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 программной инженерии»

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

_____
(подпись)

Преподаватель:
Воронкин Р.А., канд. тех. наук, доцент,
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.

Результаты выполнения

1. Создали репозиторий с лицензией MIT, добавили в .gitignore необходимые правила для работы с IDE PyCharm, клонировали репозиторий, организовали репозиторий в соответствии с моделью git-flow.

```
Last login: Tue Oct 24 20:29:02 on ttys000
itssyoma@MacBook-Air-Sema Основы программной инженерии % git clone https://github.com/itssyoma/megarepo_21.git
Cloning into 'megarepo_21'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
itssyoma@MacBook-Air-Sema Основы программной инженерии % cd megarepo_21
itssyoma@MacBook-Air-Sema megarepo_21 % git checkout -b develop
Switched to a new branch 'develop'
itssyoma@MacBook-Air-Sema megarepo_21 % git branch release
itssyoma@MacBook-Air-Sema megarepo_21 % git branch develop
fatal: a branch named 'develop' already exists
itssyoma@MacBook-Air-Sema megarepo_21 % git branch hotfix
itssyoma@MacBook-Air-Sema megarepo_21 % git branch feature
itssyoma@MacBook-Air-Sema megarepo_21 %
```

Рисунок 1 – Работа с репозиторием в командной строке.

2. Проработали пример.

example.py > ...

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


def add_two(a):
    x = 2
    return a + x


def add_four(a):
    x = 2

    def add_some():
        print(f'Вывод внутри вложенной функции add_some(): {str(x)}')
        return a + x

    return add_some()


x = 4


def fun():
    print(f'Вывод при вызове функции fun(): {x + 3}')


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


if __name__ == '__main__':
    print(f'Вывод функции add_two(4): {add_two(4)}')
    print(f'Вывод функции add_four(5): {add_four(5)}')
    fun()
    test_fun = fun1(4)
    print(f'Вывод замыкания: {test_fun(7)}')

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_211 % /usr/local/bin/python3 "/Users/itssyrepo_211/example.py"
Вывод функции add_two(4): 6
Вывод внутри вложенной функции add_some(): 2
Вывод функции add_four(5): 7
Вывод при вызове функции fun(): 7
Вывод замыкания: 19
```

Рисунок 2 – Код и результат работы программы.

3. Выполнили индивидуальное задание (вариант 1). Используя замыкания функций, определите вложенную функцию, которая бы увеличивала значение переданного параметра на 3 и возвращала бы вычисленный результат. Вызовите внешнюю функцию для получения ссылки на внутреннюю функцию и присвойте ее переменной с именем cnt. Затем, вызовите внутреннюю функцию через переменную cnt со значением k, введенным с клавиатуры.

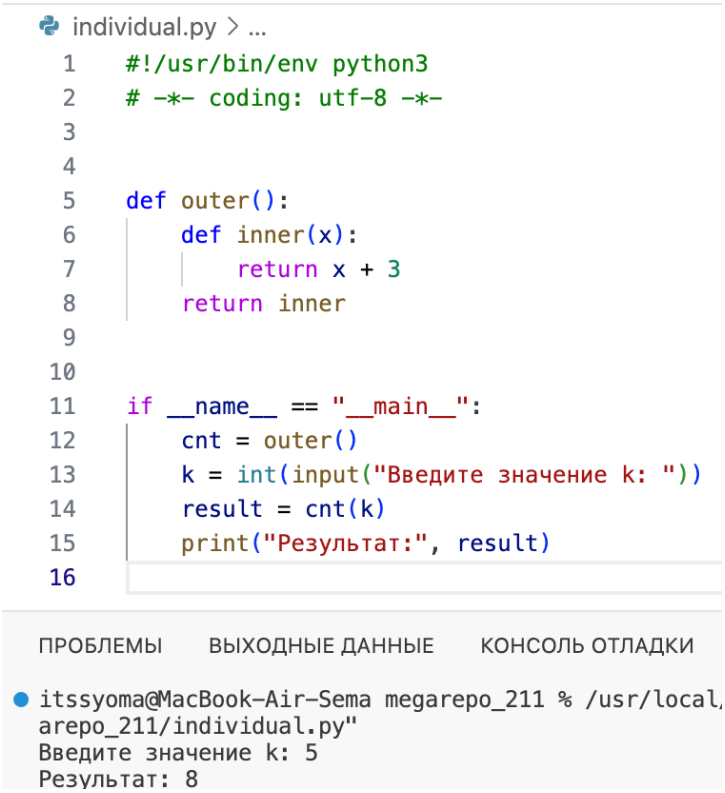

Рисунок 3 – Код и результат работы программы.

Ответы на контрольные вопросы

1. Что такое замыкание?

Замыкание - это функция, которая запоминает окружение, в котором она была создана, и имеет доступ к переменным из этого окружения.

2. Как реализованы замыкания в языке программирования Python?

В Python замыкания реализуются путем вложения функций, то есть определения одной функции внутри другой. Внутренняя функция имеет доступ к переменным внешней функции, что позволяет создавать замыкания.

3. Что подразумевает под собой область видимости Local?

Область видимости Local относится к переменным, определенным внутри текущей функции. Эти переменные видны только внутри этой функции.

4. Что подразумевает под собой область видимости Enclosing?

Область видимости Enclosing относится к переменным, определенным во внешних функциях, в которых текущая функция была определена. Эти переменные также доступны в текущей функции.

5. Что подразумевает под собой область видимости Global?

Область видимости Global относится к переменным, определенным на уровне модуля или глобальной области видимости. Эти переменные доступны из любого места в программе.

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

Область видимости Build-in относится к встроенным функциям и переменным, предоставляемым интерпретатором Python.

7. Как использовать замыкания в языке программирования Python?

Для использования замыканий в Python нужно определить внутреннюю функцию внутри внешней функции и вернуть ее как результат выполнения внешней функции. После этого можно вызывать возвращенную внутреннюю функцию для доступа к переменным из окружения внешней функции.

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

Замыкания могут быть использованы для построения иерархических данных, таких как деревья или связанные списки. Вложенные функции могут быть использованы для представления узлов или элементов данных, а замыкания позволяют сохранять состояние и связи между этими элементами.