ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

**Ciobanu Stanislav**

**Отчет**

по дисциплине „ПРОГРАММИРОВАНИЕ В PYTHON”

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

(подпись)

Автор: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Кишинев, 2024

**Содержание:**

1. Задача
2. Описание Алгоритмов
3. Взаимодействие с пользователем
4. Описание структур данных
5. Описание функционала
6. Выводы
7. **Задача:**

Вариант 1

Необходимо создать приложение для взаимодействия со словарём авторов. В этом словаре необходимо представить имя автора, как ключ, а книги, как значения по ключу.

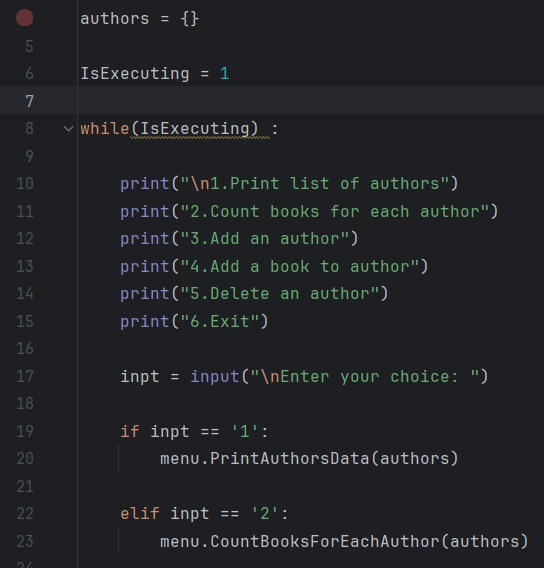
Необходимо создать меню, в котором пользователь сможет вызывать функции для взаимодействия со словарём авторов.

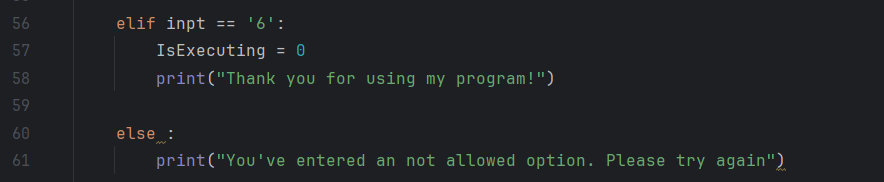
Функции должны быть импортированы из отдельного файла.

Программа должна реагировать на неправильные действия пользователя.

1. **Описание алгоритмов:**

Для представления меню используется чикл while, который будет выполняться до тех пор, пока значение переменной IsExecuting не станет 0. Внутри цикла каждый раз при помощи функции input считывается ввод пользователя и операторами if, elif, else выбирается необходимое действие.



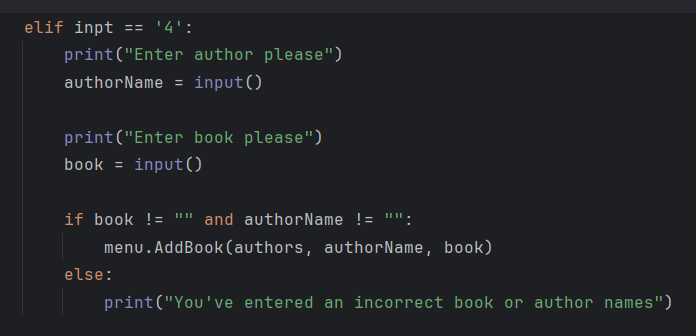


Внутри цикла в зависимости от выбора пользователя будут вызываться функции.

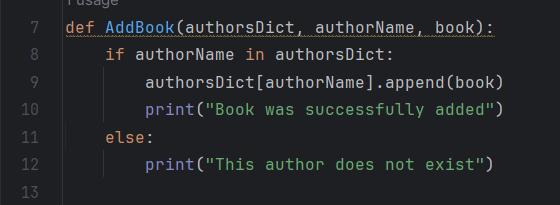
1. **Взаимодействие с пользователем:**

Тут же происходит начальная проверка дынных, введённых пользователем. Если пользователю необходимо ввести имя автора или название книги, а он вводит пустую строку, программа сообщит ему, что он ввёл неправильные данные.

Аналогично если пользователь выберет несуществующую опцию в меню, он получит об этом сообщение.

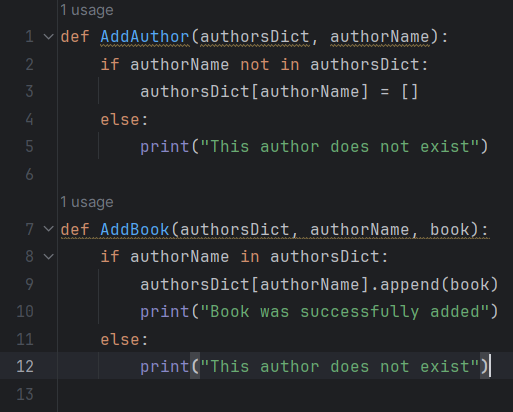


В самих функциях так-же происходит проверка данных. Если пользователь ввёл несуществующего автора, то функция выведет на экран сообщение, что такого автора нет.

****

1. **Описание структур данных:**

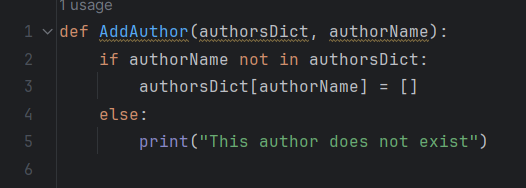
Данные хранятся в словаре. Ключ – это автор, значение – список книг. Изначально, задавая автора мы создаём ему пустой список. В последствии, чтобы добавить книгу, мы будем пользоваться методом append, который будет добавлять элемент в конец списка.



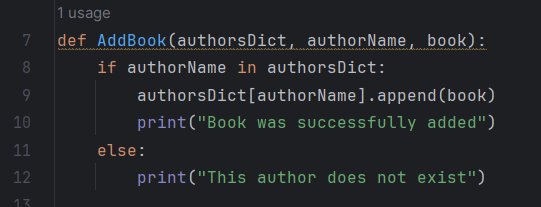
1. **Описание функционала:**

В приложении представлено 5 функций, которые позволяют взаимодействовать с авторами.

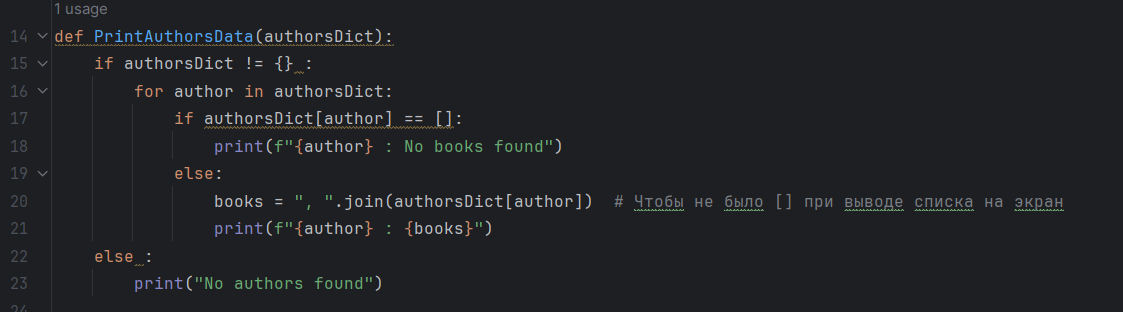
Функция AddAuthor принимает внутрь список авторов и имя автора. Она задаёт значение по ключу (имени автора) в виде пустого списка.



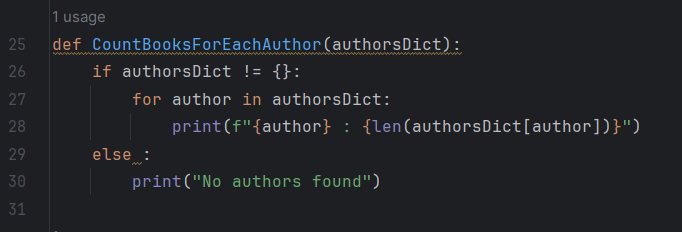
Функция AddBook принимает внутрь список авторов, имя автора и название книги. Она добавляет название книги в конец списка книг по ключу.



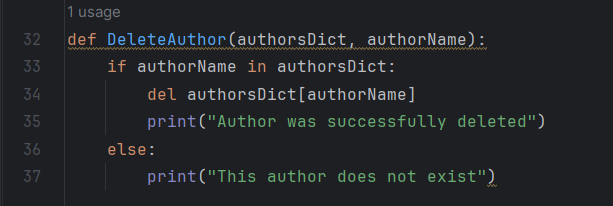
Функция PrintAuthorsData принимает внутрь только список авторов и выводит на экран имя автора и все его книги. Если книг нет, она так и напишет, что книги не были найдены.



Функция CountBooksForEachAuthor так-же принимает внутрь только список авторов и выводит на экран имя автора и количество элементов внутри списка его книг.



Функция DeleteAuthor принимает внутрь список авторов и имя автора и по ключу удаляет элемент словаря.



1. **Выводы:**

Python обладает удобными инструментами для работы с различными массивами данных. В языках C/C++/C# и др. тот же функционал пришлось бы реализовывать при помощи гораздо более массивного и плохо-читаемого кода.

Однако в Python, языке со статической типизацией, тем не менее функции могут принимать любые типы данных, что требует постоянной проверки корректности передаваемых данных. В других языках для передачи двух разных типов в одну функцию необходима была бы перегрузка функции, в Python – не обязательно. Из-за этого могут возникать непредвиденные результаты.