Курсов проект

БИБЛИОТЕКА

Автор:

Веселин Василев Живков

Факултетен номер: 72103

Телефон: +359888801246

E-mail: jivkovvesselin@gmail.com

Университет: СУ „Св. Климент Охридски“

Курс: 1

Група: 4

1. Цели и целева група

Линк към проекта в Github: https://github.com/GritzMaze/Library.git

1.1. Цели

- Представянето на информация на дадена книга по-удобен и лесен за четене начин
- Да поддържа база от данни със книги и потребители
- Да извършва операции за намиране, сортиране, добавяне, изтриване на книги и потребители
- Четене и записване на информацията във/от файл
- Изграждането на подходяща структура, която да поддържа наличните книги и потребителите, които да извършват операции с тях

1.2. Целева група

- Студенти
- Ученици
- Академични лица

2. Основни етапи:

2.1. Проучване на сложността на даденото задание

2.2. Определяне на основните елементи от структурата на програмата и техните функции

2.3. Планиране на графичното оформление и композиция

2.4. Избиране на методите, технологиите и библиотеките, чрез които да се постигнат поставените цели

2.5. Създаване на библиотеката

2.6. Събиране на информация за книги и попълване на базата данни

2.7. Тестване

2.8. Оценка на постигнатото

2.9. При наличие на трудности при използването на програмата, тя ще бъде редактирана и тествана повторно

2.10. Финално публикуване

3. Стартиране на проекта

3.1. Изисквания за стартиране:

- CMake
- C++11 или по-висока версия на c++
- C++ компилатор (i.e. MingW)

3.2. Стартиране на проект със CMake - How to Build a CMake-Based Project (preshing.com)

4. Логическо и функционално описание

4.1. Програмата е с двуслойна архитектура:

4.1.1. Презентационен слой – чрез класът Draw данните от програмата се представят на крайния потребител

4.1.2. Данните се съхраняват във файлове със разширения .books и .users. Книгите се съхраняват в .books, а потребителите в .users

4.2. Модули – програмата е разделена на два модула – представителен(потребителски) и административен

4.3. Функции на модулите

4.3.1. *Представителен* – представителният модул е видим за всички потребителите на програмата.

4.3.2. *Административен* – видим само за регистрирани потребители с администраторски права.

Чрез него се добавят/изтриват потребители в програмата и се запазва съдържанието на програмата.

5. Реализация

Използва се Обектно-ориентирано програмиране на C++ за реализацията на проекта.

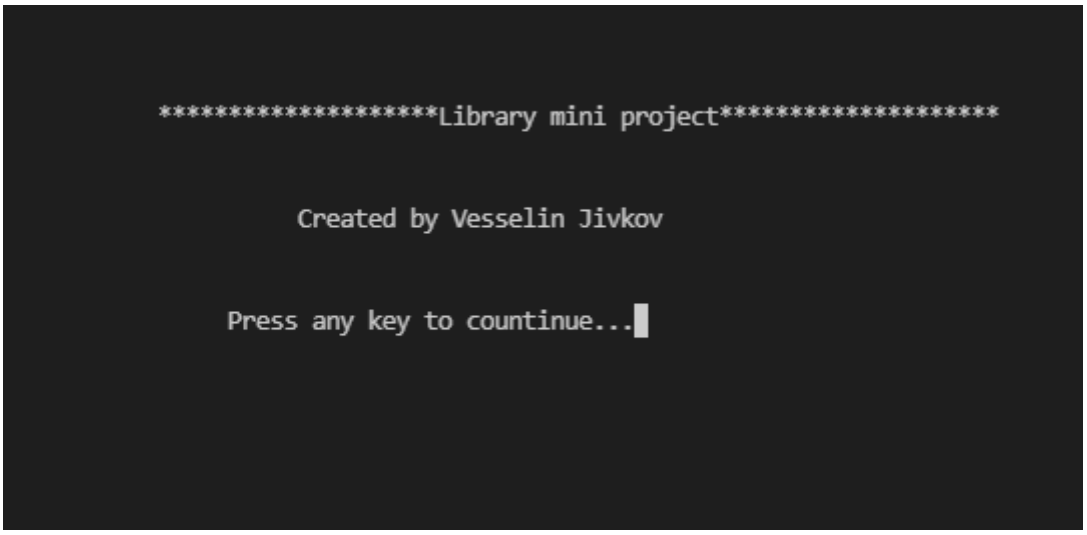
За интерфейса са използвани биоблиотеките “windows” и “conio”.

Работата със низове е постигната чрез ръчно написан String. Програмата използва и ръчно написан вектор.

За управление на проекта и конструиране на проекта е използвано CMake.

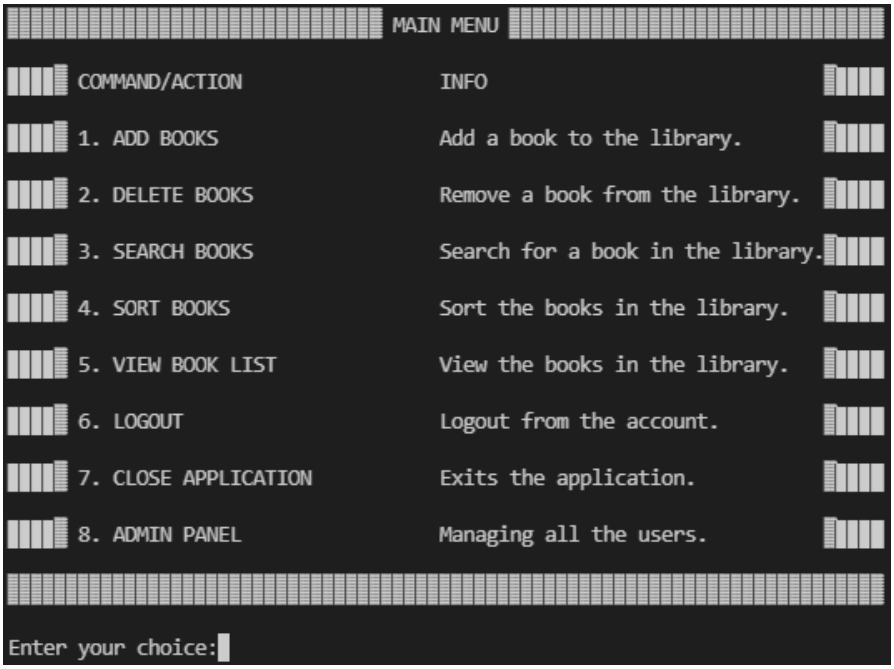
6. Описание

6.1. Начален екран при стартиране на програмата:



6.2. Главно меню

- *Добавяне на книга* – добавяне на книга в системата;
- *Изтриване на книга* – изтриване на книга от системата;
- *Търсене на книга* – Търсене на книга по име/автор/ключова дума;
- *Сортиране на книга* – Сортира книгите по име/автор/година/рейтинг;
- *Списък с книги* – Показва списък с наличните книги в системата;
- *Вход/Изход* – Дава възможност на потребителя да влезе в акаунта си;
- *Затваряне на програмата* – Затваря програмата;
- *Администраторски панел* – Само потребителите с администраторски достъп виждат и могат да достъпят това меню;

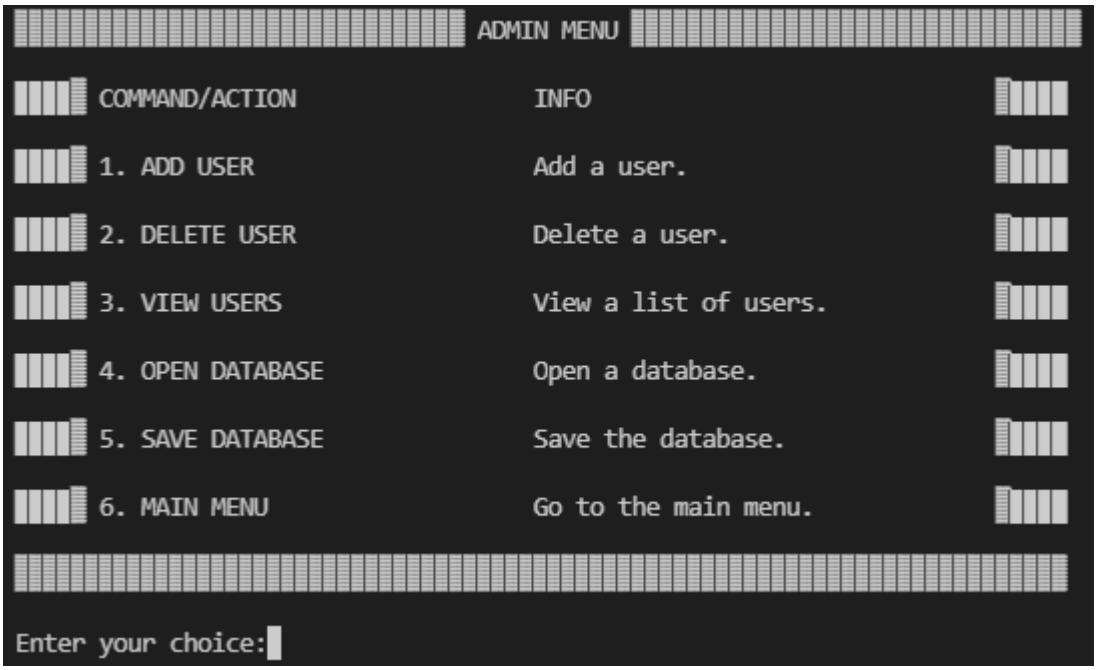


6.3. потребителите в програмата, както и за запазването на съдържанието от програмата в файлове.

При стартирането на програмата за първи път, по подразбиране се създава потребител с потребителско име – **admin** и парола – **i<3c++.** Потребителят е администратор.

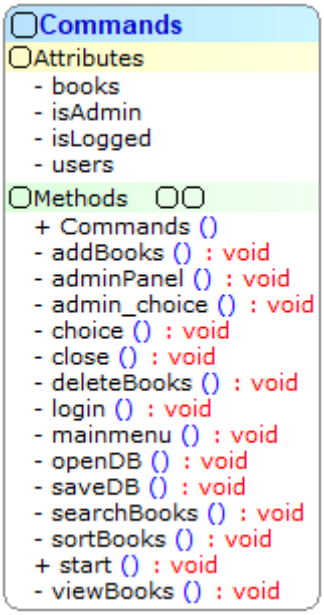
6.3.1. Администраторско меню

- *Добавяне на потребител* – добавя потребител в системата
- *Изтриване на потребител* – изтрива потребител от системата
- *Списък с потребители* – извежда списък списък с наличните потребители
- *Отваряне на база данни* – отваря база данни с книги и потребители
- *Запазване на база данни* – запазва базата данни на компютъра
- *Главно меню* – връща обратно към главното меню

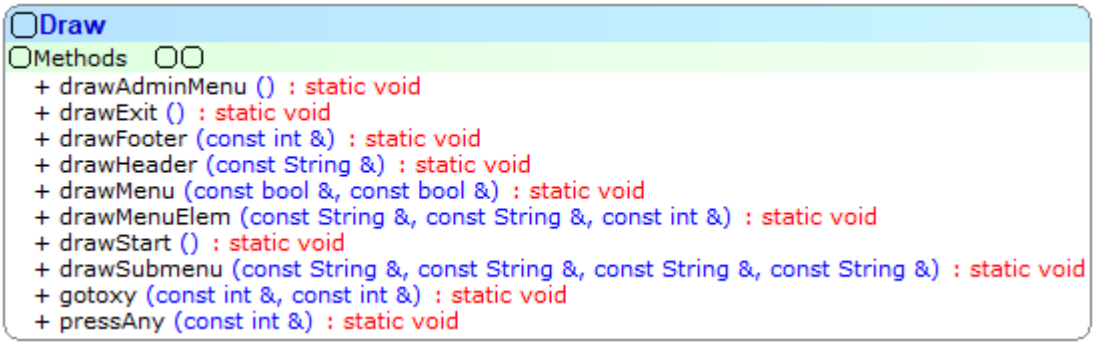


7. Описание на класовете

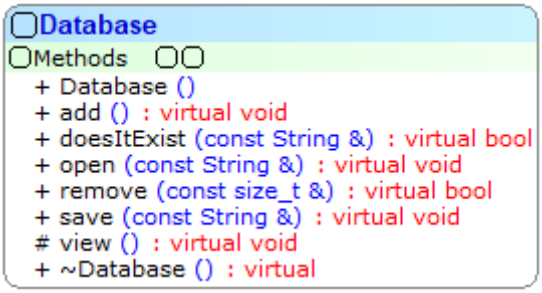
7.1. Клас *Commands* – Сърцето на програмата. Съдържа основната информация за това дали потребител е влязъл в системата и дали е администратор. Разпределя задачите по класове.



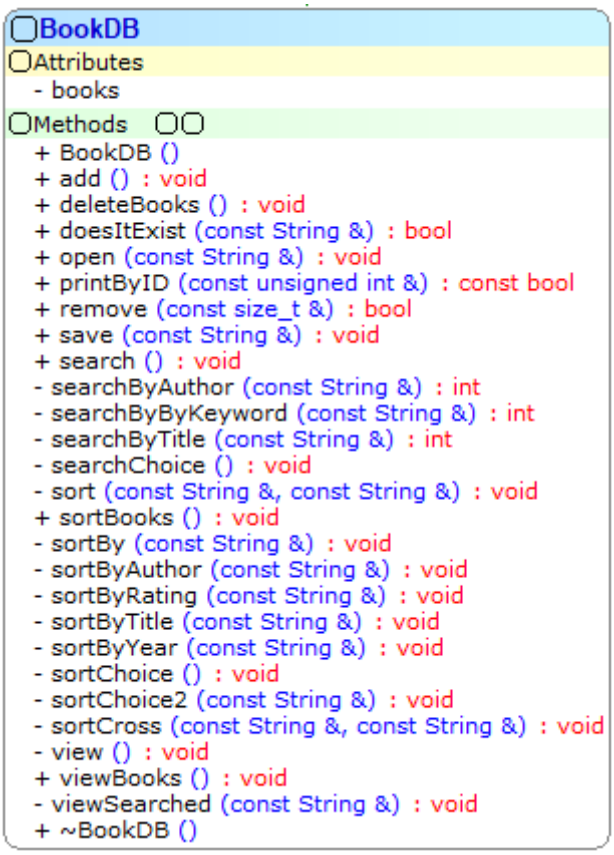
7.2. Клас *Draw* – Грижи се за графичната част от приложението. Чертае рамките, таблиците и менютата. Позиционира информацията в конзолата.



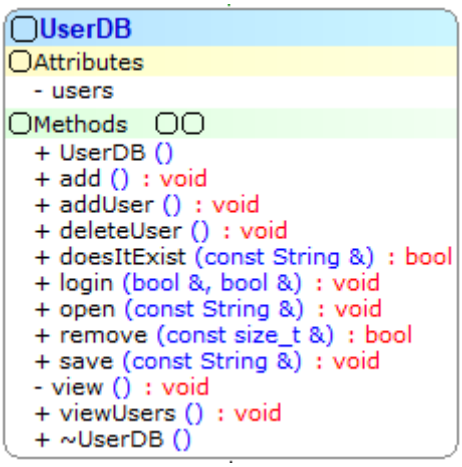
7.3. Клас *Database* – Интерфейсен клас за базата данни от потребители и книги.



7.4. Клас *BookDB* – Наследява Database. Грижи за запазените книги в системата и тяхната обработка.

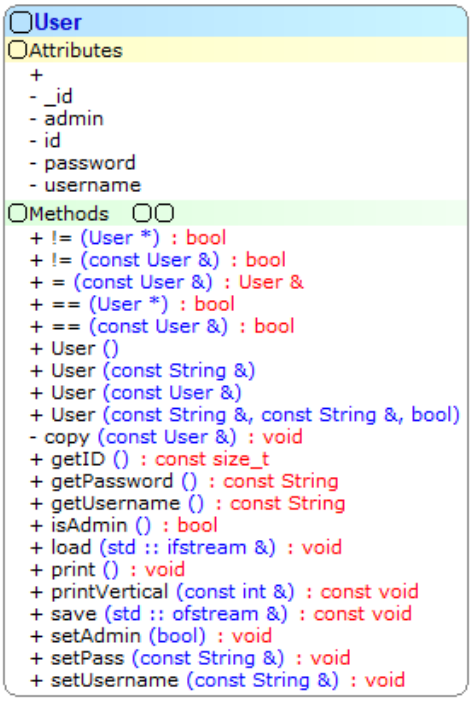


7.5. Клас *UserDB* – Наследява Database. Грижи се за запазените потребители в системата и тяхната обработка.

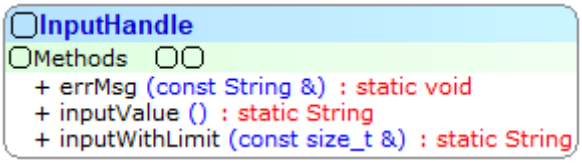


7.6. Клас *Book* – Описва дадена книга. Грижи се за обработката на единична книга.

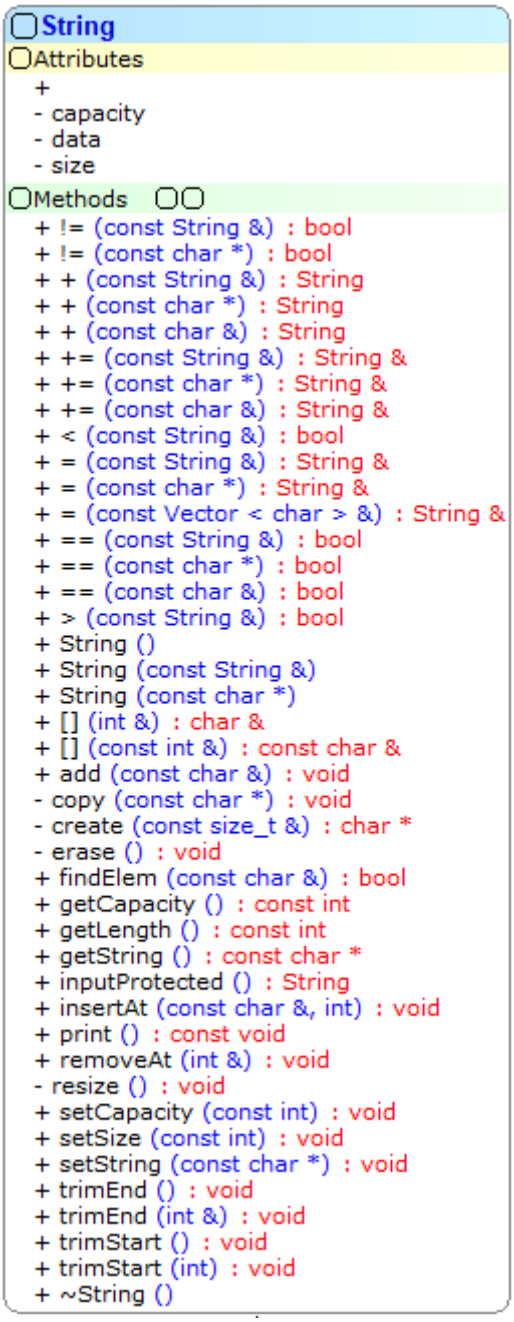
7.7. Клас *User* – Описва даден потребител. Грижи се за обработката на единичен потребител.



7.8. Клас *InputHandle* – Занимава се с извеждането на грешките в конзолата по добър за четене начин.



7.9. Клас *String* – Използва се за представяне на информацията във формата на текстов вид.



7.10. Клас *Vector* – Едномерен масив за съхранение на информация.