**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. 3](#_Toc160747666)

[Логика реализованных алгоритмов. 3](#_Toc160747667)

[Выполнение задания: 4](#_Toc160747668)

[Функции: 7](#_Toc160747669)

[Выводы. 10](#_Toc160747670)

[Библиография 10](#_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.

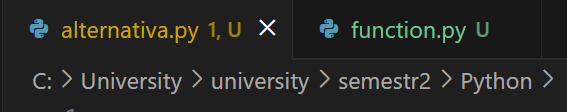


Рис. 1

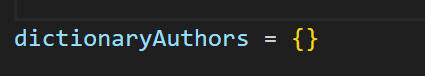


Рис. 2

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

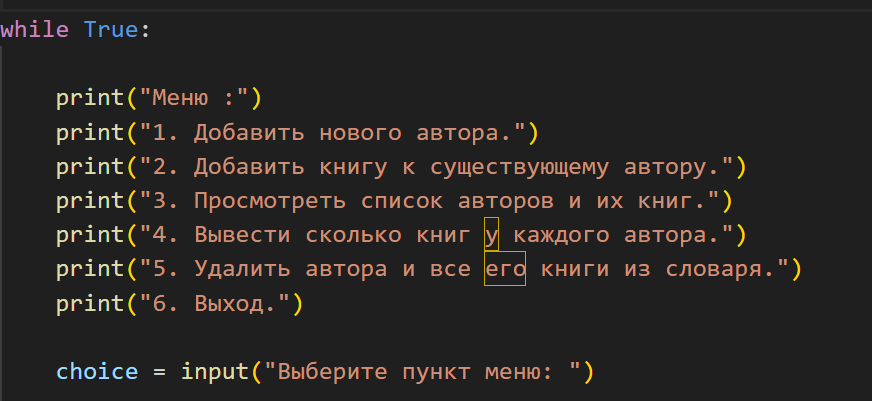


Рис. 3

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..

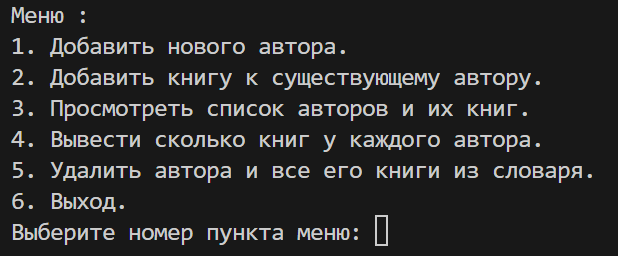


Рис. 4

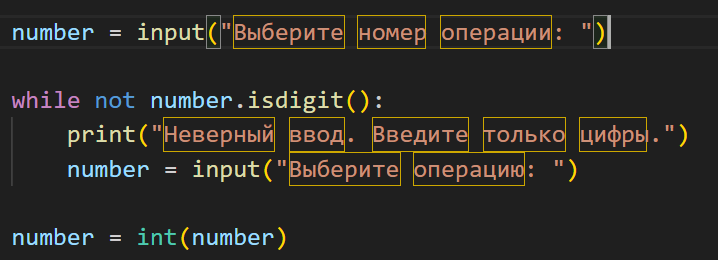


Рис. 5

***number = input***("Выберите номер операции: "): Запрашивает у пользователя ввод номера операции и сохраняет введенное значение в переменной number.

***while not number.isdigit()***:: цикл while, продолжает выполняться, пока введенное значение не является целым числом. Метод isdigit() проверяет, состоит ли строка только из цифр.

***print***("Неверный ввод. Введите только цифры."): Если введенная строка не является цифрой, выводится сообщение об ошибке.

***number = input***("Выберите операцию: "): Снова запрашивает ввод пользователя.

***number = int(number):*** Когда пользователь вводит корректное целое число, программа преобразует строку в целое число с помощью int(). Теперь переменная number содержит целочисленное значение.

В заключение можно сказать что обеспечивает контроль за тем, чтобы пользователь ввел цифры, и преобразует их в целое число для дальнейшей обработки.

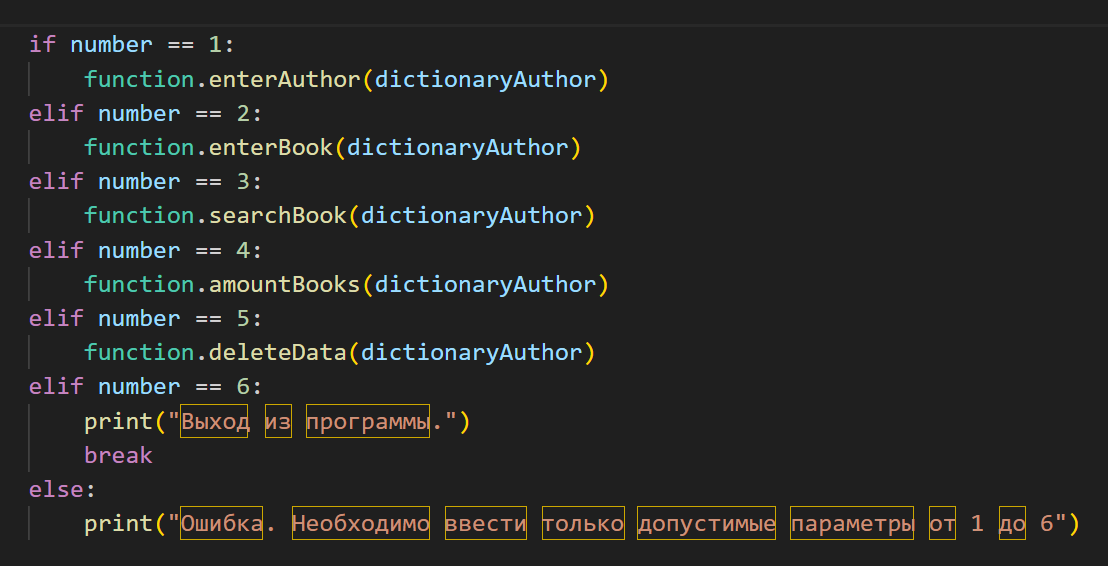


Рис. 6

Кода использует значение переменной number, которое пользователь ввел и сконвертировал в целое число, для выбора соответствующей операции. Давайте рассмотрим каждую ветвь условного оператора:

if number == 1:: Если введенное число равно 1, вызывается функция function.enterAuthor(dictionaryAuthor).

elif number == 2:: Если введенное число равно 2, вызывается функция function.enterBook(dictionaryAuthor).

elif number == 3:: Если введенное число равно 3, вызывается функция function.searchBook(dictionaryAuthor).

elif number == 4:: Если введенное число равно 4, вызывается функция function.amountBooks(dictionaryAuthor).

elif number == 5:: Если введенное число равно 5, вызывается функция function.deleteData(dictionaryAuthor).

elif number == 6:: Если введенное число равно 6, программа выводит сообщение о выходе из программы (print("Выход из программы.")) и завершает выполнение цикла с помощью break.

else:: Если введенное число не соответствует ни одной из вышеуказанных ветвей (не находится в диапазоне от 1 до 6), выводится сообщение об ошибке.

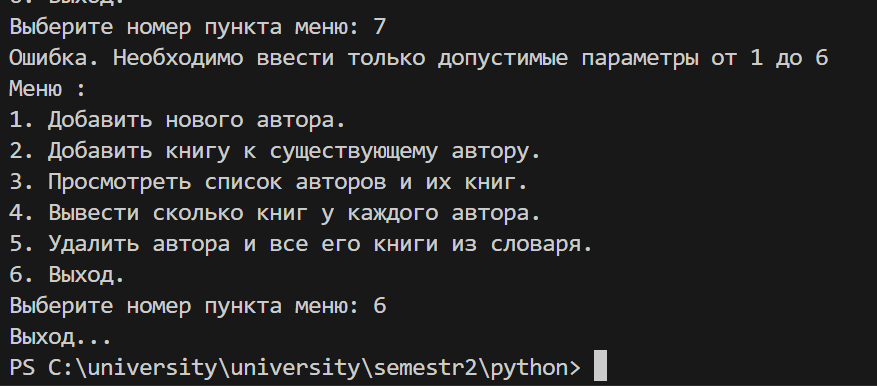


Рис. 7

# Функции:

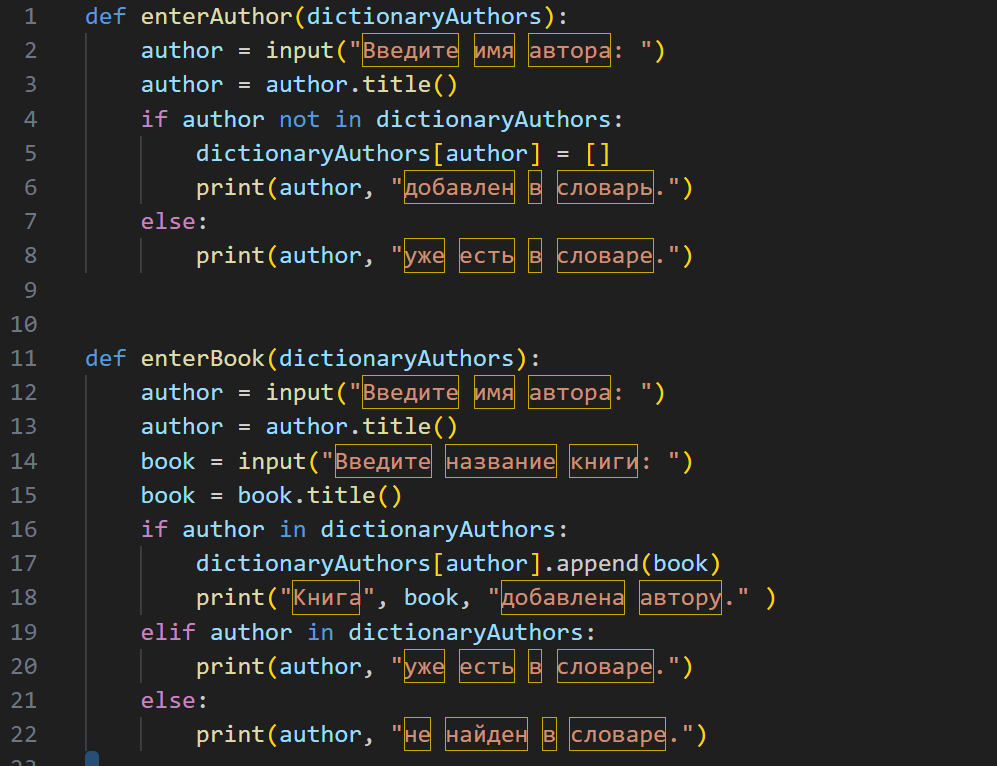


Рис. 8

Функция (***def enterAuthor(dictionaryAuthors)*** запрашивает у пользователя ввод имени автора и приводит его к формату с заглавной буквы (.title()).

Затем проверяет, есть ли автор уже в словаре (if author not in dictionaryAuthors:).

Если автора нет, то создается новая запись в словаре с пустым списком книг для этого автора.

Если автор уже есть в словаре, выводится сообщение о том, что автор уже существует.

Функция (***def enterBook(dictionaryAuthors***): запрашивает у пользователя ввод имени автора и названия книги, приводит их к формату с заглавной буквы.

Проверяет, есть ли автор в словаре (if author in dictionaryAuthors:).

Если автор есть в словаре, то название книги добавляется к списку книг для данного автора.

Если автор не найден в словаре, выводится соответствующее сообщение***.***

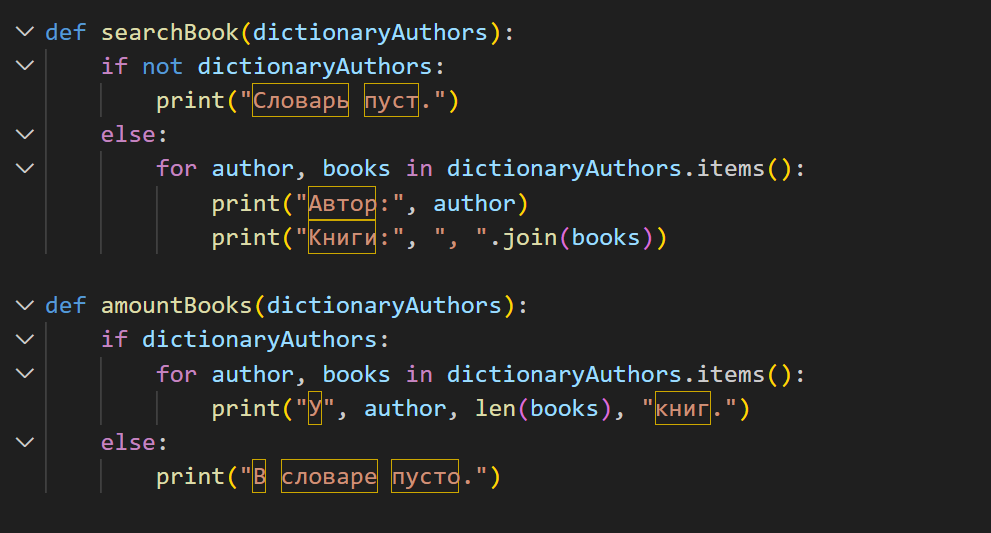


Рис. 9

***searchBook(dictionaryAuthors)***

if not dictionaryAuthors:: Проверяет, пуст ли словарь. Если словарь пуст, выводится сообщение о том, что словарь пуст.

else:: Если словарь не пуст, проходится по каждому элементу словаря (автору и его книгам) с помощью цикла for.

for author, books in dictionaryAuthors.items():: Перебор элементов словаря, где author - это ключ (имя автора), а books - значение (список книг автора).

print("Автор:", author): Выводит имя автора.

print("Книги:", ", ".join(books)): Выводит книги автора, разделенные запятой и пробелом.

***amountBooks(dictionaryAuthors)***

if dictionaryAuthors:: Проверяет, не является ли словарь пустым. Если словарь не пуст, проходится по каждому элементу словаря с помощью цикла for.

for author, books in dictionaryAuthors.items():: Перебор элементов словаря.

print("У", author, len(books), "книг."): Выводит количество книг для каждого автора.

else:: Если словарь пуст, выводится сообщение о том, что словарь пуст.

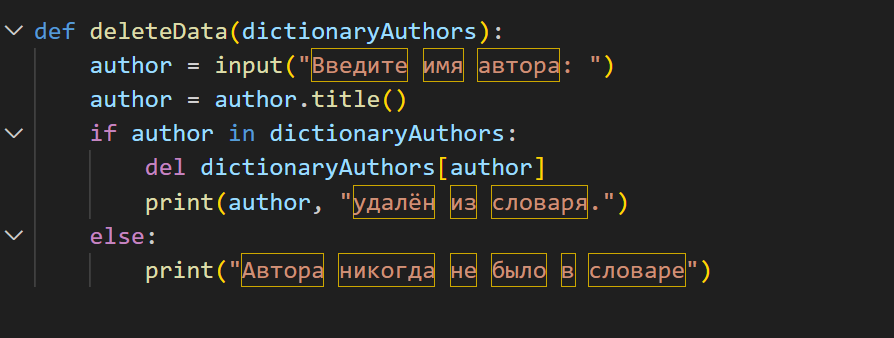


Рис. 10

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

author = input("Введите имя автора: "): Запрашивает у пользователя ввод имени автора.

author = author.title(): Преобразует введенное имя автора так, чтобы первая буква каждого слова была заглавной, обеспечивая единообразие в данных.

if author in dictionaryAuthors:: Проверяет, существует ли введенный автор в словаре.

del dictionaryAuthors[author]: Если автор найден в словаре, его удаляют с помощью оператора del.

print(author, "удалён из словаря."): Выводит сообщение о том, что автор успешно удален.

else:: Если введенного автора нет в словаре, выводится сообщение "Автора никогда не было в словаре".

Эта функция предоставляет возможность пользователю удалить автора из словаря и предостерегает, если введенного автора нет 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/