**МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет Математики и Информатики**

**Департамент Информатики**

Шелестян Анастасия

отчет

по дисциплине «Программирование в PYTHON»

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Плешка Наталья, лектор

(подпись)

Автор: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шелестян Анастасия

(подпись)

Кишинев, 2024

Оглавление

[Цель работы. 2](#_Toc160747666)

[Логика реализованных алгоритмов. 3](#_Toc160747667)

[Выполнение задания: 4](#_Toc160747668)

[Функции: 7](#_Toc160747669)

[Выводы. 9](#_Toc160747670)

[Библиография 9](#_Toc160747671)

# Цель работы.

**Вариант 1. Управление данными о книгах автора**

Напишите скрипт *Python*, отвечающий за учет информацией об авторах и их книгах, используя условные инструкции, циклы и методы работы с коллекциями. Инициализируйте пустой словарь, где ключами будут имена авторов, а значениями – списки их книг

Представить пользователю в виде меню следующие возможности:

1. Добавить нового автора (его фамилию).
2. Добавить книгу к существующему автору (книг может быть и несколько).
3. Просмотреть список авторов и их книг.
4. Вывести сколько книг у каждого автора.
5. Удалить какой-то элемент из словаря - автора и все его книги.
6. Выход из программы.

Предусмотреть обработку ошибок, таких как ввод некорректных данных.

Выводить обновленную информацию после каждой операции.

Скрипт гарантирует, что пользователь получит соответствующие сообщения, если он введет неверные значения.

Для решения задачи используйте функции и все функции храните в отдельном модуле.

# Логика реализованных алгоритмов.

Для того чтоб выполнить поставленную задачу необходимо следующее:

**Инициализация пустого словаря:**

* + Создать пустой словарь, где ключами будут имена авторов, а значениями – списки их книг.

**Добавление нового автора (его имя (и/или) фамилии):**

* + Запросить у пользователя ввод фамилии нового автора.
  + Проверить, существует ли уже такой автор в словаре.
  + Если нет, добавить нового автора в словарь.

**Добавление книги к существующему автору:**

* + Запросить у пользователя ввод фамилии автора.
  + Проверить, существует ли такой автор в словаре.
  + Если существует, запросить у пользователя ввод названия книги.
  + Добавить книгу в список книг данного автора.

**Просмотр списка авторов и их книг:**

* + Пройтись по элементам словаря и вывести на экран имена авторов и их списки книг.

**Вывод количества книг у каждого автора:**

* + Пройтись по элементам словаря и вывести на экран количество книг для каждого автора.

**Удаление элемента из словаря (автора и всех его книг):**

* + Запросить у пользователя ввод фамилии автора, которого нужно удалить.
  + Если такой автор существует, удалить его из словаря.

**Выход из программы:**

* + Предусмотреть выход из программы по выбору пользователя.

**Обработка ошибок:**

* + Обеспечить обработку некорректного ввода данных, чтобы предотвратить сбои программы.

**Вывод обновленной информации:**

* + После выполнения каждой операции выводить обновленную информацию о состоянии словаря.

# Выполнение задания:

Для начала создаю два документа с разрешением .ру в одном из которых будет прописан скрипт программы, в другом созданные функции.

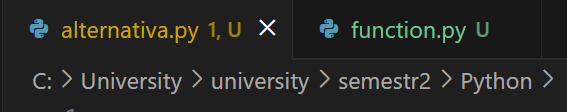


Рис. 1

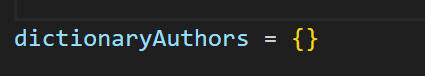


Рис. 2

Инициализация пустого словаря dictionaryAuthors, где ключами будут имена авторов, а значениями – списки их книг.

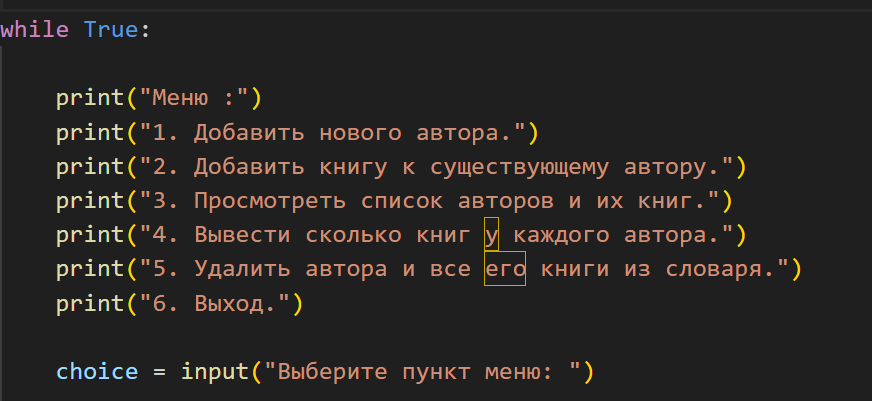


Рис. 3

На данном изображении показано использование бесконечного цикла while true для отображения меню и обработки выбора пользователя, а также отображение с доступными опциями меню для пользователя. После запуска программы в консоли будет показано следующее..

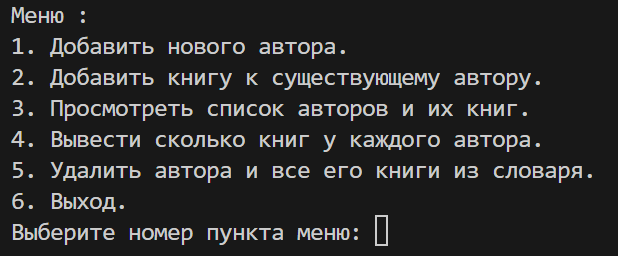


Рис. 4

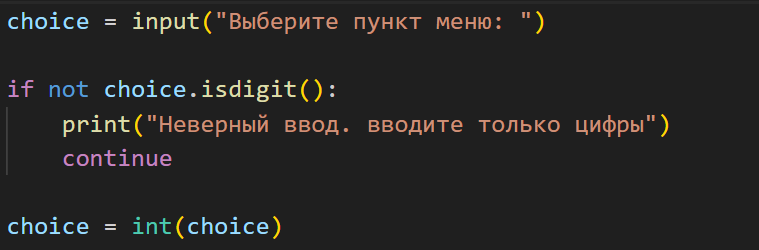


Рис. 5

Получение выбора от пользователя, проверка на то, что веденное значение является числом. Если это не число, то выводиться сообщение об ошибке, и цикл начинается заново. Последняя строчка позволяет преобразование введенного пользователем значения в целое число.

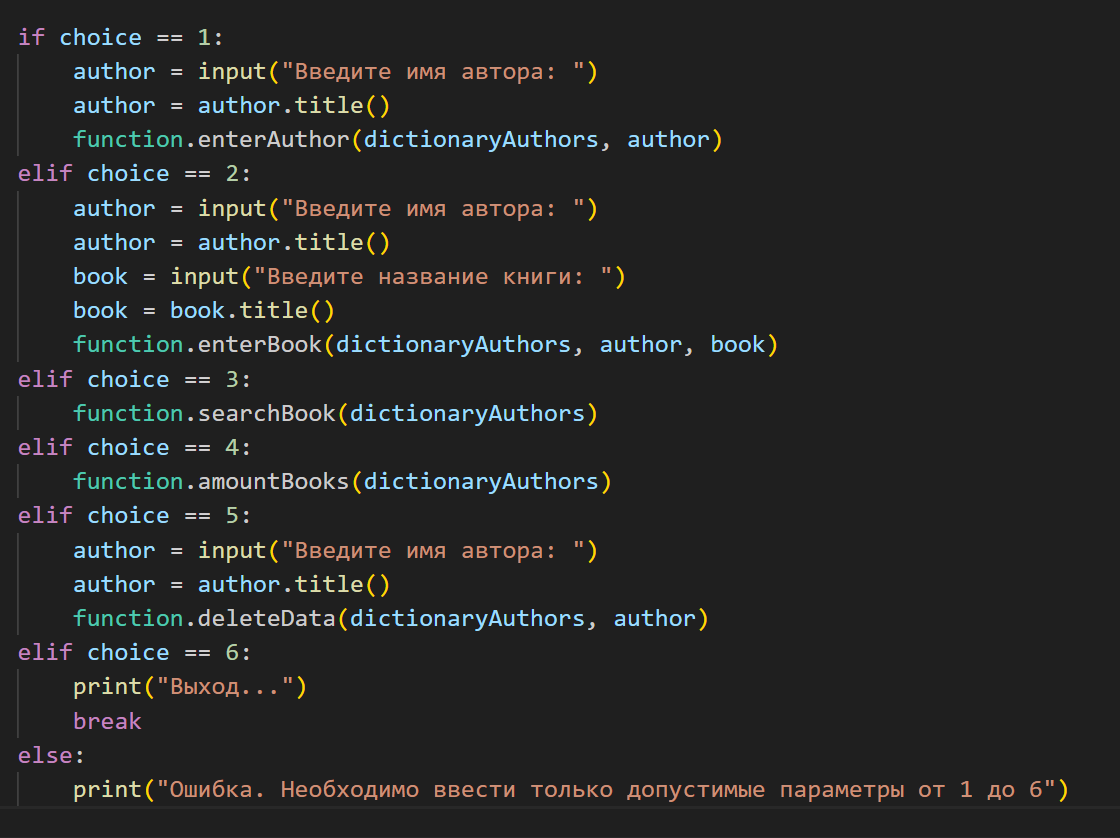


Рис. 6

Далее прописываю кейсы выбора для по меню для пользователя. Если пользователь вводит имена или фамилии с маленькой буквы то вывод преобразуется в заглавную буквы благодаря методу .title и так для каждого кейска. Если выбор пользователя равен 6, выводится сообщение о выходе из программы, и цикл завершается. Если введенное значение не соответствует ни одному из допустимых вариантов, выводится сообщение об ошибке.

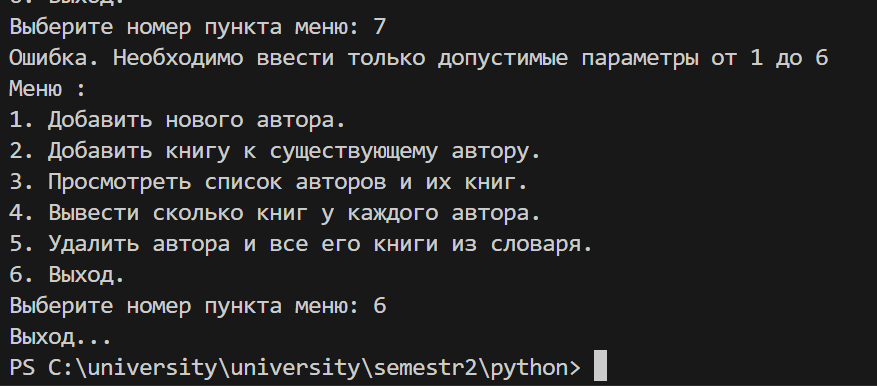


Рис. 7

# Функции:

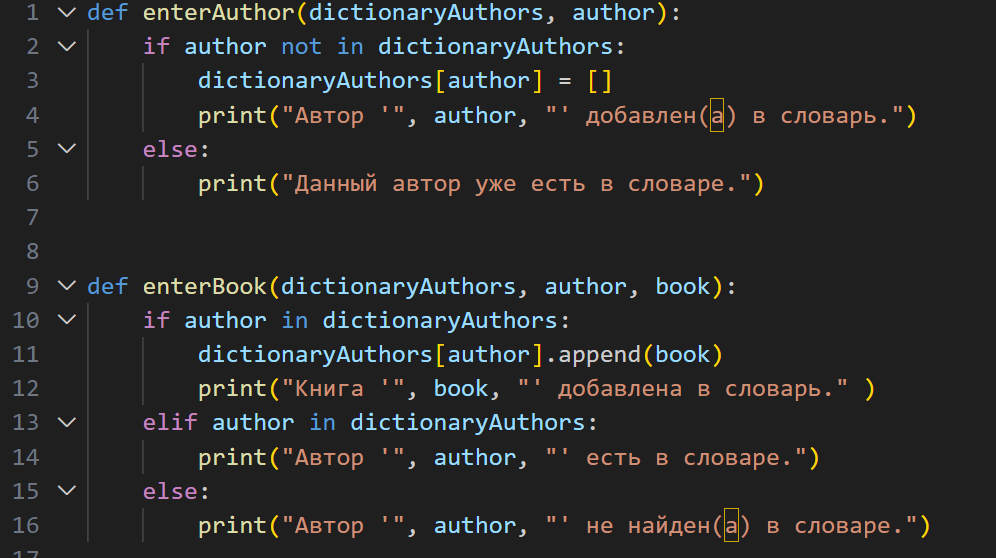


Рис. 8

***enterAuthor(dictionaryAuthors, author)***

Функция проверяет, есть ли указанный автор в словаре dictionaryAuthors.

Если автора нет, добавляет его в словарь и присваивает пустой список в качестве значения.

Выводит сообщение об успешном добавлении или сообщение, что автор уже существует.

***enterBook(dictionaryAuthors, author, book)***

Функция проверяет, существует ли указанный автор в словаре dictionaryAuthors.

Если автор существует, добавляет указанную книгу в список книг данного автора.

Выводит сообщение об успешном добавлении или сообщение, что указанный автор не найден в словаре.

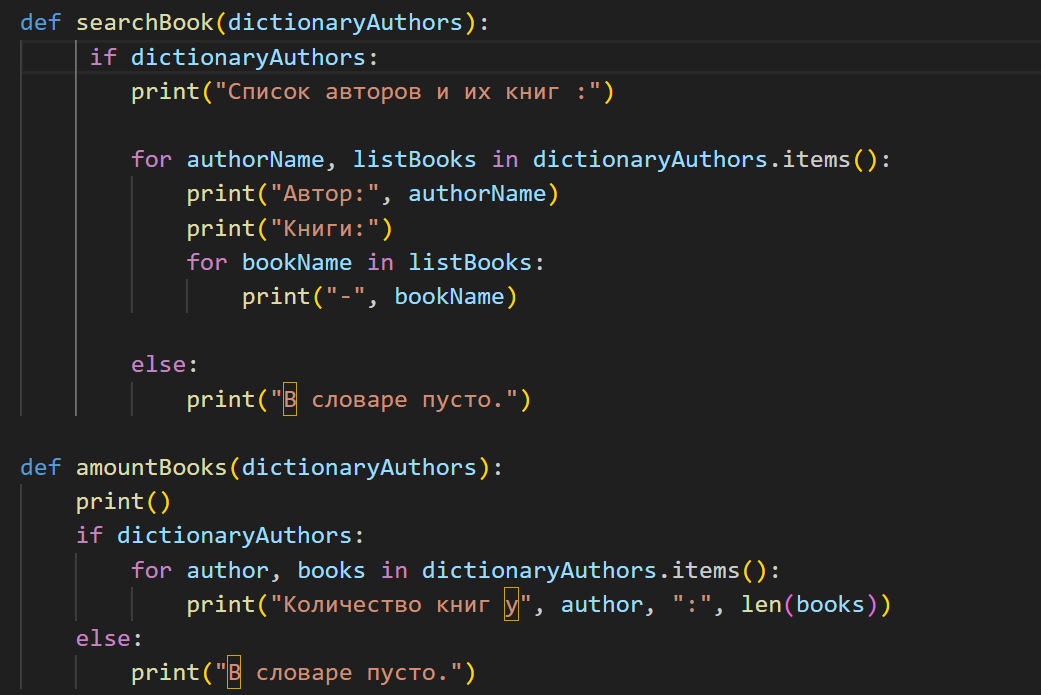


Рис. 9

***searchBook(dictionaryAuthors)***

Функция выводит список авторов и их книг.

Если словарь dictionaryAuthors не пуст, проходится по каждому автору и выводит его книги.

Если словарь пуст, выводит сообщение о том, что словарь пуст.

***amountBooks(dictionaryAuthors)***

Функция выводит количество книг у каждого автора в словаре.

Если словарь dictionaryAuthors не пуст, проходится по каждому автору и выводит количество его книг.

Если словарь пуст, выводит сообщение о том, что словарь пуст.

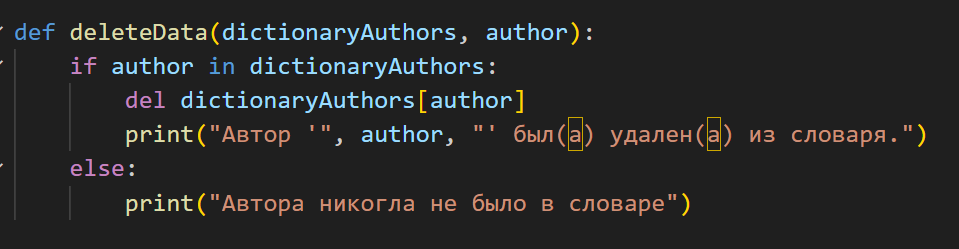


Рис. 10

***deleteData(dictionaryAuthors, author)***

Функция проверяет, существует ли указанный автор в словаре dictionaryAuthors.

Если автор существует, удаляет его из словаря.

Выводит сообщение об успешном удалении или сообщение, что указанный автор не найден в словаре.

# Выводы.

Язык Python предоставляет удобные средства для решения задач по учету информации об авторах и их книгах. Он позволяет использовать словари для организации данных об авторах и их книгах, а также предоставляет лаконичный и понятный синтаксис для взаимодействия с пользователем.

В коде используются функции для разделения логики программы на более мелкие и понятные блоки, что способствует повторному использованию кода и облегчает понимание программы. Встроенные функции, такие как input(), print(), isdigit(), title(), а также удобные методы работы со словарями, делают код более ясным и компактным.

Обработка ошибок, таких как некорректный ввод данных, также реализована, что улучшает надежность программы. Вывод обновленной информации после каждой операции предоставляет пользователю обратную связь о выполненных действиях.

В целом, язык Python предоставляет простой и эффективный способ решения подобных задач, делая код читаемым и понятным.

# Библиография

1. Материалы на сайте курса (https://moodle.usm.md)

2. https://www.python.org/