

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 №13
дисциплины «Программирование на Python»

Выполнил:
Говоров Егор Юрьевич
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 _____

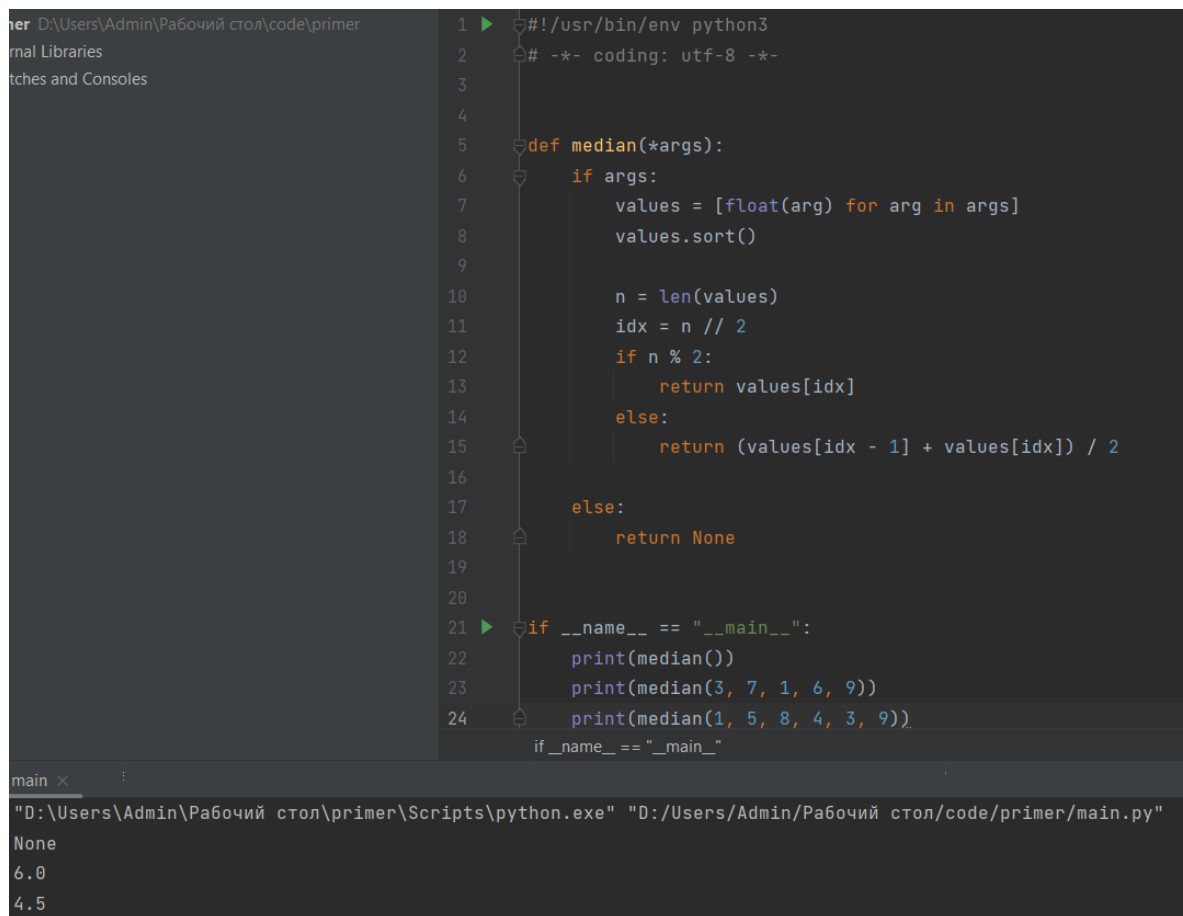
Ставрополь, 2023 г.

Тема: Функции с переменным числом параметров в Python

Цель работы: приобретение навыков по работе с функциями с переменным числом параметров 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.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def median(*args):
6      if args:
7          values = [float(arg) for arg in args]
8          values.sort()
9
10         n = len(values)
11         idx = n // 2
12         if n % 2:
13             return values[idx]
14         else:
15             return (values[idx - 1] + values[idx]) / 2
16
17     else:
18         return None
19
20
21  if __name__ == "__main__":
22      print(median())
23      print(median(3, 7, 1, 6, 9))
24      print(median(1, 5, 8, 4, 3, 9))
25
26  if __name__ == "__main__":
```

main x

"D:\Users\Admin\Рабочий стол\primer\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/primer/main.py"

None

6.0

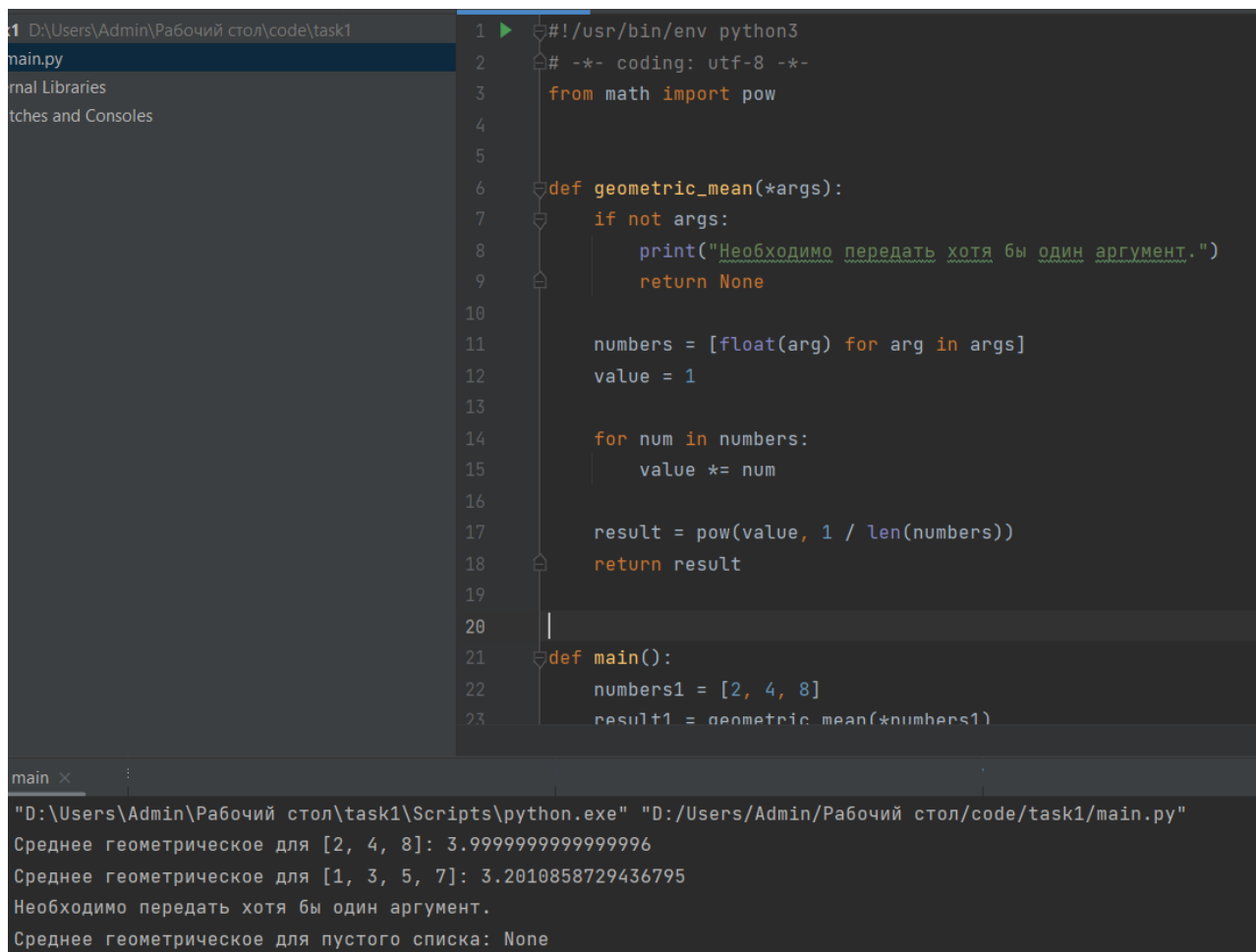
4.5

Рис 1. Выполнение задания из примера

5. Решил следующую задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

$$G = \sqrt[n]{\prod_{k=1}^n a_k}.$$

Если функции передается пустой список аргументов, то она должна возвращать значение None



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  from math import pow
4
5
6  def geometric_mean(*args):
7      if not args:
8          print("Необходимо передать хотя бы один аргумент.")
9          return None
10
11         numbers = [float(arg) for arg in args]
12         value = 1
13
14         for num in numbers:
15             value *= num
16
17         result = pow(value, 1 / len(numbers))
18         return result
19
20
21 def main():
22     numbers1 = [2, 4, 8]
23     result1 = geometric_mean(*numbers1)
```

main x

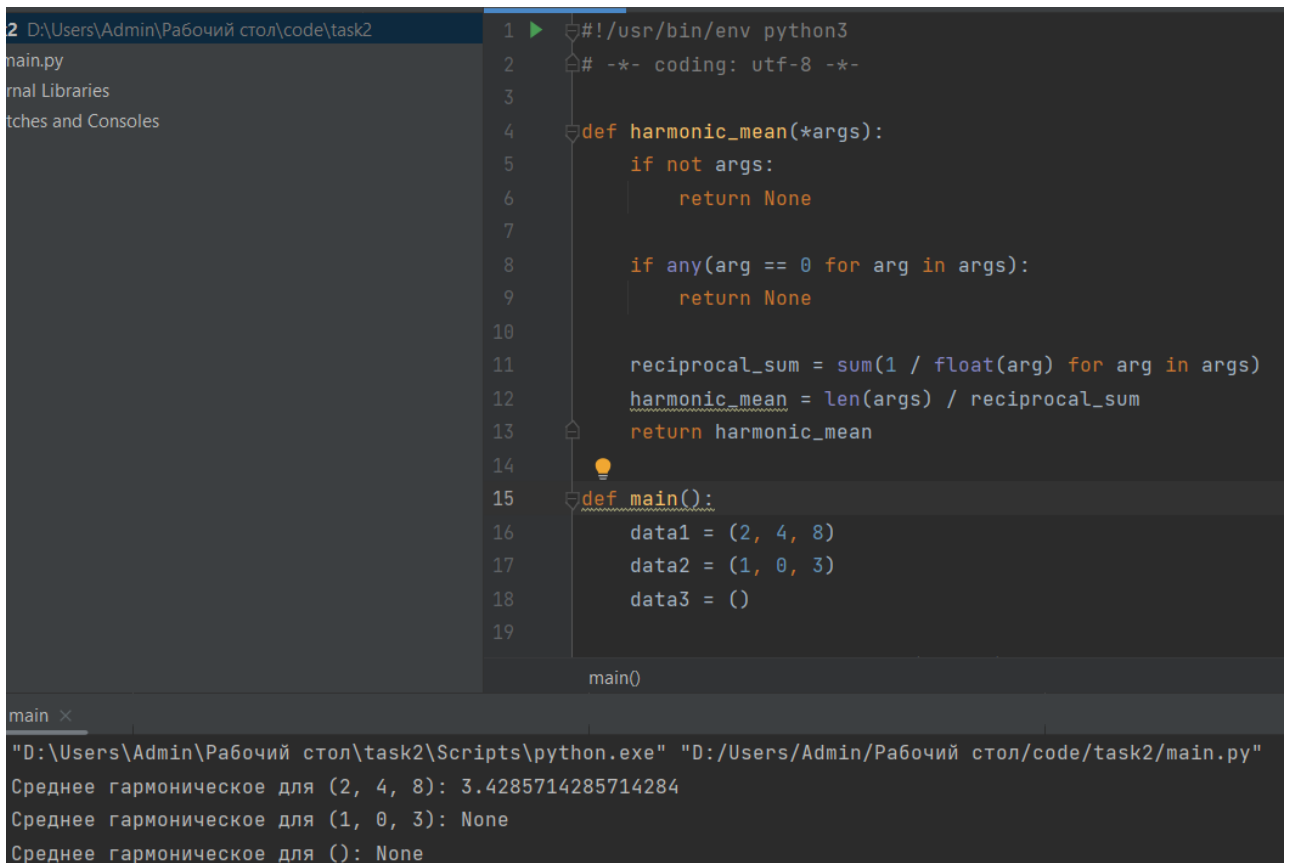
```
"D:\Users\Admin\Рабочий стол\task1\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task1/main.py"
Среднее геометрическое для [2, 4, 8]: 3.9999999999999996
Среднее геометрическое для [1, 3, 5, 7]: 3.2010858729436795
Необходимо передать хотя бы один аргумент.
Среднее геометрическое для пустого списка: None
```

Рис 2. Выполнение задания 1

6. Решил следующую задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов a_1, a_2, \dots, a_n

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}.$$

Если функции передается пустой список аргументов, то она должна возвращать значение `None`.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def harmonic_mean(*args):
5      if not args:
6          return None
7
8      if any(arg == 0 for arg in args):
9          return None
10
11     reciprocal_sum = sum(1 / float(arg) for arg in args)
12     harmonic_mean = len(args) / reciprocal_sum
13     return harmonic_mean
14
15 def main():
16     data1 = (2, 4, 8)
17     data2 = (1, 0, 3)
18     data3 = ()
19
20     main()
```

main ×

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task2/main.py"

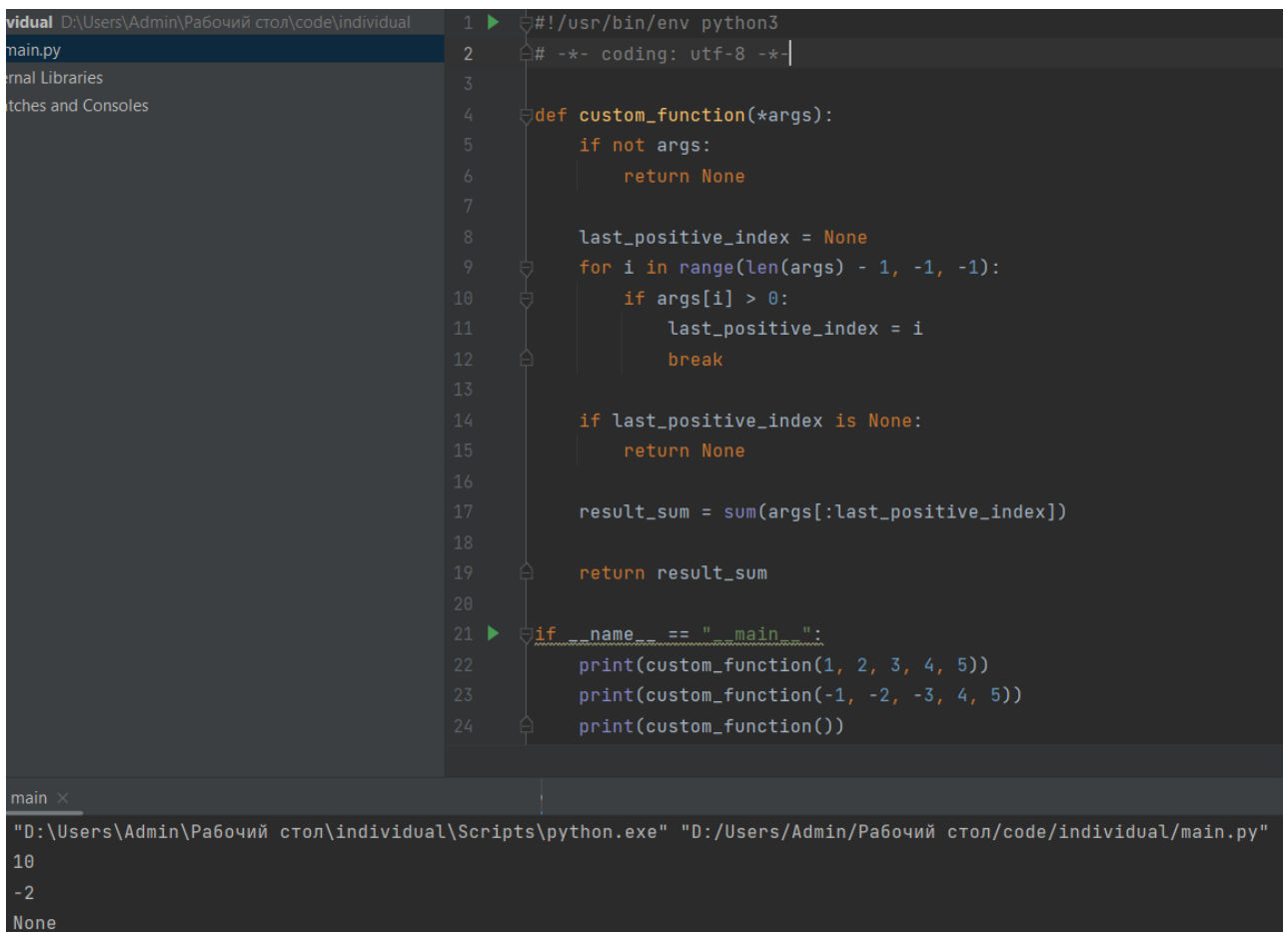
Среднее гармоническое для (2, 4, 8): 3.4285714285714284

Среднее гармоническое для (1, 0, 3): None

Среднее гармоническое для (): None

Рис 3. Выполнение задания 2

7. Выполнил индивидуальное задание, согласно варианту 5. Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None . Номер варианта определяется по согласованию с преподавателем. В процессе решения не использовать преобразования конструкции *args в список или иную структуру данных. Согласно варианту 5, необходимо найти сумму аргументов, расположенных до последнего положительного аргумента.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def custom_function(*args):
5     if not args:
6         return None
7
8     last_positive_index = None
9     for i in range(len(args) - 1, -1, -1):
10         if args[i] > 0:
11             last_positive_index = i
12             break
13
14     if last_positive_index is None:
15         return None
16
17     result_sum = sum(args[:last_positive_index])
18
19     return result_sum
20
21 if __name__ == "__main__":
22     print(custom_function(1, 2, 3, 4, 5))
23     print(custom_function(-1, -2, -3, 4, 5))
24     print(custom_function())
```

main x

"D:\Users\Admin\Рабочий стол\individual\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/individual/main.py"

10

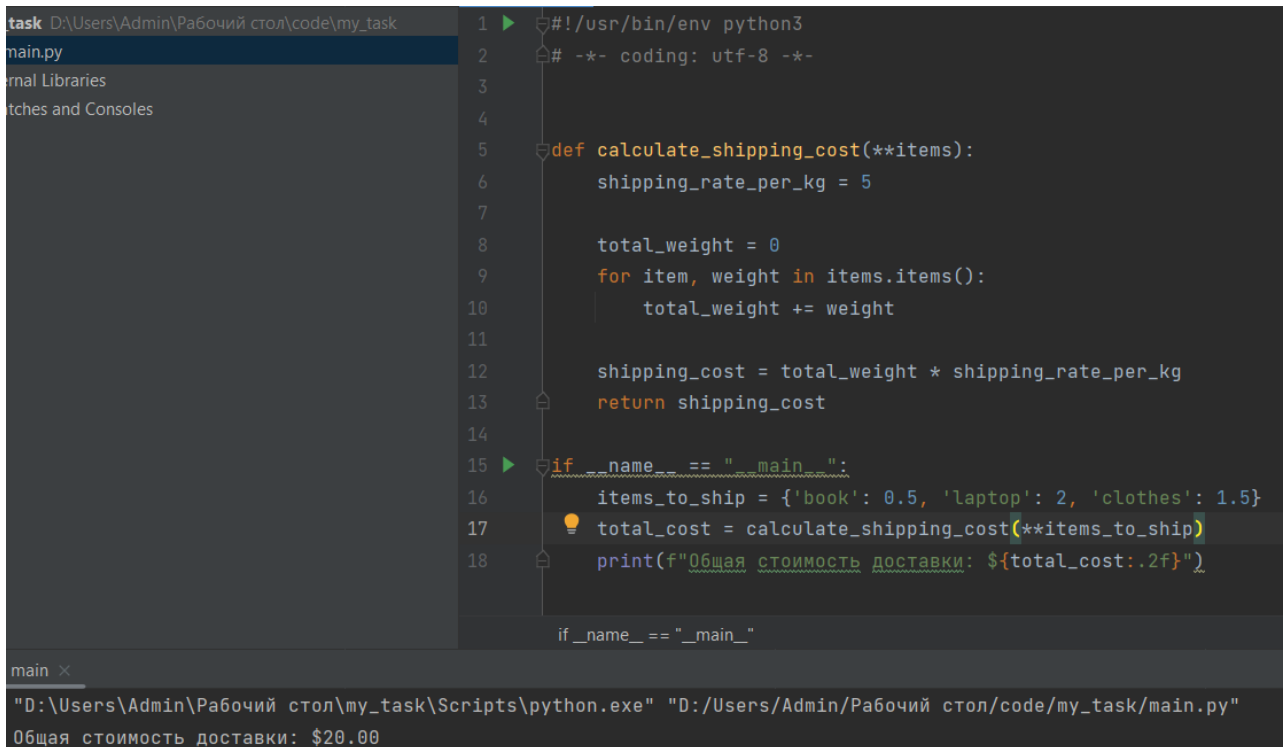
-2

None

Рис 4. Выполнение индивидуального задания

8. Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Задача: Решить задачу, в которой функция `calculate_shipping_cost` использует переменное число именованных аргументов (`**items`), где каждый аргумент представляет товар и его вес. Функция затем рассчитывает общую стоимость доставки на основе суммарного веса всех товаров и фиксированной стоимости доставки за килограмм.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def calculate_shipping_cost(**items):
6     shipping_rate_per_kg = 5
7
8     total_weight = 0
9     for item, weight in items.items():
10         total_weight += weight
11
12     shipping_cost = total_weight * shipping_rate_per_kg
13     return shipping_cost
14
15 if __name__ == "__main__":
16     items_to_ship = {'book': 0.5, 'laptop': 2, 'clothes': 1.5}
17     total_cost = calculate_shipping_cost(**items_to_ship)
18     print(f"Общая стоимость доставки: ${total_cost:.2f}")
19
20 if __name__ == "__main__":
```

main x

"D:\Users\Admin\Рабочий стол\my_task\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/my_task/main.py"

Общая стоимость доставки: \$20.00

Рис 5. Выполнение задания

Контрольные вопросы

1. Какие аргументы называются позиционными в Python?

В Python аргументы называются позиционными, если они передаются функции в том же порядке, в котором они определены в функции.

В функцию также можно передать переменное количество позиционных аргументов. Это делается с помощью оператора `*` перед именем аргумента в определении функции.

2. Какие аргументы называются именованными в Python?

В Python аргументы называются именованными, если они передаются функции с указанием имени аргумента, за которым следует значение аргумента.

В функцию также можно передать переменное количество именованных аргументов. Это делается с помощью оператора `**` перед именем аргумента в определении функции.

3. Для чего используется оператор `*` ?

Оператор `*` чаще всего ассоциируется у людей с операцией умножения, но в Python он имеет и другой смысл. Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

4. Каково назначение конструкций `*args` и `**kwargs` ?

Итак, мы знаем о том, что оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Знаем мы и о том, что существует два вида параметров функций. А именно, `*args` — это сокращение от «arguments» (аргументы), а `**kwargs` — сокращение от «keyword arguments» (именованные аргументы).

Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины.

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