

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.11
дисциплины «Программирование на Python»

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

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

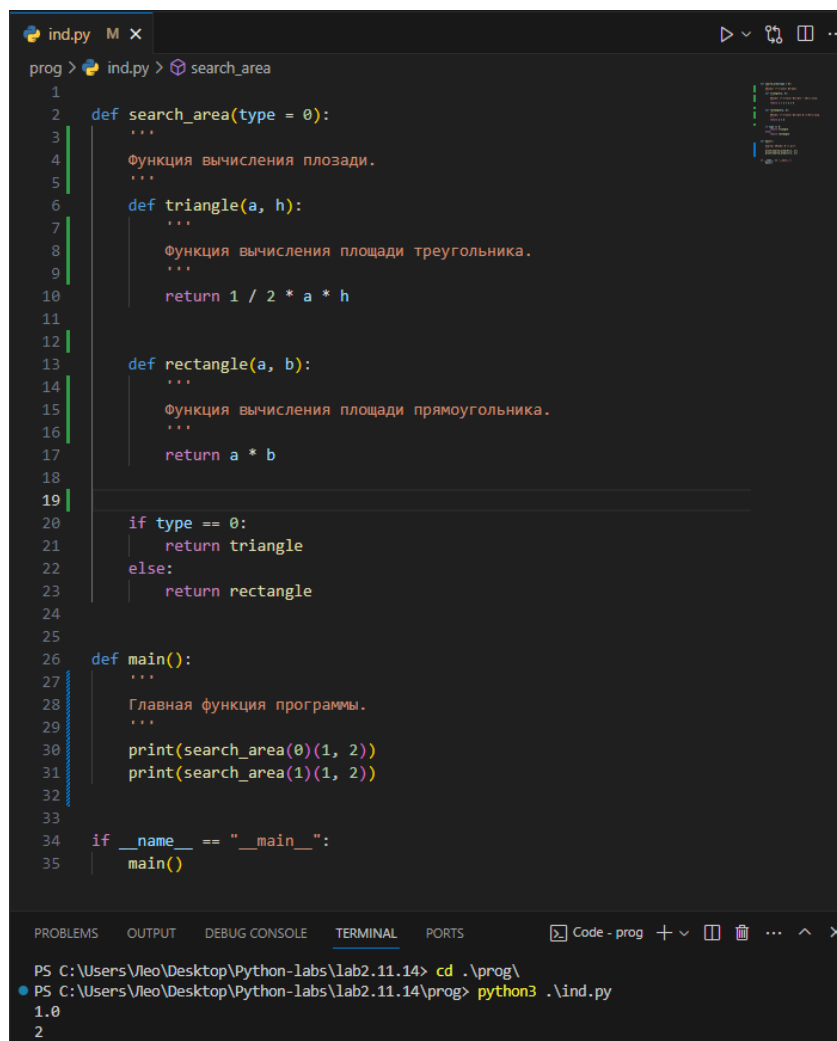
Ставрополь, 2023 г.

Тема: Замыкания в языке Python

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

Порядок выполнения работы:

Индивидуальное задание (Вариант 7): Используя замыкания функций, объявите внутреннюю функцию, которая на основе двух параметров вычисляет площадь фигуры. Какой именно фигуры: треугольника или прямоугольника, определяется параметром `type` внешней функции. Если `type` принимает значение 0, то вычисляется площадь треугольника, а иначе – прямоугольника. По умолчанию параметр `type` должен быть равен 0. Вычисленное значение должно возвращаться внутренней функцией. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.



```
1
2 def search_area(type = 0):
3     """
4     Функция вычисления площади.
5     """
6     def triangle(a, h):
7         """
8         Функция вычисления площади треугольника.
9         """
10        return 1 / 2 * a * h
11
12    def rectangle(a, b):
13        """
14        Функция вычисления площади прямоугольника.
15        """
16        return a * b
17
18
19    if type == 0:
20        return triangle
21    else:
22        return rectangle
23
24
25
26 def main():
27     """
28     Главная функция программы.
29     """
30    print(search_area(0)(1, 2))
31    print(search_area(1)(1, 2))
32
33
34 if __name__ == "__main__":
35    main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Ileo\Desktop\Python-labs\lab2.11.14> cd .\prog\
PS C:\Users\Ileo\Desktop\Python-labs\lab2.11.14\prog> python3 .\ind.py
1.0
2

Рисунок 1 – Результат выполнения программы ind.py

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

Контрольные вопросы

1. Что такое замыкание?

Функция, которая запоминает окружение, в котором она была создана, и может получать доступ к переменным из этого окружения, даже после того, как оно было вызвано и покинуло свою область видимости.

2. Как реализованы замыкания в языке программирования Python?

В языке программирования Python замыкания реализованы путем вложения функций, то есть функция внутри другой функции. Внутренняя функция имеет доступ к переменным внешней функции, что позволяет создавать и использовать замыкания.

3. Что подразумевает под собой область видимости Local?

Означает, что переменная определена внутри функции и доступна только внутри этой функции. Это означает, что переменная имеет локальную область видимости

4. Что подразумевает под собой область видимости Enclosing?

Означает, что переменная определена во внешней функции и доступна внутри вложенной функции. Это означает, что переменная имеет область видимости, охватывающую внешнюю функцию и все вложенные функции.

5. Что подразумевает под собой область видимости Global?

Означает, что переменная определена в основной программе или модуле и доступна в любом месте программы. Это означает, что переменная имеет глобальную область видимости.

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

означает, что переменная является встроенной в язык Python и доступна в любом месте программы. Это означает, что переменная имеет встроенную область видимости.

7. Как использовать замыкания в языке программирования Python?

Для использования замыканий в Python необходимо определить внутреннюю функцию внутри внешней функции и вернуть эту внутреннюю функцию в качестве результата внешней функции. Это позволит внутренней функции сохранить доступ к переменным внешней функции даже после завершения работы внешней функции

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

Замыкания могут быть использованы для построения иерархических данных путем создания функций, которые возвращают другие функции или объекты данных. Это позволяет создавать иерархию функций или объектов, которые могут иметь доступ к переменным и состоянию своих родительских функций или объектов. Такой подход позволяет эффективно управлять иерархическими структурами данных в программировании.