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 «Программирование на Python»
Вариант 31

Выполнил:
Репкин Александр Павлович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Замыкания в языке Python.

Цель: приобрести навыки работы с замыканиями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Выполнены примеры из лабораторной работы в них рассматривались область видимости переменных (В результатах получена ошибка из-за недоступности переменной), а также замыкания в Python.

```
Traceback (most recent call last):
  File "J:\Python_Vuz\Программы 14\Examples.py", line 38, in <module>
    print(x)
NameError: name 'x' is not defined
5
x = 2
7
10
```

Рисунок 1. Полученный результат выполнения примеров.

2. Выполнено индивидуальное задание. В нём требовалось, используя замыкания функций, определить вложенную функцию, увеличивающую значение переданного ей параметра на 3. Вычисленный результат возвращается. Задание также требовало присвоение переменной cnt ссылки на внутреннюю функцию при помощи вызова внешней функции. Затем, через переменную cnt вызывается внутренняя функция со значением k, введенным с клавиатуры.

```
Good day! Please, enter value: 6
According to our calculations, answer is -  9
```

Рисунок 2. Полученный результат индивидуального задания.

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

"""Внутренняя функция увеличивает полученное значение на 3."""


def calculaion():

    def plus_3(value):
        return value + 3
    return plus_3


if __name__ == '__main__':
    cnt = calculaion()
    k = int(input("Good day! Please, enter value: "))
    print("According to our calculations, answer is - ", cnt(k))

```

Рисунок 3. Полученный код индивидуального задания.

3. Ответы на вопросы.

1) Что такое замыкание?

Ответ: замыкание – функция, у которой есть доступ к области видимости, сформированной внешней по отношению к ней функции даже после того, как эта внешняя функция завершила работу. Это значит, что в замыкании могут храниться переменные, объявленные во внешней функции и переданные ей аргументы.

2) Как реализованы замыкания в Python?

Ответ: в Python вложенные функции имеют доступ к переменным внешней функции даже после того, как внешняя функция прекратила свою работу.

3) Что подразумевает под собой область видимости Local?

Ответ: область видимости Local имеют переменные, которые создаются и используются внутри функций, снаружи доступ к ним невозможен.

4) Что подразумевает под собой область видимости Enclosing?

Ответ: внутри функции могут быть вложенные функции и локальные переменные. Эта локальная переменная функции для её вложенной функции находится в enclosing области видимости.

5) Что подразумевает под собой область видимости Global?

Ответ: переменные области видимости global – это глобальные переменные уровня модуля (.py файла).

6) Что подразумевает под собой область видимости Built-In?

Ответ: Built-in – уровень Python интерпретатора, максимально широкая область видимости. 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.

7) Как использовать замыкания в языке программирования Python?

Ответ: в качестве примера использования замыканий можно привести код, в котором создаётся переменная, хранящая ссылку на внешнюю функцию с заданным значением (5):

```
>>> def mul(a):
        def helper(b):
           return a * b
        return helper
>>> new_mul5 = mul(5)
```

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

Ответ: в качестве примера можно привести код, в котором относительно операции объединения tpl, оказались замкнуты кортежи:

```
>>> tpl = lambda a, b: (a, b)
>>> b = tpl(3, a)
>>> b
(3, (1, 2))
>>> c = tpl(a, b)
>>> c
((1, 2), (3, (1, 2)))
```

Вывод: в ходе выполнения лабораторной работы были приобретены навыки работы замыканиями при написании программ с помощью языка программирования Python версии 3.x.