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 №14
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

_____
(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: замыкания в языке Python.

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

Ход выполнения работы

1. Создал репозиторий.

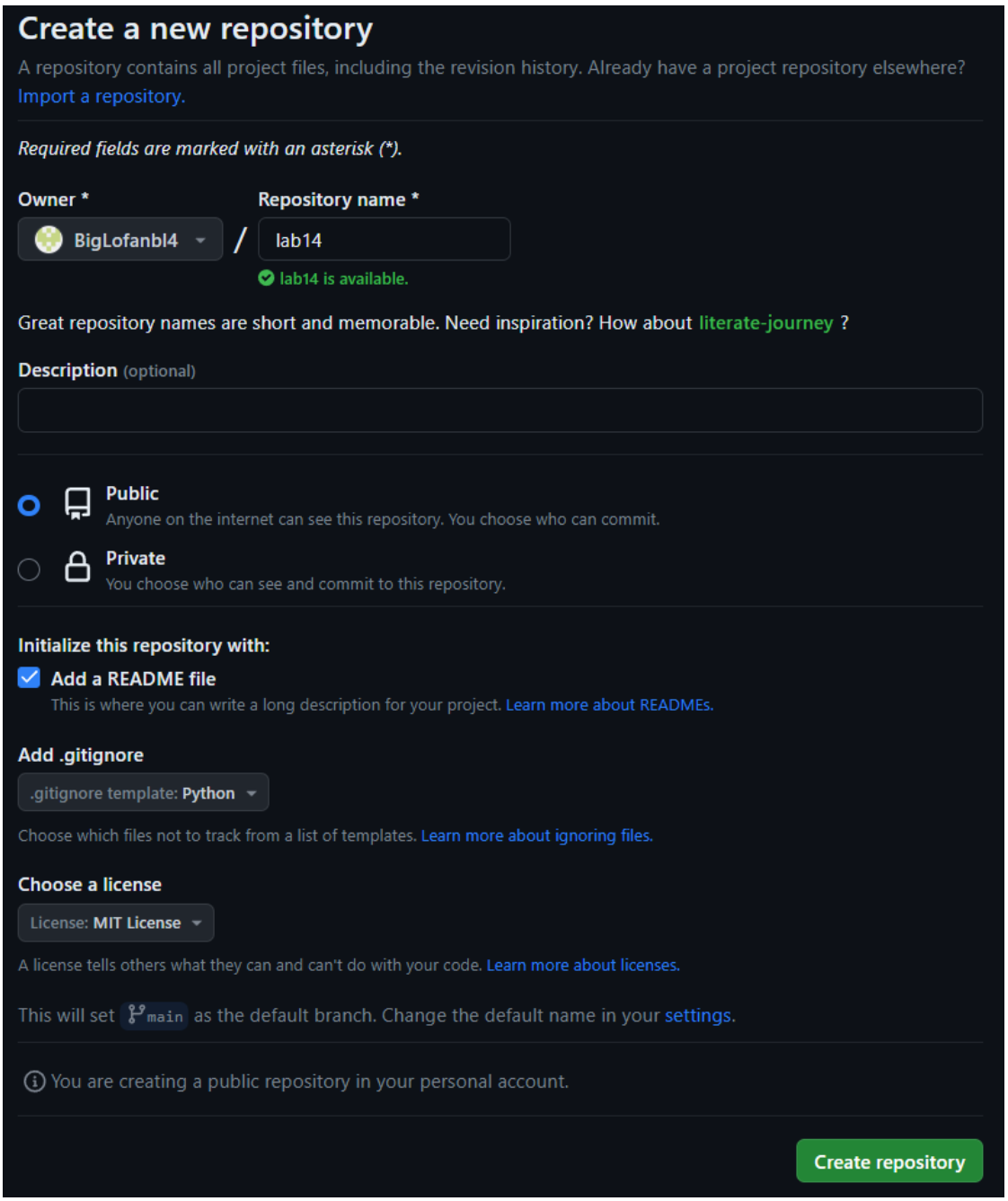

Рисунок 1.1 – Создание репозитория

2. Проработал примеры из лабораторной работы.

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

def mul(a):
    def helper(b):
        return a * b
    return helper

def fun1(a):
    x = a * 3
    def fun2(b):
        nonlocal x
        return b + x
    return fun2



if __name__ == "__main__":
    print(mul(5)(2))
    new_mul5 = mul(5)
    print(new_mul5(3))

    test_fun = fun1(4)
    print(test_fun(7))
    print(fun1(2)(4)) # x = 6 => 10
```

Рисунок 2.1 – Код программы

```
10
15
19
10
```

Рисунок 2.2 – Вывод программы

3. Индивидуальное задание. Вариант 14.

Используя замыкания функций, объявите внутреннюю функцию,
которая из переданного ей списка строк формирует многострочную строку вида:

```
<ol>
<li>строка_1</li>
…
<li>строка_N</li>
</ol>
```

и возвращает ее. Где строка*1, строка*2, ... - это строки из переданного функции списка.
Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

Рисунок 3.1 – Условие задания

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


def func():
    def form_str(lst):
        res = "<ol>\n"
        for i in lst:
            res += f"<li>{i}</li>\n"
        res += "</ol>"
        return res

    return form_str


# def func(lst):
#     def form_str():
#         res = "<ol>\n"
#         for i in lst:
#             res += f"<li>{i}</li>\n"
#         res += "</ol>"
#         return res
#
#     return form_str


if __name__ == "__main__":
    lst = ["str1", "str2", "str3"]
    a = func()
    print(a(lst))
    # print(func()(lst))
```

Рисунок 3.2 – Код программы

```
<ol>
<li>str1</li>
<li>str2</li>
<li>str3</li>
</ol>
```

Рисунок 3.3 – Вывод программы

Ответы на контрольные вопросы

1. Замыкание – это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела это функции в окружающем коде и не являющиеся ее параметрами.

2. Замыкания в Python реализованы с помощью локальных функций. Локальные функции могут использовать переменные из внешней функции.

3. Область видимости Local – это область видимости внутри функции. Эту область видимости имеют переменные, которые создаются и используются внутри функции.

4. Enclosing область видимости, которая определяется внутри вложенной функции, т.е. Encolsing область видимости подразумевает, что переменная находится во внешней функции.

5. Переменные области видимости global – это глобальные переменные уровня модуля.

6. Уровень Python интерпретатора. В рамках этой области видимости находятся функции open, len и т. п., также туда входят исключения. Built-in – это максимально широкая область видимости.

7. Для инкапсуляции данных, построения иерархических структур, создание внутренних структур данных.

8. Операция комбинирования объектов данных обладает свойством замыкания в том случае, если результаты соединения объектов с помощью этой операции сами могут соединяться этой же операцией