

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 №11
дисциплины «Программирование на Python»

Выполнил:
Горбунов Данила Евгеньевич
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. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

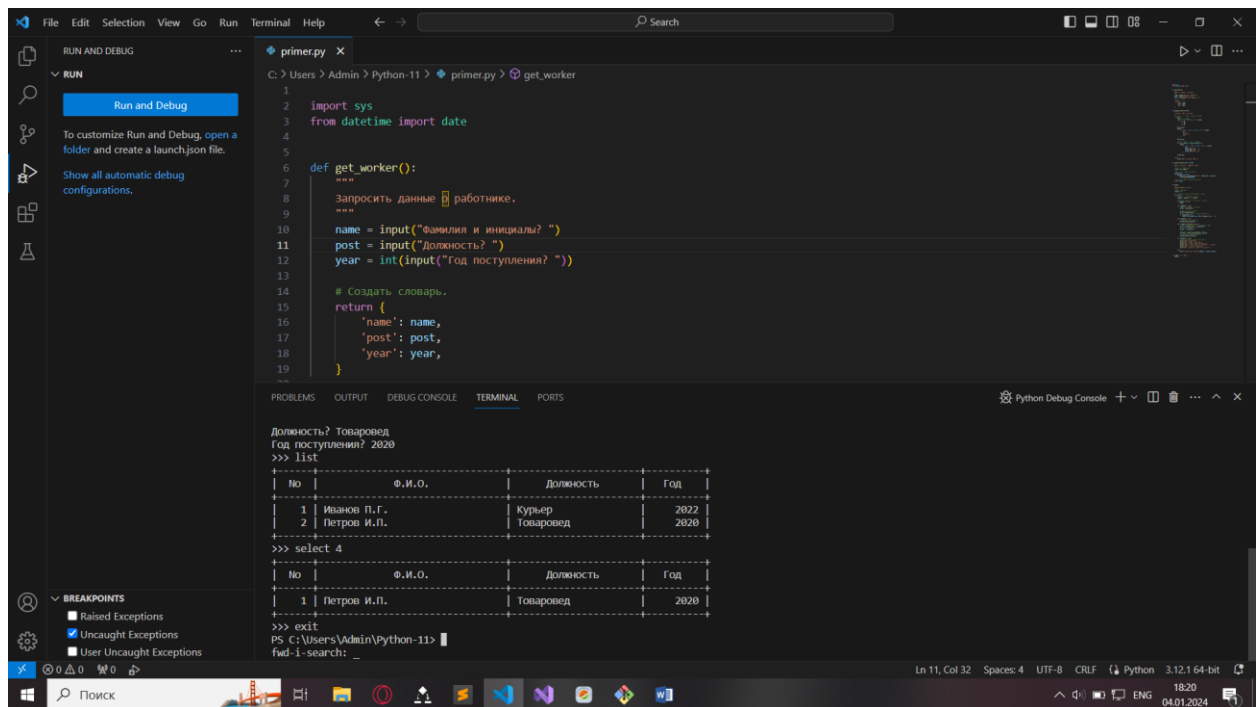
Ставрополь, 2023 г.

Тема: Работа с функциями в языке Python

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

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера.



The screenshot shows a Python IDE with a file named `primer.py`. The code defines a function `get_worker()` that prompts the user for an employee's name, position, and year of entry, and returns a dictionary with this information. The terminal output shows the function being called twice, resulting in two tables of employee data.

```
1
2 import sys
3 from datetime import date
4
5
6 def get_worker():
7     """
8     Запросить данные о работнике.
9     """
10    name = input("Фамилия и инициалы? ")
11    post = input("Должность? ")
12    year = int(input("Год поступления? "))
13
14    # Создать словарь.
15    return {
16        'name': name,
17        'post': post,
18        'year': year,
19    }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Terminal Output:

```
Должность? Товаровед
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Иванов П.П. | Курьер | 2022 |
| 2 | Петров И.И. | Товаровед | 2020 |
+-----+-----+-----+-----+
>>> select 4
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Петров И.И. | Товаровед | 2020 |
+-----+-----+-----+-----+
>>> exit
PS C:\Users\Admin\Python-11>
fwd-i-search: _
```

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции `test()` и инструкции `if __name__ == '__main__':`. В ней запрашивается на ввод целое

число. Если оно положительное, то вызывается функция `positive()`, тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция `negative()`, ее тело содержит выражение вывода на экран слова "Отрицательное".

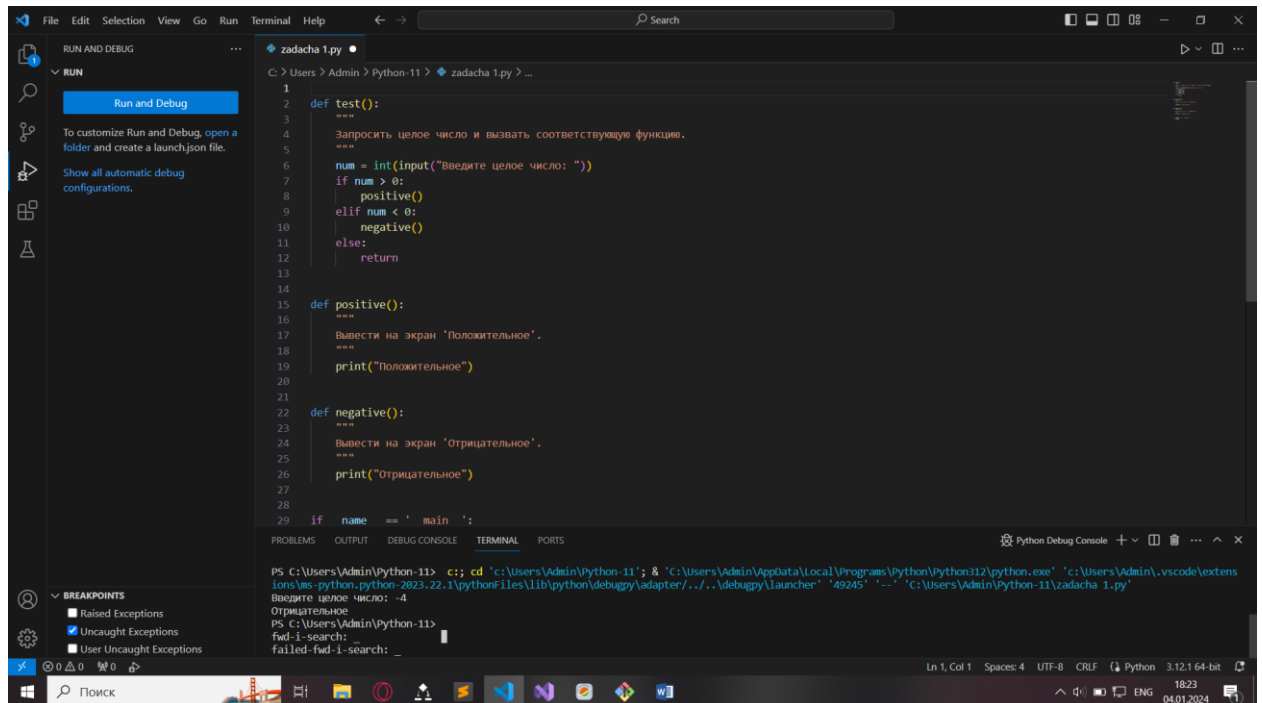


Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: в основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле $S = \pi r^2$. В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле $S_{\text{бок}} = 2\pi r h$, или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.

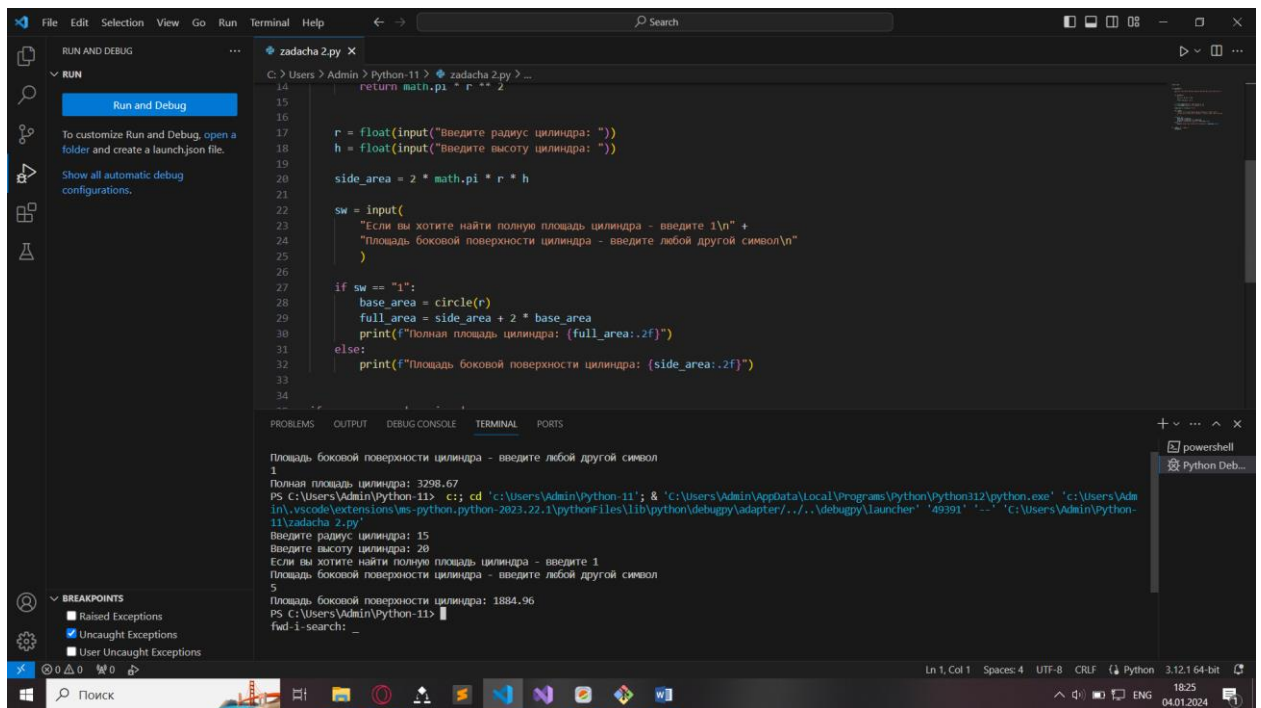


Рисунок 3. Результат работы программы из задачи 2

7. Решил следующую задачу: напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.

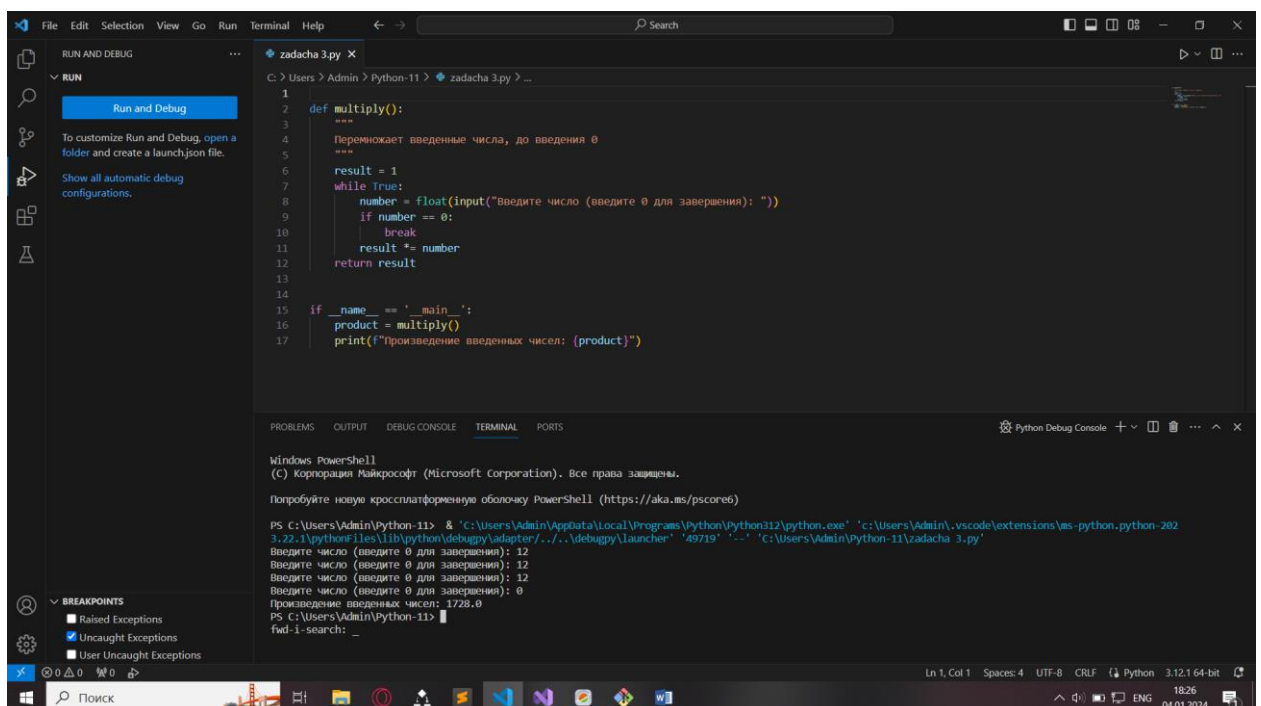


Рисунок 4. Результат работы программы из задачи 3

8. Решил следующую задачу: напишите программу, в которой определены следующие четыре функции:

- 1) Функция `get_input()` не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку;
- 2) Функция `test_input()` имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое `True`. Если нельзя – `False`;
- 3) Функция `str_to_int()` имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число;
- 4) Функция `print_int()` имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

В основной ветке программы вызовите первую функцию. То, что она вернула, передайте повторную функцию. Если вторая функция вернула `True`, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.

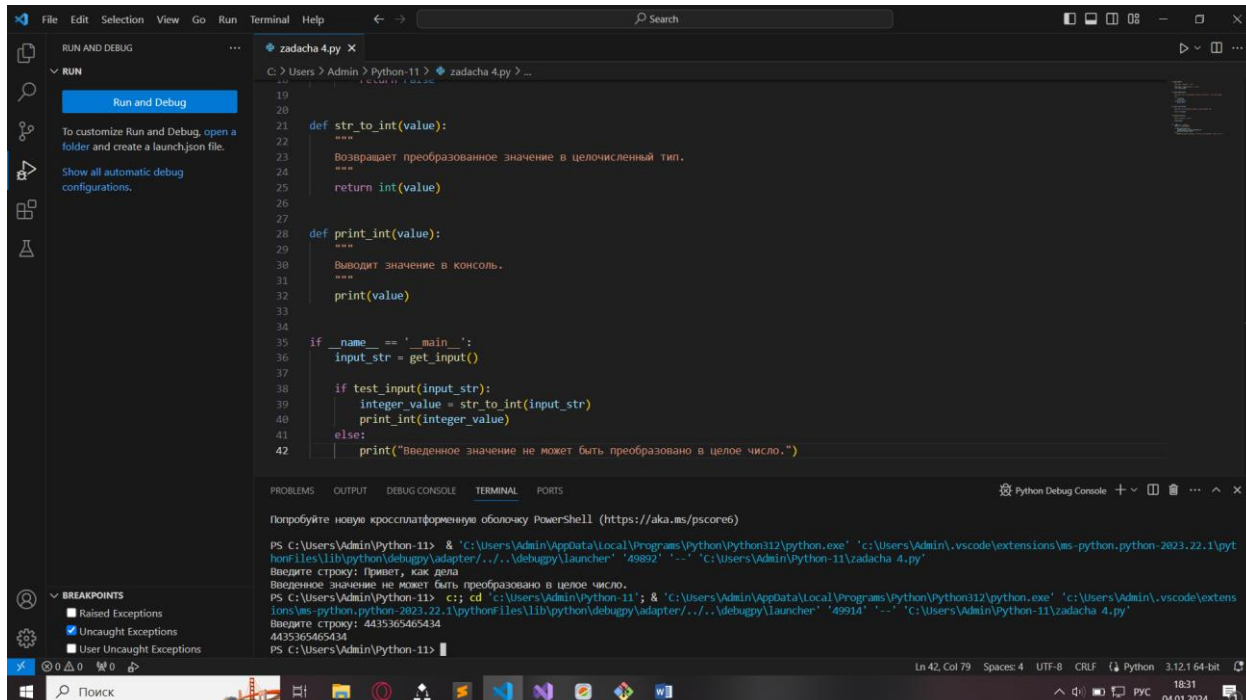


Рисунок 5. Результат работы программы из задачи 4

7. Выполнил индивидуальное задание. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде

отдельной функции.

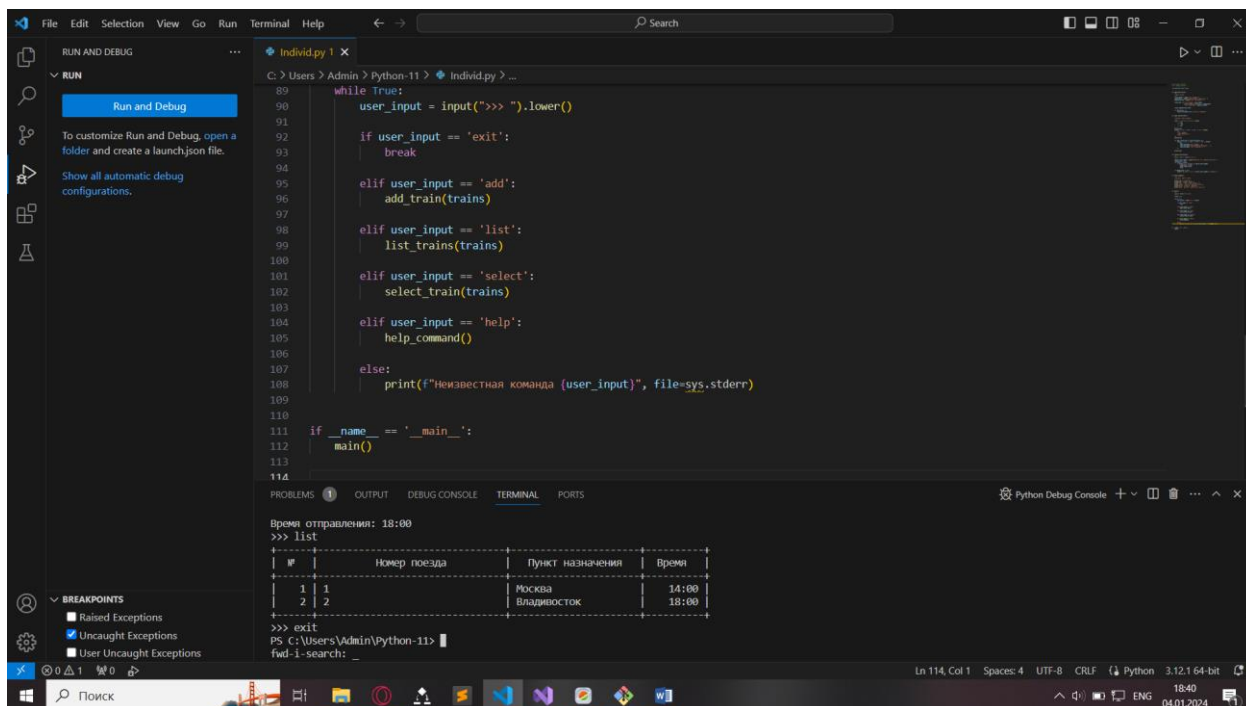


Рисунок 6. Результат работы программы из индивидуального задания

Контрольные вопросы

1. Каково назначение функций в языке программирования Python?

Функции в языке программирования Python используются для группировки кода, чтобы он мог быть многократно использован, делая программу более читаемой и легко управляемой. Функции также позволяют разделить большие программы на более мелкие, что упрощает разработку и управление кодом.

2. Каково назначение операторов def и return ?

Оператор def используется для определения функции в Python. Он указывает интерпретатору, что следующий блок кода является телом функции. Оператор return используется для возврата значения из функции.

Когда интерпретатор Python достигает оператора return, он возвращает указанное значение и завершает выполнение функции.

3. Каково назначение локальных и глобальных переменных при написании функций в Python?

Локальные переменные объявляются внутри функции и доступны

только внутри этой функции. Глобальные переменные объявляются вне функций и доступны во всем коде программы. При написании функций в Python, локальные переменные используются для временного хранения данных, в то время как глобальные переменные могут быть использованы в различных частях программы.

4. Как вернуть несколько значений из функции Python?

Для возврата нескольких значений из функции в Python используется механизм кортежей. Функция может вернуть кортеж, содержащий несколько значений, и затем эти значения могут быть присвоены различным переменным при вызове функции.

5. Какие существуют способы передачи значений в функцию?

Значения могут быть переданы в функцию в Python через позиционные аргументы, ключевые аргументы и аргументы по умолчанию.

Позиционные аргументы передаются в порядке, в котором они определены в сигнатуре функции. Ключевые аргументы передаются с указанием имени параметра. Аргументы по умолчанию имеют значения по умолчанию и могут быть пропущены при вызове функции.

6. Как задать значение аргументов функции по умолчанию?

Для задания значения аргументов функции по умолчанию в Python используется синтаксис "переменная=значение" в сигнатуре функции. Если значение не передается при вызове функции, будет использовано значение по умолчанию.

7. Каково назначение lambda-выражений в языке Python?

Lambda-выражения в Python представляют собой анонимные функции, которые могут содержать только одно выражение. Они обычно используются в ситуациях, когда требуется небольшая функция в одном месте кода.

8. Как осуществляется документирование кода согласно PEP257?

Документирование кода в Python согласно PEP257 включает в себя

использование строк документации (docstrings) для описания модулей, функций, классов и методов. Строки документации должны быть заключены в тройные кавычки и предоставлять информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.

9. В чем особенность однострочных и многострочных форм строк документации?

Однострочные строки документации начинаются и заканчиваются тройными кавычками и предназначены для краткого описания модулей, функций, классов или методов. Они обычно используются для кратких пояснений и описаний.

Многострочные строки документации также начинаются и заканчиваются тройными кавычками, но могут занимать несколько строк и предоставлять более подробное описание. Они обычно используются для более полного и подробного документирования, включая информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.

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