ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

**Михайлов Пётр**

**Отчет**

по дисциплине „ПРОГРАММИРОВАНИЕ В PYTHON”

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

(подпись)

Автор: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Кишинев, 2024

Краткая постановка задачи

Этот проект нацелен на создание небольшого приложения на Python, которое позволит управлять списком авторов и их произведений. Пользователь сможет добавлять новых авторов и книги, просматривать имеющийся список авторов и их книг, а также удалять авторов вместе со всеми их произведениями из списка.

Логика реализованных алгоритмов

Для реализации функционала приложения использовались следующие алгоритмы:

1)Добавление автора: Если введенное имя автора уже присутствует в словаре authors, выводится сообщение об ошибке. В противном случае, новый автор добавляется в словарь authors с пустым списком книг.

2)Добавление книги к существующему автору: Проверяется наличие введенного имени автора в словаре authors. Если автор существует, название книги добавляется в список книг данного автора. В противном случае, выводится сообщение об ошибке.

3)Просмотр списка авторов и их книг: Выводится список всех авторов и соответствующих им книг. Если список пуст, выводится соответствующее сообщение.

4)Подсчет количества книг у каждого автора: Выводится количество книг у каждого автора. Если список пуст, выводится соответствующее сообщение.

5)Удаление автора и его книг: Если введенное имя автора существует в словаре authors, данный автор и все его книги удаляются из словаря. В противном случае, выводится сообщение об ошибке.

Код:

def add\_writer(writers, name):

    if name in writers:

        print("Этот автор уже добавлен.")

    else:

        writers[name] = []

        print(f"Автор {name} добавлен.")

    return writers

def add\_book\_to\_writer(writers, name, title):

    if name not in writers:

        print("Ошибка! Такого автора нет в списке!")

    elif title in writers[name]:

        print(f"Ошибка! Книга '{title}' уже есть у автора: {name}.")

    else:

        writers[name].append(title)

        print(f"Книга '{title}' добавлена автору: {name}.")

    return writers

def view\_writers(writers):

    if not writers:

        print("Список авторов пуст.")

    else:

        print("Список авторов и их книг:")

        for writer, books in writers.items():

            print(f"{writer}: {', '.join(books)}")

def count\_books\_of\_writers(writers):

    if not writers:

        print("Список авторов пуст.")

    else:

        print("Количество книг у каждого автора:")

        for writer, books in writers.items():

            print(f"{writer}: {len(books)}")

def remove\_writer(writers, name):

    if name not in writers:

        print("Такого автора нет в списке.")

    else:

        del writers[name]

        print(f"Автор {name} и все его книги успешно удалены.")

    return writers

В отдельном файле я создаю пять функций, которые будут использоваться в моей маленькой программе.

Первая функция add\_writer предназначена для добавления нового автора в список. Когда я вызываю эту функцию, программа проверяет, существует ли уже автор с таким именем. Если да, мне сообщается, что такой автор уже есть. Если нет, программа добавляет нового автора в список.

Вторая функция, add\_book\_to\_writer, предназначена для добавления книги к указанному автору. Программа сначала проверяет, существует ли такой автор в списке. Если да, она проверяет, есть ли уже такая книга у этого автора. Если книга уже существует, мне выводится сообщение об этом. В противном случае, книга добавляется к списку книг этого автора.

Третья функция, view\_writers, предназначена для вывода списка всех авторов и их книг. Если список пуст, мне сообщается, что список авторов пуст.

Четвертая функция, count\_books\_of\_writers, предназначена для вывода количества книг у каждого автора. Если список пуст, мне также сообщается, что список авторов пуст.

Пятая функция, remove\_writer предназначена для удаления указанного автора из списка, а также всех его книг.

from functions import \*

def main():

    writers = {}

    while True:

        print("\nЧего желаете?:")

        print("1. Добавить автора")

        print("2. Добавить книгу к существующему автору")

        print("3. Просмотреть список авторов и их книг")

        print("4. Вывод количества книг у авторов")

        print("5. Удаление автора и его книг")

        print("6. Выход")

        choice = input("Выберите действие: ")

        if choice == "1":

            name = input("Введите имя автора: ")

            writers = add\_writer(writers, name)

        elif choice == "2":

            name = input("Введите имя существующего автора: ")

            if name not in writers:

                print("Ошибка! Такого автора нет в списке!")

                continue

            title = input("Введите название книги: ")

            writers = add\_book\_to\_writer(writers, name, title)

        elif choice == "3":

            view\_writers(writers)

        elif choice == "4":

            count\_books\_of\_writers(writers)

        elif choice == "5":

            name = input("Введите имя автора, которого хотите удалить: ")

            writers = remove\_writer(writers, name)

        elif choice == "6":

            print("Программа завершена.")

            break

        else:

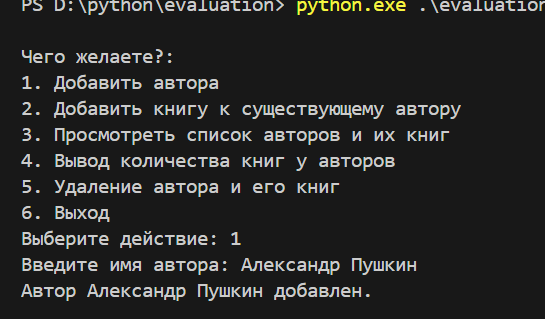
            print("Ошибка!!! Неверный ввод.")

main()

Здесь представлен мэйн. Для начала я импортирую функции из отдельного фалйа, которые были представлены выше. С помощью цикла while создается меню, которое предоставляет автору выбор. По заданию были заданы следующие пункты меню : добавление автора, добавление книги к существующему автору, просмотр списка авторов и их книг, вывод количества книг у авторов, удаление автора и его книг и выход из программы. Далее пользователь может выбрать нужный ему пункт и программа будет работать с тем, чем он выбрал. Были использованы операторы if , elif, else, которые помогают программе обрабатывать различные варианты, в зависимости выбора пользователя. Если пользователь введет все правильно, автора, которого нет в списке и книги, то все данные успешно запишутся. Также там есть и вывод ошибок, например, если пользователь введет книгу с одним и тем же названием у одного автора, выведется ошибка.

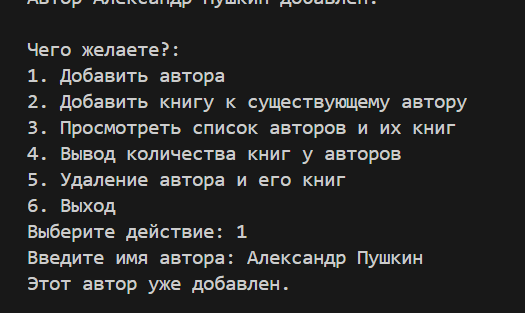
Скриншоты из терминала:

Первый пункт из меню:

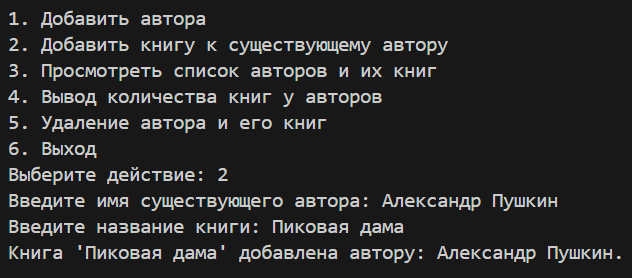


Как мы видим, автор добавился без никаких проблем.

При вводе того же автора, программа выводит на экран, что такой автор уже существует

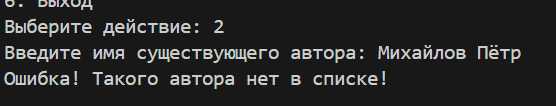


Второй пункт из меню:

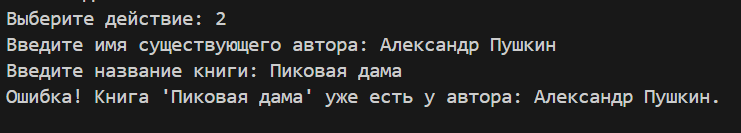


При вводе существующего автора и книги, никаких ошибок не возникло.

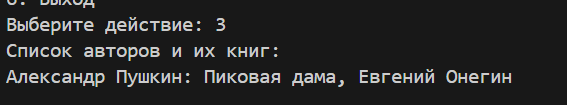
Но при вводе не существующего автора, выводится ошибка



Также при вводе такой же книги, которая уже существует, тоже будет ошибка:

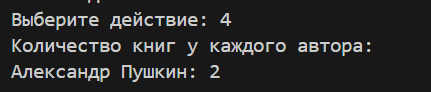


Третий пункт из меню:



Как мы можем видеть, при выборе третьего пункта в терминал выводится список авторов и их книг, введенные пользователем.

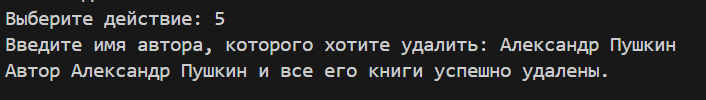
Четвертый пункт из меню:



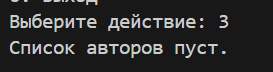
При выборе четвертого пункта, в терминал выводится автор и его количество книг, которые были введены пользователем.

Пятый пункт из меню:

Пятый пункт удаляет автора и его книги из списка.

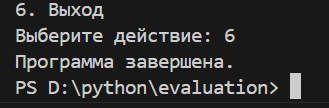


Выбрав третий пункт, который показывает список авторов и книг, в терминале будет выведено, что список пуст и никаких данных там нет.



Шестой пункт из меню:

Шестой пункт простой, он позволяет нам выйти из программы.



Структуры данных, с которыми работали в приложении

В данном приложении используются следующие структуры данных:

1. Словарь : словарь writers используется для хранения информации об авторах и их книгах. Ключами словаря являются имена авторов, а значениями - списки их книг.
2. Список: в значениях словаря writers хранятся списки книг для каждого автора.

Заключение:

Python оказался отличным выбором для разработки данного мини-приложения благодаря своей простоте и мощным встроенным структурам данных, таким как словари. Язык Python позволил легко реализовать весь необходимый функционал и обеспечить удобное взаимодействие с пользователем. Несмотря на то, что я привык к синтаксису C++ и при написании данной программы встречались моменты, когда я путал синтаксис Python и С++, все равно данный язык программирования мне очень понравился в написании и если сравнивать с С++, то я могу сказать, что если при выборе, где легче написать такое мини-приложение, то без сомнений я бы сказал, что в Python легче сделать это.