

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 №2.8**  
**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:** Работа с функциями в языке Python.

**Цель:** приобретение навыков по работе с функциями 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.

```
example1.py > get_worker
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from datetime import date
6
7
8  def get_worker():
9      """
10     Запросить данные о работнике.
11     """
12     name = input("Фамилия и инициалы? ")
13     post = input("Должность? ")
14     year = int(input("Год поступления? "))
15
16     # Создать словарь.
17     return {
18         'name': name,
19         'post': post,
20         'year': year,
21     }
22
23
24  def display_workers(staff):
25      """
26     Отобразить список работников.
27     """
28     # Проверить, что список работников не пуст.
29     if staff:
30         # Заголовок таблицы.
31         line = '4--{}--{}--{}--{}--'.format(
32             '-' * 4,
33             '-' * 30,
34             '-' * 20,
35             '-' * 8
36         )
37         print(line)
38         print(
39             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
40                 "No",
41                 "Ф. И. О.",
42                 "Должность",
43                 "Год"
44             )
45         )
46         print(line)
47
48     # Вывести данные о всех сотрудниках.
49     for idx, worker in enumerate(staff, 1):
50         print(
51             '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
52                 idx,
53                 worker.get('name', ''),
```

Рисунок 2 – Код программы (часть 1)

```

example1.py > get_worker
54         worker.get('post', ''),
55         worker.get('year', 0)
56     )
57 )
58 print(line)
59
60 else:
61     print("Список работников пуст.")
62
63
64 def select_workers(staff, period):
65     """
66     Выбрать работников с заданным стажем.
67     """
68     # Получить текущую дату.
69     today = date.today()
70
71     # Сформировать список работников.
72     result = []
73     for employee in staff:
74         if today.year - employee.get('year', today.year) >= period:
75             result.append(employee)
76
77     # Возвратить список выбранных работников.
78     return result
79
80
81 def main():
82     """
83     Главная функция программы.
84     """
85     # Список работников.
86     workers = []
87     # Организовать бесконечный цикл запроса команд.
88     while True:
89         # Запросить команду из терминала.
90         command = input(">>> ").lower()
91
92         # Выполнить действие в соответствие с командой.
93         if command == 'exit':
94             break
95
96         elif command == 'add':
97             # Запросить данные о работнике.
98             worker = get_worker()
99
100             # Добавить словарь в список.
101             workers.append(worker)
102             # Отсортировать список в случае необходимости.
103             if len(workers) > 1:
104                 workers.sort(key=lambda item: item.get('name', ''))
105
106         elif command == 'list':

```

Рисунок 3 – Код программы (часть 2)

```

107         # Отобразить всех работников.
108         display_workers(workers)
109
110     elif command.startswith('select '):
111         # Разбить команду на части для выделения стажа.
112         parts = command.split(' ', maxsplit=1)
113         # Получить требуемый стаж.
114         period = int(parts[1])
115
116         # Выбрать работников с заданным стажем.
117         selected = select_workers(workers, period)
118         # Отобразить выбранных работников.
119         display_workers(selected)
120
121     elif command == 'help':
122         # Вывести справку о работе с программой.
123         print("Список команд:\n")
124         print("add - добавить работника;")
125         print("list - вывести список работников;")
126         print("select <стаж> - запросить работников со стажем;")
127         print("help - отобразить справку;")
128         print("exit - завершить работу с программой.")
129
130     else:
131         print(f"Неизвестная команда {command}", file=sys.stderr)
132
133
134 if __name__ == '__main__':
135     main()
136

```

Рисунок 4 – Код программы (часть 3)

```

itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/megarepo_28/example1.py"
>>> help
Список команд:

add – добавить работника;
list – вывести список работников;
select <стаж> – запросить работников со стажем;
help – отобразить справку;
exit – завершить работу с программой.
>>> add
Фамилия и инициалы? Баратов С.Г.
Должность? Дизайнер
Год поступления? 2023
>>> add
Фамилия и инициалы? Васин Д.Г
Должность? Копирайтер
Год поступления? 2022
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Баратов С.Г.             |      Дизайнер       |      2023     |
|  2 | Васин Д.Г                |      Копирайтер     |      2022     |
+-----+-----+-----+-----+
>>> select 1
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Васин Д.Г                |      Копирайтер     |      2022     |
+-----+-----+-----+-----+
>>> select 0
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Баратов С.Г.             |      Дизайнер       |      2023     |
|  2 | Васин Д.Г                |      Копирайтер     |      2022     |
+-----+-----+-----+-----+
>>> exit
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/megarepo_28/example1.py"
>>> hello
Неизвестная команда hello
>>>

```

Рисунок 5 – Результат выполнения программы.

3. Выполнили задание №1. 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".

```
task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def positive():
6      print("Положительное")
7
8
9  def negative():
10     print("Отрицательное")
11
12
13 def test():
14     num = int(input("Введите целое число: "))
15     if num > 0:
16         positive()
17     elif num < 0:
18         negative()
19
20
21 if __name__ == "__main__":
22     test()
23
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    Т

```
/usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.lo
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
еба/Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: 8
Положительное
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: -1
Отрицательное
```

Рисунок 6 – Код и результат выполнения программы.

При смене позиций функций `positive()`, `negative()` и `test()` программа будет функционировать так же. в языке Python функции определяются и сохраняются в памяти при выполнении программы. При вызове функции интерпретатор Python просто находит ее в памяти и выполняет код, который она содержит. Для проверки поменяем функции местами:

```
task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def test():
6      num = int(input("Введите целое число: "))
7      if num > 0:
8          positive()
9      elif num < 0:
10         negative()
11
12
13 def negative():
14     print("Отрицательное")
15
16
17 def positive():
18     print("Положительное")
19
20
21 if __name__ == "__main__":
22     test()
23
```

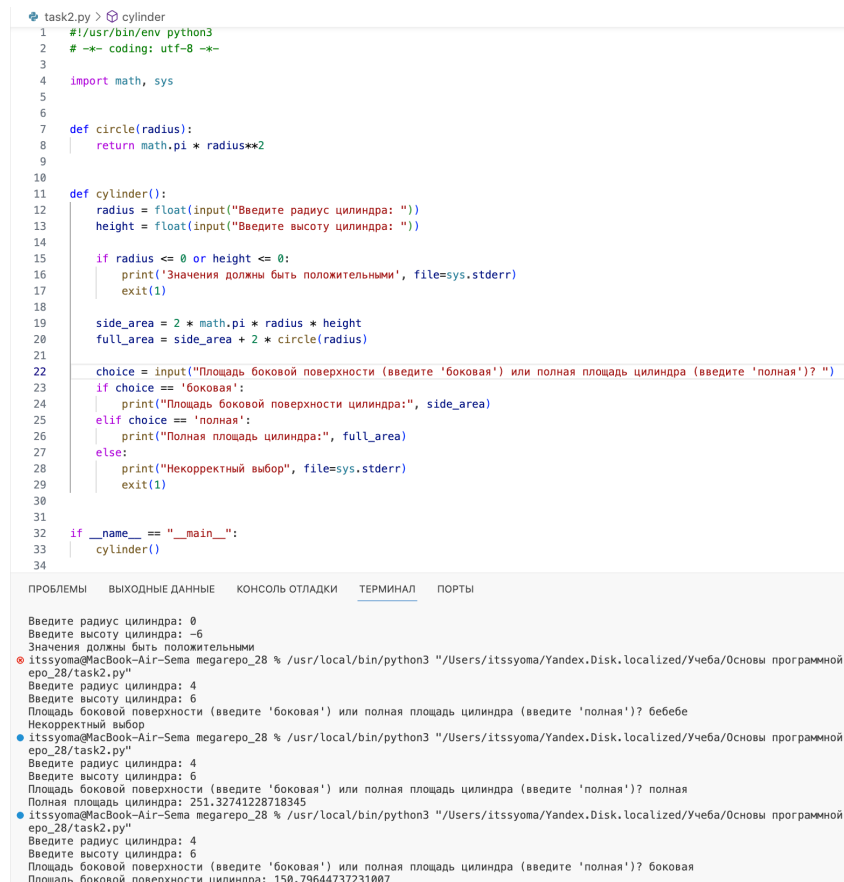
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    Т

```
/usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.lo
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
еба/Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: 8
Положительное
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: -1
Отрицательное
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
еба/Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: 8
Положительное
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin
еба/Основы программной инженерии/megarepo_28/task1.py"
Введите целое число: -1
Отрицательное
```

Рисунок 7 – Код и результат выполнения программы.

Результат работы программы не изменился.

4. Выполнили задание №2. В основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле . В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.



```

task2.py > cylinder
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math, sys
5
6
7  def circle(radius):
8      return math.pi * radius**2
9
10
11 def cylinder():
12     radius = float(input("Введите радиус цилиндра: "))
13     height = float(input("Введите высоту цилиндра: "))
14
15     if radius <= 0 or height <= 0:
16         print("Значения должны быть положительными", file=sys.stderr)
17         exit(1)
18
19     side_area = 2 * math.pi * radius * height
20     full_area = side_area + 2 * circle(radius)
21
22     choice = input("Площадь боковой поверхности (введите 'боковая') или полная площадь цилиндра (введите 'полная')? ")
23     if choice == 'боковая':
24         print("Площадь боковой поверхности цилиндра:", side_area)
25     elif choice == 'полная':
26         print("Полная площадь цилиндра:", full_area)
27     else:
28         print("Некорректный выбор", file=sys.stderr)
29         exit(1)
30
31
32 if __name__ == "__main__":
33     cylinder()
34

```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

```

Введите радиус цилиндра: 0
Введите высоту цилиндра: -6
Значения должны быть положительными
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Основы программной
ero_28/task2.py"
Введите радиус цилиндра: 4
Введите высоту цилиндра: 6
Площадь боковой поверхности (введите 'боковая') или полная площадь цилиндра (введите 'полная')? бебебе
Некорректный выбор
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Основы программной
ero_28/task2.py"
Введите радиус цилиндра: 4
Введите высоту цилиндра: 6
Площадь боковой поверхности (введите 'боковая') или полная площадь цилиндра (введите 'полная')? полная
Полная площадь цилиндра: 251.32741228718345
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Основы программной
ero_28/task2.py"
Введите радиус цилиндра: 4
Введите высоту цилиндра: 6
Площадь боковой поверхности (введите 'боковая') или полная площадь цилиндра (введите 'полная')? боковая
Площадь боковой поверхности цилиндра: 158.79644737231807

```

Рисунок 8 – Код и результат выполнения программы.

5. Выполнили задание №3. 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.

```
task3.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def multiply():
6      product = 1
7      while True:
8          num = float(input("Введите число (введите 0 для завершения): "))
9          if num == 0:
10             print("Итоговое произведение: ", product)
11             break
12             product *= num
13             print("Промежуточное произведение: ", product)
14
15
16 if __name__ == "__main__":
17     multiply()
18
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Ya
egarepo_28/task3.py"
Введите число (введите 0 для завершения): 1
Промежуточное произведение: 1.0
Введите число (введите 0 для завершения): 7
Промежуточное произведение: 7.0
Введите число (введите 0 для завершения): 4
Промежуточное произведение: 28.0
Введите число (введите 0 для завершения): 7
Промежуточное произведение: 196.0
Введите число (введите 0 для завершения): 3
Промежуточное произведение: 588.0
Введите число (введите 0 для завершения): 7
Промежуточное произведение: 4116.0
Введите число (введите 0 для завершения): -4
Промежуточное произведение: -16464.0
Введите число (введите 0 для завершения): -1
Промежуточное произведение: 16464.0
Введите число (введите 0 для завершения): 0
Итоговое произведение: 16464.0
```

Рисунок 9 – Код и результат выполнения программы.

6. 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.

```
task4.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  def get_input():
8      return input("Введите значение: ")
9
10
11 def test_input(value):
12     if value.isdigit():
13         return True
14     else:
15         return False
16
17
18 def str_to_int(value):
19     return int(value)
20
21
22 def print_int(value):
23     print(value)
24
25
26 if __name__ == "__main__":
27     input_value = get_input()
28     if test_input(input_value):
29         int_value = str_to_int(input_value)
30         print_int(int_value)
31     else:
32         print("Введенное значение нельзя преобразовать в целое число.", file=sys.stderr)
33         exit(1)
34
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.locali
epo_28/task4.py"
Введите значение: 475
475
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.locali
epo_28/task4.py"
Введите значение: 743.4
Введенное значение нельзя преобразовать в целое число.
```

Рисунок 10 – Код и результат выполнения программы.

7. Выполнили индивидуальное задание. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.



```

individual.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  def command_add():
8      # Список студентов
9      students = []
10     # Запросить данные о студенте.
11     name = input("Фамилия и инициалы? ")
12     group = input("Номер группы? ")
13     grade = list(map(int, input("Успеваемость студента? (Пять оценок через пробел) ").split()))
14     while True:
15         if len(grade) < 5:
16             print("Введенное количество оценок меньше 5, введите оценки еще раз: ", file=sys.stderr)
17             grade = list(map(int, input("Успеваемость студента? (Пять оценок через пробел) ").split()))
18         else:
19             break
20
21     # Создать словарь.
22     if sum(grade)/len(grade) >= 4.0:
23         student = {
24             'name': name,
25             'group': group,
26             'grade': sum(grade)/len(grade),
27         }
28         # Добавить словарь в список.
29         students.append(student)
30
31     # Отсортировать список в случае необходимости.
32     if len(students) > 1:
33         students.sort(key=lambda item: item.get('name', ''))
34
35     return students
36
37
38 def command_list(students):
39     if students:
40         print("Список студентов с успеваемостью больше 4.0")
41         # Заголовок таблицы.
42         line = '+--{}--{}--{}--+'.format(
43             '-' * 4,
44             '-' * 30,
45             '-' * 20
46         )
47         print(line)
48         print(
49             '| {:^4} | {:^30} | {:^20} |'.format(
50                 "No",
51                 "Ф.И.О.",
52                 "Группа"
53             )
54         )
55         print(line)
56
57         # Вывести данные о всех сотрудниках.
58         for idx, student in enumerate(students, 1):
59             print(
60                 '| {:>4} | {:<30} | {:<20} |'.format(
61                     idx,
62                     student.get('name', ''),
63                     student.get('group', '')
64                 )
65             )
66
67         print(line)
68
69     else:
70         print("Студентов с успеваемостью выше 4.0 нет")
71
72
73 if __name__ == '__main__':
74     # Организовать бесконечный цикл запроса команд.
75     while True:
76         # Запросить команду из терминала.
77         command = input(">>> ").lower()
78
79         # Выполнить действие в соответствие с командой.
80         if command == 'exit':
81             break
82         elif command == 'add':
83             students = command_add()
84         elif command == 'list':
85             command_list(students)
86         else:
87             print(f"Неизвестная команда {command}", file=sys.stderr)
88

```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

```

itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Основы
>>> add
Фамилия и инициалы? Баратов С.Г.
Номер группы? 221
Успеваемость студента? (Пять оценок через пробел) 5 4 5 5 4
>>> list
Список студентов с успеваемостью больше 4.0
+-----+-----+-----+
| No |          Ф.И.О.          |          Группа          |
+-----+-----+-----+
| 1 | Баратов С.Г. | 221 |
+-----+-----+-----+
>>> exit
itssyoma@MacBook-Air-Sema megarepo_28 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Основы
>>> add
Фамилия и инициалы? Иванов Д.А.
Номер группы? 223
Успеваемость студента? (Пять оценок через пробел) 2 3 3 3 2
>>> list
Студентов с успеваемостью выше 4.0 нет

```

Рисунок 11 – Код и результат выполнения программы.

## **Ответы на контрольные вопросы**

### **1. Каково назначение функций в языке программирования Python?**

Функции в языке программирования Python используются для организации кода, повторного использования, абстракции и структурирования программы.

### **2. Каково назначение операторов def и return ?**

Оператор def используется для определения функции, а оператор return используется для возврата результата из функции.

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

Локальные переменные существуют только внутри функции и видны только внутри этой функции, тогда как глобальные переменные могут быть доступны из любого места программы.

### **4. Как вернуть несколько значений из функции Python?**

Для возврата нескольких значений из функции в Python можно использовать кортеж, список или словарь.

### **5. Какие существуют способы передачи значений в функцию?**

Значения можно передавать в функцию через аргументы, ключевые слова или аргументы по умолчанию.

### **6. Как задать значение аргументов функции по умолчанию?**

Значения аргументов функции по умолчанию задаются путем указания значения после имени аргумента в определении функции.

### **7. Каково назначение lambda-выражений в языке Python?**

Lambda-выражения в Python представляют собой анонимные функции, которые могут содержать только одно выражение.

### **8. Как осуществляется документирование кода согласно PEP257?**

Документирование кода согласно PEP257 осуществляется путем добавления строк документации (docstrings) к определению функции, модуля или класса.

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

Однострочные строки документации обычно используются для краткого описания объекта, тогда как многострочные строки документации могут содержать более подробное описание, включая примеры использования, параметры и возвращаемые значения.