

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:

Касимов Асхаб Арсенович
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)

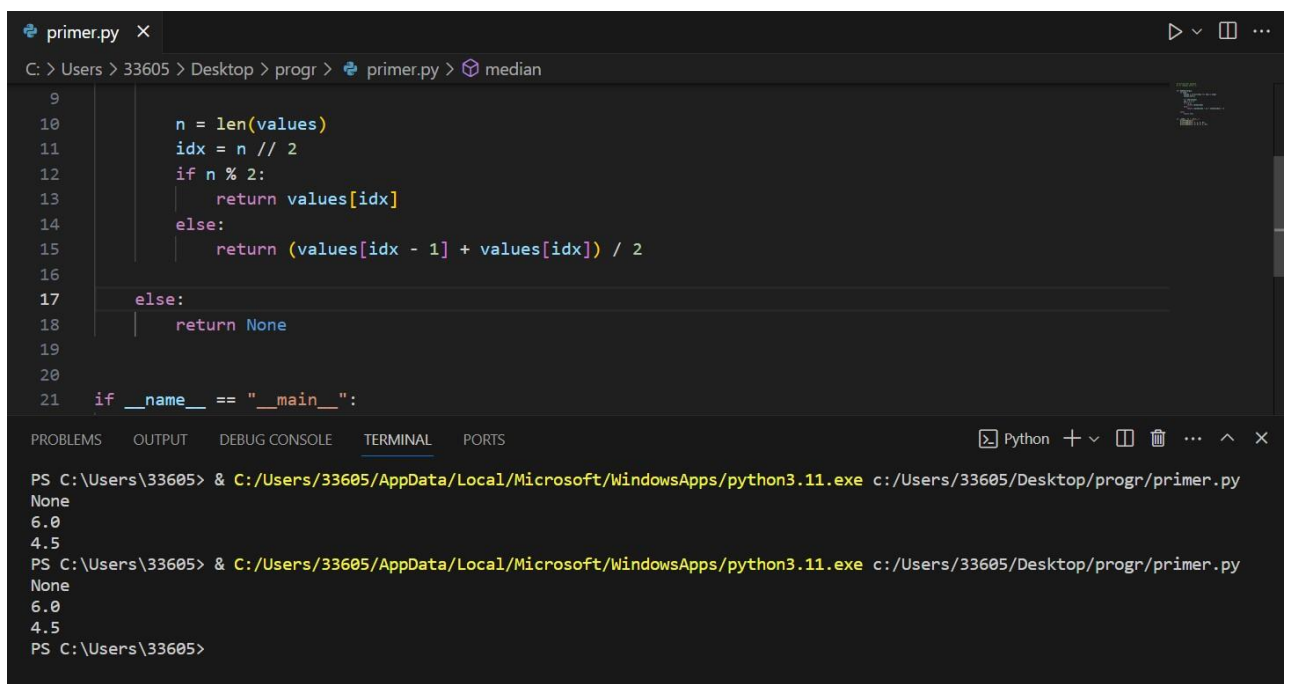
Ставрополь, 2023 г.

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.



```
primer.py x
C: > Users > 33605 > Desktop > progr > primer.py > median

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__":

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python + - □ □ □ ... ^ x

PS C:\Users\33605> & C:/Users/33605/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/33605/Desktop/progr/primer.py
None
6.0
4.5
PS C:\Users\33605> & C:/Users/33605/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/33605/Desktop/progr/primer.py
None
6.0
4.5
PS C:\Users\33605>
```

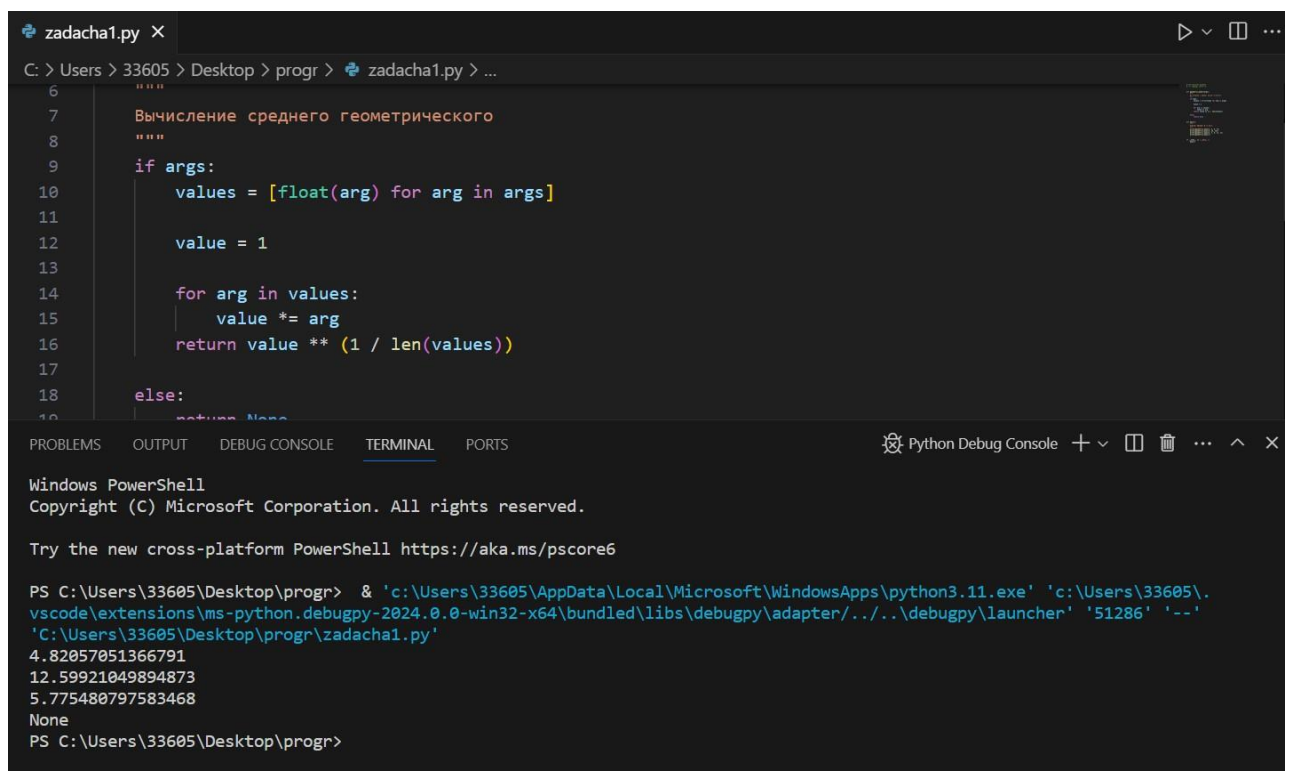
5.

Рисунок 1. Результат работы программы из примера 1

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



```
zadacha1.py X
C: > Users > 33605 > Desktop > progr > zadacha1.py > ...

6      """
7      Вычисление среднего геометрического
8      """
9      if args:
10         values = [float(arg) for arg in args]
11
12         value = 1
13
14         for arg in values:
15             value *= arg
16         return value ** (1 / len(values))
17
18     else:
19         return None

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python Debug Console + - [] [X] ... ^ X

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\33605\Desktop\progr> & 'c:\Users\33605\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\33605\vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '51286' '--' 'C:\Users\33605\Desktop\progr\zadacha1.py'
4.82057051366791
12.59921049894873
5.775480797583468
None
PS C:\Users\33605\Desktop\progr>
```

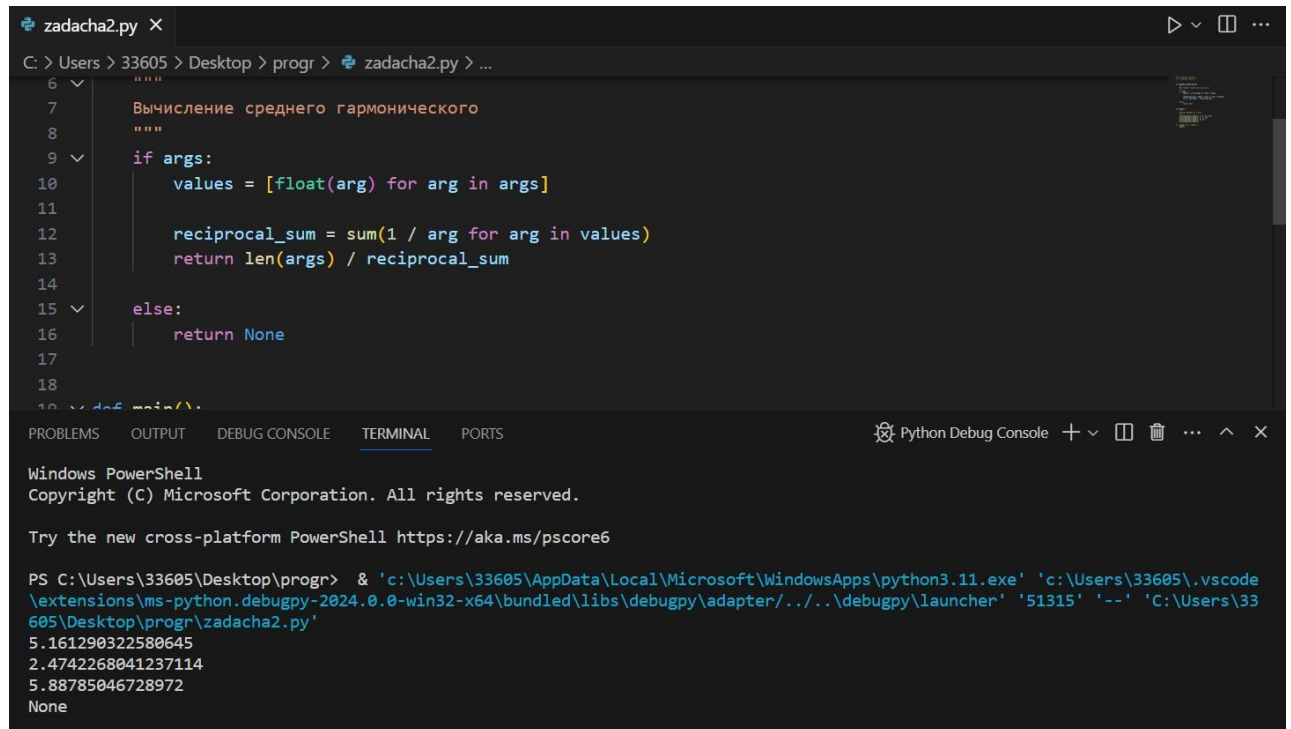
Рисунок 2. Результат работы программы из задачи 1

7.

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

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

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



```
zadacha2.py X
C: > Users > 33605 > Desktop > progr > zadacha2.py > ...
6  """
7  Вычисление среднего гармонического
8  """
9  if args:
10     values = [float(arg) for arg in args]
11
12     reciprocal_sum = sum(1 / arg for arg in values)
13     return len(args) / reciprocal_sum
14
15 else:
16     return None
17
18
19 def main():
20     ...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console + - [] ... ^ x

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

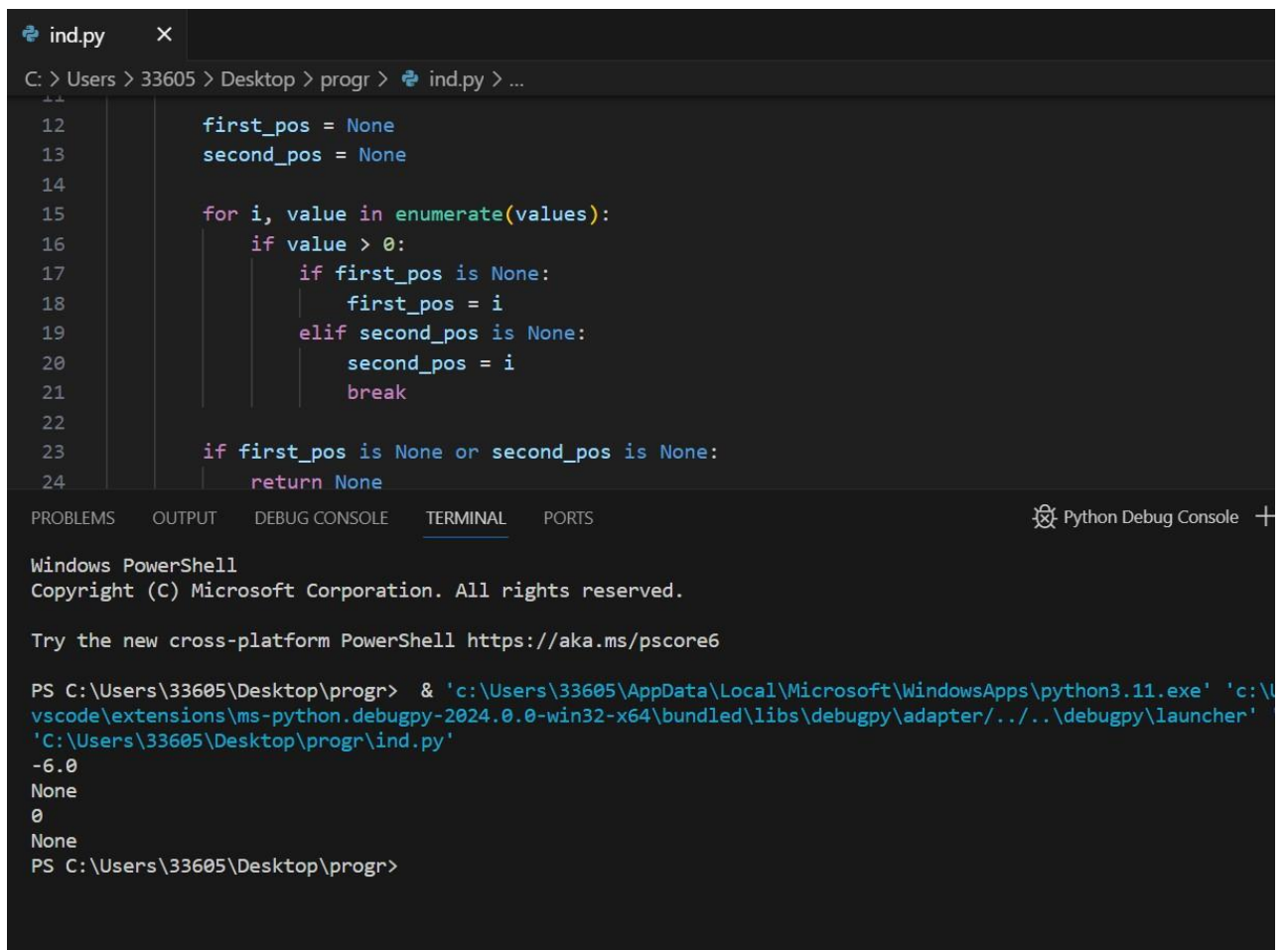
Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\33605\Desktop\progr> & 'c:\Users\33605\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\33605\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '51315' '--' 'C:\Users\33605\Desktop\progr\zadacha2.py'
5.161290322580645
2.4742268041237114
5.88785046728972
None
```

Рисунок 3. Результат работы программы из задачи 2

8.

Выполнил индивидуальное задание, согласно варианту 8. 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 8, необходимо найти сумму аргументов, расположенных между первым и вторым положительными аргументами.



```
ind.py x
C: > Users > 33605 > Desktop > progr > ind.py > ...
12     first_pos = None
13     second_pos = None
14
15     for i, value in enumerate(values):
16         if value > 0:
17             if first_pos is None:
18                 first_pos = i
19             elif second_pos is None:
20                 second_pos = i
21             break
22
23     if first_pos is None or second_pos is None:
24         return None

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python Debug Console +

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

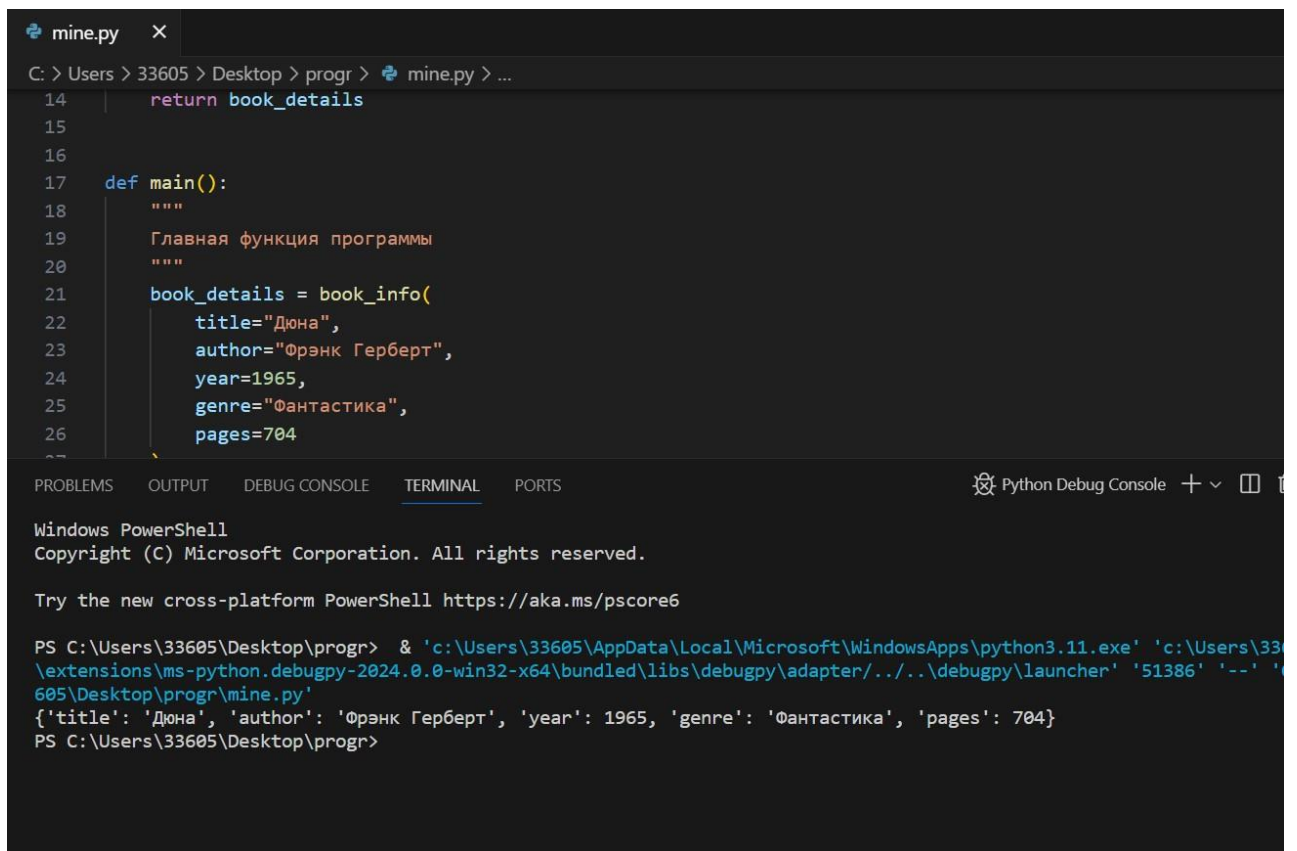
PS C:\Users\33605\Desktop\progr> & 'c:\Users\33605\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\33605\Desktop\progr\ind.py'
-6.0
None
0
None
PS C:\Users\33605\Desktop\progr>
```

9.

Рисунок 4. Результат работы программы из индивидуального задания

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

Задача: Напишите функцию, которая принимает произвольное количество именованных аргументов, представляющих информацию о книге (например, название, автор, год издания и т. д.). Функция должна собирать эту информацию в словарь и выводить его.



```
mine.py x
C: > Users > 33605 > Desktop > progr > mine.py > ...
14     return book_details
15
16
17 def main():
18     """
19     Главная функция программы
20     """
21     book_details = book_info(
22         title="Дюна",
23         author="Фрэнк Герберт",
24         year=1965,
25         genre="Фантастика",
26         pages=704
27     )
28
29
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python Debug Console + - []
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\33605\Desktop\progr> & 'c:\Users\33605\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\33605\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '51386' '--' 'c:\Users\33605\Desktop\progr\mine.py'
{'title': 'Дюна', 'author': 'Фрэнк Герберт', 'year': 1965, 'genre': 'Фантастика', 'pages': 704}
PS C:\Users\33605\Desktop\progr>
```

10.

Рисунок 5. Результат работы программы из придуманного 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