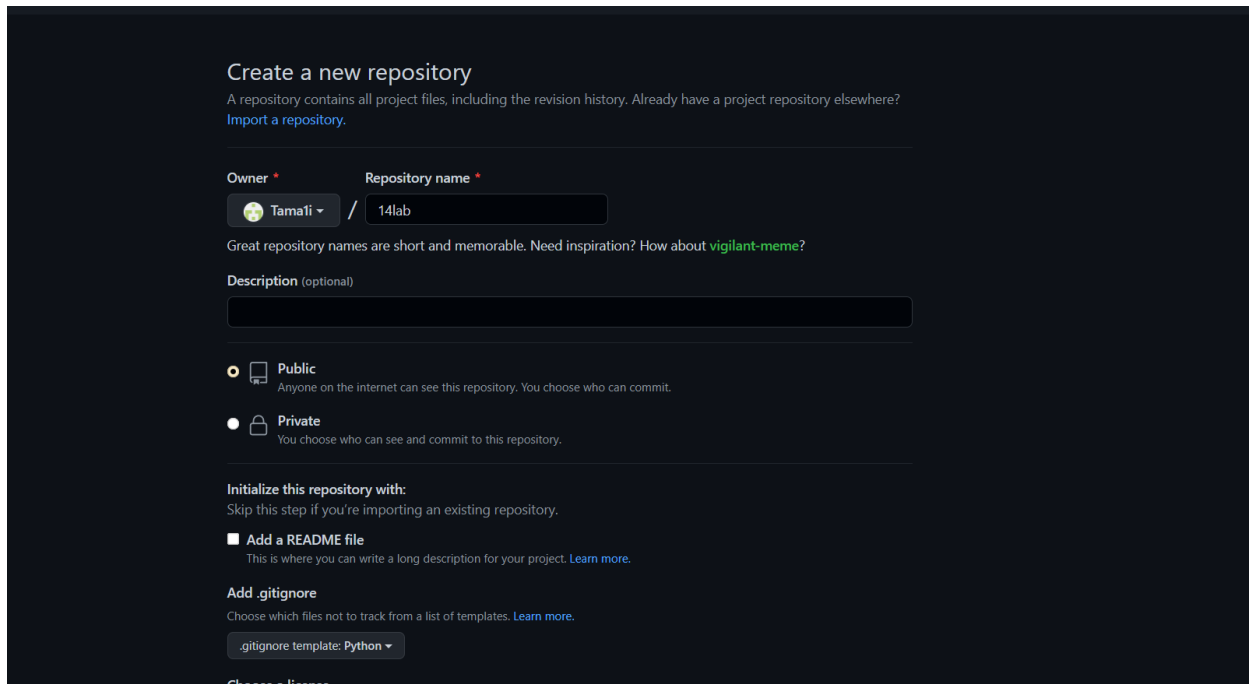


Рисунок 1 -создание репозитория

```
D:\gite\14lab>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? [D:/gite/14lab/.git/hooks]

D:\gite\14lab>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Рисунок 2 - Организация репозитория по модели ветвления git flow

Код общего задания:

```
#!/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("ex1")
    print(test_fun(7))

    print("\nex2")
    tpl = lambda d, e: (d, e)
    s = tpl(1, 2)
    print(s)
    f = tpl(3, s)
    print(f)
    c = tpl(s, f)
    print(c)

```

Индивидуальное задание

Код программы:

```

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

def f():
    def g(x):
        return x + 3

    return g

k = int(input())
cnt = f()
print(cnt(k))

```

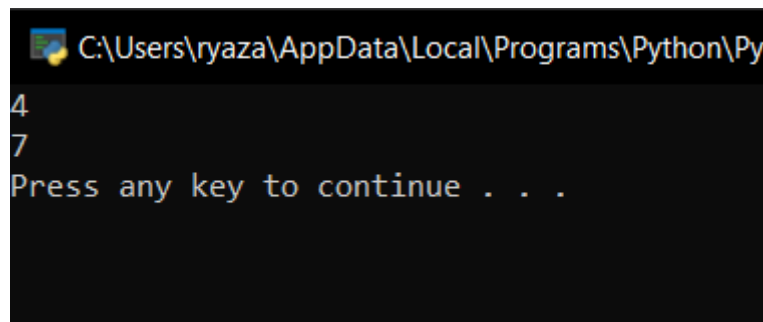


Рисунок 3 – результат работы программы

Контрольные вопросы

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?

```
def fun1 (a):  
    x = a * 3  
    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))
```

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)
```