**Софийски университет „Св. Климент Охридски“**

Факултет по математика и информатика

Специалност „**Компютърни науки**“

1-ви курс, група 2

ООП, летен семестър (2021/2022 г.)

Проект на тема „**Електронна библиотека**“

Изготвен от: Йоанна Любомирова Пешева (Факултетен номер: **2MI0800103**)

Връзка към хранилището в GitHub:

*https://github.com/YoannaPesheva/ElectronicLibrary*

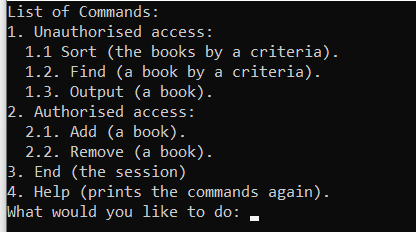
**Съдържание:**

1. Анализ на поставената задача и подход на решаване
2. Как работи програмата?
3. Кратко описание на използваните класове и алгоритми
4. Идеи за бъдещи подобрения
5. Използвани източници
6. **Анализ на поставената задача и подход на решаване**

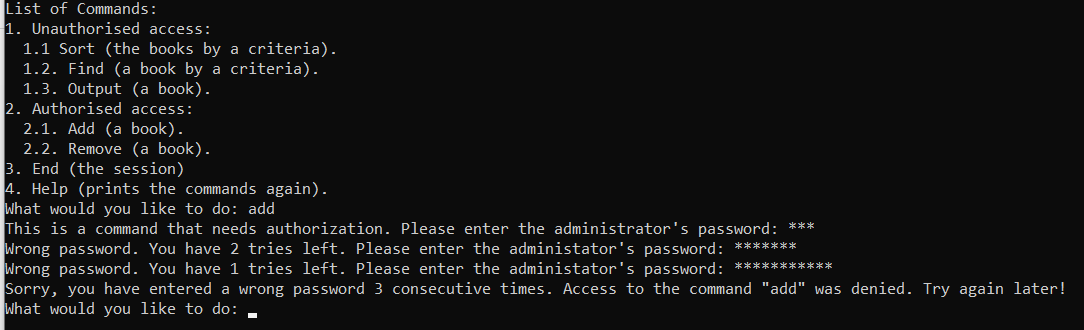
Проектът „Електронна библиотека“ цели изграждането на програма, която приема информация от потребителя, изпълнява набор от команди и запазва промените във файл. Програмата се изпълнява в диалогов режим, като поддържа две нива на достъп – свободен и оторизиран (изисква парола, за да може да се изпълни). Идеята ми за осъществяване на проекта беше да се създаде динамичен масив от обекти, в който, при отваряне на програмата, да се записват обектите от тип „книга“. Върху него също да се изпълняват и извиканите команди, а накрая, когато потребителят прекрати програма, изписвайки ключова дума, промените във файла да бъдат запазени.

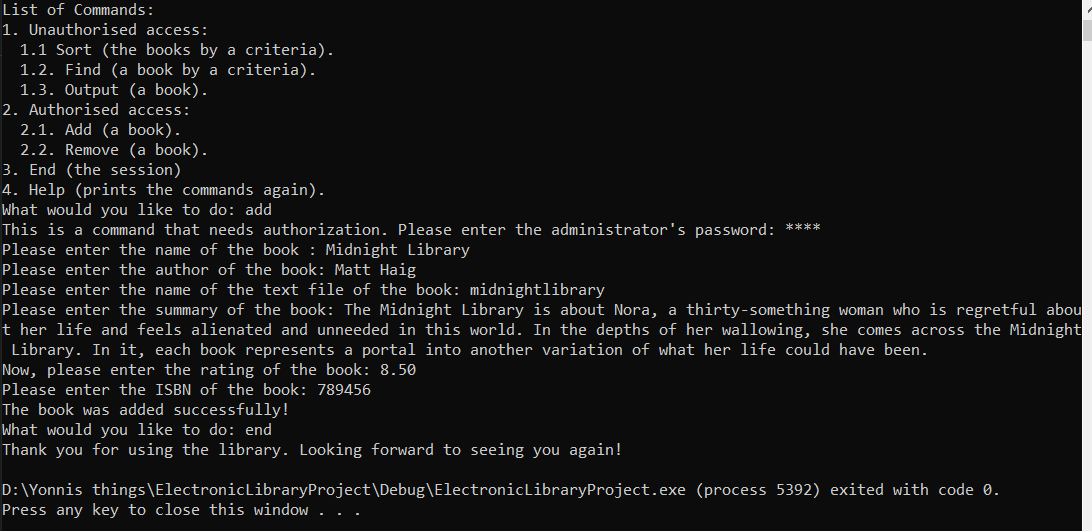
1. **Как работи програмата?**

При стартиране на програмата, на конзолата се появява списък с команди, които програмата може да изпълни, както и запитване към потребителя, от когото се изисква да въведе ключовата дума на командата, която иска да бъде изпълнена (Приложение №1).

*Приложение №1*

Потребителят въвежда някоя от командите, като може да я въведе по два начина – първата буква може да бъде и главна, и малка (тоест, ако искаме да се изпълни командата „*add*“, тогава можем да я изпишем и по този начин: „*Add*“). При изписана дума, която не отговаря на ключовата за някоя от тези команди, на екрана се изписва съответното съобщение за невалидна команда, след което се дава възможност на потребителя отново да въведе ключова дума. В списъка може да се проследи и кои команди изискват парола. Командите за добавяне и премахване не са със свободен достъп и ще поискат парола при опит за изпълняването им. Потребителят има 3 възможности за въвеждане на паролата. В случай на грешна парола за трети път, програмата прекратява изпълняването на съответната команда, уведомява потребителя и изисква отново въвеждането на команда (приложение №2).

 *Приложение №2*

При добавяне на нова книга, от потребителя се изисква да въведе цялата информация за книгата (в случай на въвеждане на заглавие на книга, която вече съществува в системата, програмата изписва съответното уведомително съобщение и изисква друго заглавие на книга), а при премахване – само заглавието на книгата, като може да бъде избрано дали файлът на съответната книга също да бъде изтрит. Командата за сортиране извежда целия набор от книги (изписват се само заглавие, автор и код за всяка книга), сортирани по едно от полетата: заглавие, автор или рейтинг, както и дали сортирането да е низходящо или възходящо. При командата за намиране на книга се изисква от потребителя да избере по какъв критерий (заглавие, автор, код или част от описанието) да се потърси книгата. Програмата извежда книгата, ако е намерена, а в противен случай – съответното уведомително съобщение. Функцията извеждане на книга извежда на конзолата текста на файла на книга по определен брой редове или до достигането на препинателен знак. Функцията„завърши“ прекратява програмата и запазва промените в главния файл, където се пази информацията за всички въведени книги (Приложение №3), а функцията „*help*“ принтира командите отново*.*

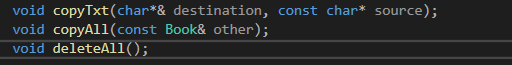
*Приложение №3*

1. **Кратко описание на използваните класове**

Класовете, използвани за реализиране на програмата, са:

1. *Book*

1.1 При модификатор за достъп *„Private“*

* Има 6 полета – 4 динамични масива от тип char за заглавие, автор, име на текстов файл и описание на съответната книга, едно поле от тип *„double“,* което служи за рейтинга на книгата, както и поле от тип *„int“* за кода на книгата.
* Има също три член-функции, които помагат при писането на голямата четворка (която ни е нужна, понеже използваме динамична памет) - Приложение №4.

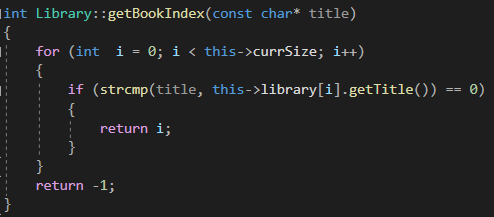
*Приложение №4*

* 1. При модификатор за достъп *„Public“*
* Канонично представяне(голяма четворка). Състои се от конструктор по подразбиране, копи конструктор, конструктор = и деструктор.
* Селектори(*get*) и мутатори(*set*).
* Метод за принтиране на полетата.

1. *Library*

2.1 При модификатор за достъп *„Private“*

* 3 полета – динамичен масив от тип *„Book“,* капацитетът на масива и неговия настоящ размер.
* Има 3 член функции – за разпределяне на повече памет за масива, помощна функция за каноничното представяне, както и за намиране на индекса на книга, която връща индекс на книга, ако заглавието й съвпада с подаденото като параметър такова и -1, ако няма книга с такова заглавие в библиотеката (приложение №5).

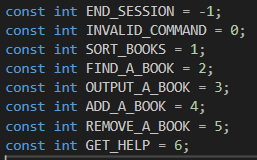
**

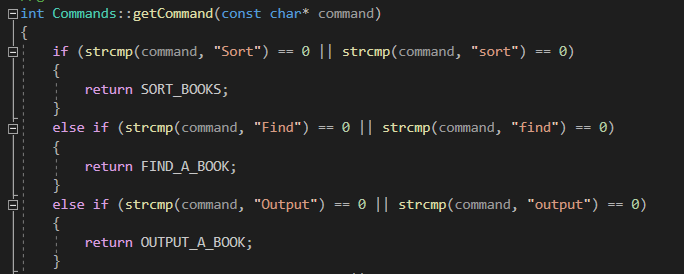
*Приложение №5*

* 1. При модификатор за достъп *„Public“*
* Канонично представяне(голяма четворка). Състои се от конструктор по подразбиране, копи конструктор, конструктор = и деструктор.
* Предефиниране на оператор [].
* Селектори за взимане на текущия размер на масива от обекти и индексът на книгата, получен от функцията в приложение №5
* Методи за добавяне и премахване на книга към масива от книги
* Методи за сортиране на книгите в масива по различните критерии
* Метод за принтиране на масива от книги по специален критерий – да бъде само заглавието, авторът и кодът на всяка книга, както и метод за изписване на масива във файла.

1. *Commands*

Класът е помощен и има два метода при модификатора за достъп *„Public“.* Първият метод служи за принтиране на командите на конзолата, а вторият за разпознаване на командата, подадена от потребителя (за целта са използвани константи от тип *„int“,* като всяка е число, което определя командата – приложение №6 и №7). Функциите са „static“, защото нямаме никакви полета в класа.

* Приложение №6*

**

*Приложение №7*

1. *CommandExecuter*

4.1 При модификатор за достъп *„Private“*

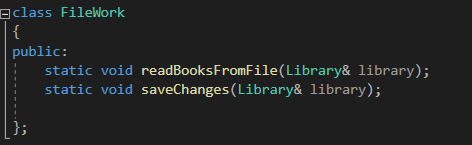
* Има едно поле от тип *„Library“*
  1. При модификатор за достъп *„Public“*

Тук се извършва изпълнението на командите, подадени от потребителя

* Метод за стартиране на програмата.
* Методи за невалидна команда, сортиране, намиране, извеждане, добавяне и премахване на книга, както и за прекратяване на програмата.

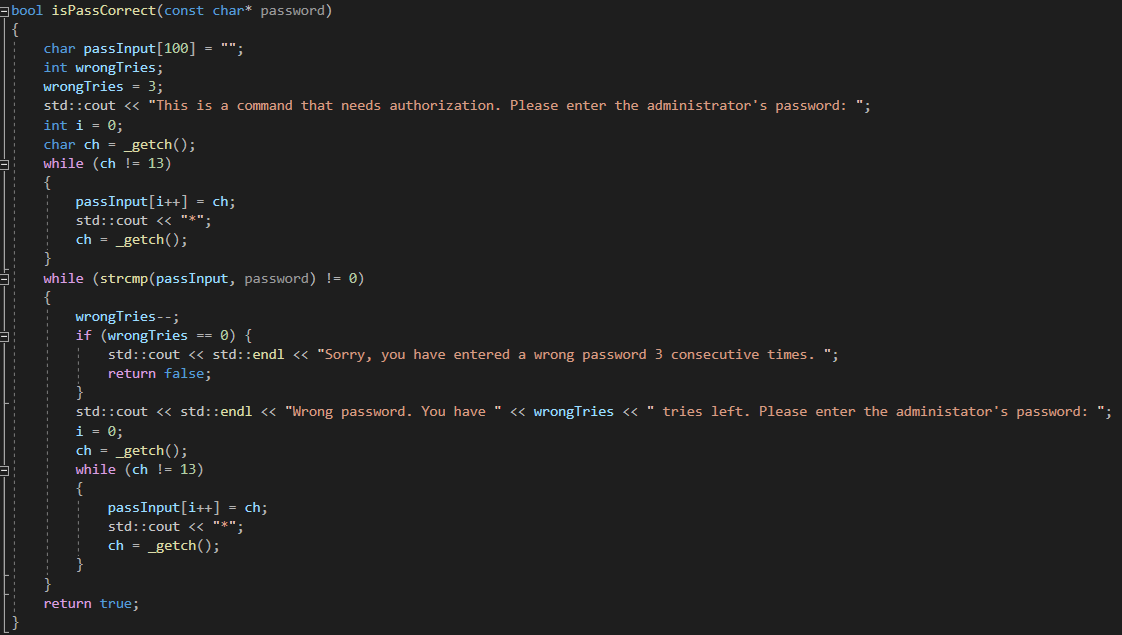
1. *FileWork*

Помощен клас за работа с файловете. Както при класа *„Commands“,* нямаме полета, само две функции, които са при модификатора за достъп *„Public“.* Първата служи за прочитане на данните за книги, които са били добавяни преди, а втората – за да запази промените във файла след приключване на работата на програмата (приложение №8).

* Приложение №8*

Допълнение:

Има две помощни функции – за валидиране на въведена парола и за проверка дали два низа са еднакви, като не взима под внимание дали символите са главни или малки букви (приложение №9 и №10 ). За изпълнение на командата за сортиране е използван методът на мехурчетата.



*Приложение №9*

Във функцията за валидиране като функционалност е направено така, че да се появяват „\*“ вместо реалните символи, въвеждани от клавиатурата. Поради тази причина трябва да се включи и библиотеката conio.h.

*Приложение №10*

**

1. **Идеи за бъдещи подобрения**

Проектът би могъл да се разшири така, че потребителите да могат да се регистрират и да им бъде позволено да използват библиотеката, но само когато са в системата. Също така може да има място (файл), където потребителите да могат да оставят своите коментари и да има команда, с която всеки човек ще може да прочете мненията на други читатели за съответната книга. Друга идея е човек да може да добавя собствена оценка за дадена книга и тя да се прибавя към общата оценка за съответното четиво.

1. **Използвани източници**
2. <https://github.com/triffon>

* <https://github.com/triffon/oop-2021-22/tree/main/practicum/4/Week03-ClassesAndRuleOf0-3/lab> - за функцията *getBookIndex()*

1. За идеята за извеждането на паролата като „\*“: <http://www.cplusplus.com/forum/general/3570/>
2. <https://stackoverflow.com/questions/2390912/checking-for-an-empty-file-in-c> - за ред 27, *FileWork.h*