

Библиотека

Github Link: https://github.com/IrinaG24/Project10_Library

1. Увод

1.1. Описание и идея на проекта

Проектът „Библиотека“ реализира информационна система, която би трябвало да поддържа една библиотека. Програмата извършва определен брой действия спрямо наличните книги и потребители. Информацията за книгите и потребителите може да се извлича от файлове.

1.2. Цели и задачи

За пълноценно развитие на информационната система следните цели и задачи трябва да бъдат изпълнени:

1.2.1. Представяне на информацията за дадена книга и даден потребител по подходящ начин

1.2.2. Изграждане на подходяща структура за библиотеката, която да поддържа наличните книги и промените, които могат да се извършват спрямо тях, както и регистрираните потребители

1.2.3. Прочитане на данни от файлове и записване на направените промени във файл съответно за книгите и потребителите

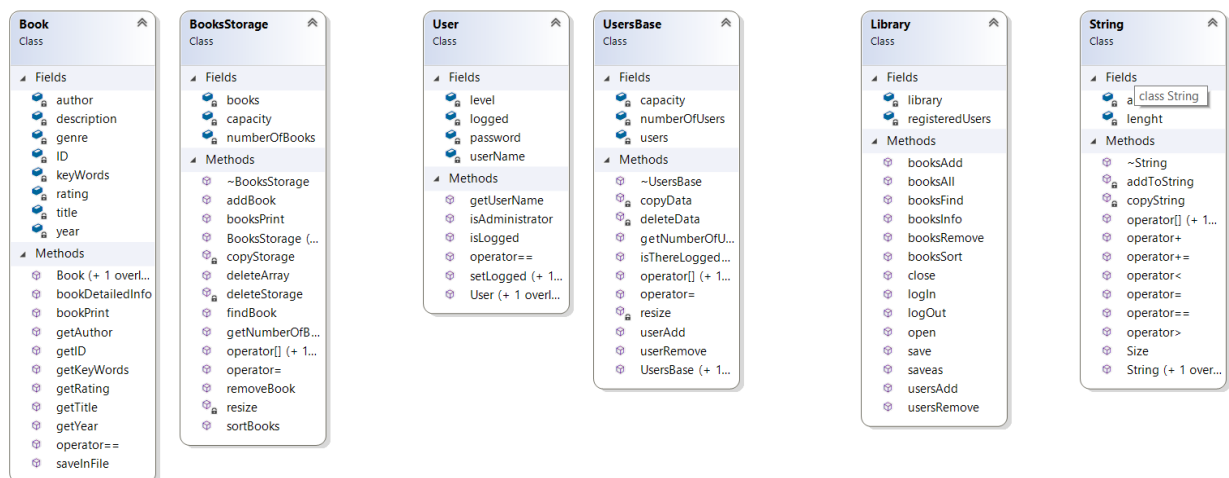
1.3 Структура на документацията

1.3.1. Увод

1.3.2. Проектиране

1.3.3. Реализация

2. Проектиране



Фигура 1

За реализацията на проекта са използвани следните класове:

❖ class String

Има следните член-данни:

- char* arr – динамичен масив от символи
- int lenght – дължина на масива

Има следните private методи:

- void copyString(const String&) – копира информацията за стринга
- void addToString(const char) – добавя елемент към стринга

Има следните методи:

- String(const char* = "") - конструктор
- String(const String&) – копи-конструктор
- String& operator=(const String&) – оператор =
- ~String() - деструктор
- int Size()const – връща дължината на масива
- bool operator==(const String&) const – оператор ==
- char& operator[](const int) – оператор []
- const char& operator[](const int)const -оператор[]
- String operator+(const String&) – оператор +
- String& operator+=(const String&) – оператор +=
- bool operator>(const String&) const – оператор >
- bool operator<(const String&) const – оператор <
- friend std::istream& operator>>(std::istream&, String&) – оператор за въвеждане >>
- friend std::ostream& operator<<(std::ostream&, const String&) – оператор за извеждане <<
- friend void getline(std::istream&, String&) – оператор за въвеждане getline с финален символ нов ред по подразбиране
- friend void getline(std::istream&, String&, char) – оператор за въвеждане getline

❖ class Book – класа съдържа нужната информация за дадена книга.

Има следните член-данни:

- String author – име на автора
- String title – заглавие на книгата
- String genre - жанр
- String description - описание
- int year – година на издаване
- String keyWords – ключови думи
- int rating - рейтинг

- `int ID` – уникален номер

Има следните методи:

- `Book()` – конструктор по подразбиране
- `Book(String, String, String, String, int, String, int, int)` – конструктор с параметри
- `void bookDetailedInfo() const` – извежда подробна информация за книгата
- `void bookPrint() const` – извежда основна информация за книгата
- `String getTitle()const` – връща името на книгата
- `String getAuthor()const` – връща автора на книгата
- `int getYear()const` – връща годината на издаване на книгата
- `int getRating()const` – връща рейтинга на книгата
- `int getID()const` – връща уникалния номер на книгата
- `String getKeyWords()const` – връща ключовите думи на книгата
- `bool operator==(const Book&)` – оператор за сравнение на две книги
- `friend std::istream& operator>>(std::istream&, Book&)` – оператор за въвеждане на книга
- `friend std::ostream& operator<<(std::ostream&, Book&)` – оператор за извеждане на книга
- `friend void saveFromFile(std::istream&, Book&)` – оператор за записване информацията за книга от файл
- `void saveInFile (std::ostream&)` – оператор за записване на информацията за книга във файл

❖ `BooksStorage` – функционира като динамичен масив от книги

Има следните член-данни:

- `Book* books` – динамичен масив от книги
- `int capacity` – капацитент на масива
- `int numberOfBooks` – броя на книги в масива

Има следните помощни `private` методи:

- `void copyStorage(const BooksStorage&)` – копира данните за масива от книги
- `void deleteStorage()` – изтрива динамично заделената памет
- `void resize()` – удвоява капацитета на масива

Има следните методи:

- `BooksStorage(int = 1)` – конструктор
- `BooksStorage(const BooksStorage&)` – копи-конструктор
- `BooksStorage& operator=(const BooksStorage&)` – оператор =
- `~BooksStorage()` – деструктор
- `void booksPrint() const` – извежда основна информация за всички книги
- `void findBook(String, String) const` – намира и извежда подробна информация за книга. Търсенето се извършва по заглавие, автор или ключова дума

- `void sortBooks(String)` – сортира книгите в масив във възходящ/низходящ ред спрямо заглавието, автора, рейтинга и годината на издаване
- `void addBook(const Book&)` – добавя книга в масива
- `void removeBook(const Book&)` – изтрива книга от масива
- `int getNumberOfBooks()const` – връща дължината на масива(броя на книгите в него)
- `Book& operator[](const int)` – оператор [], позволяващ редактирането на книга на дадена позиция
- `const Book& operator[](const int)const` – оператор [], даващ само достъп до книга на дадена позиция
- `void deleteArray()` – изтрива записаната информация в масива

❖ **User** – съдържа информация за потребител

Има следните член-данни:

- `String username` – потребителско име
- `String password` - парола
- `Int level` – 1 за потребител и 2 за админ
- `Bool logged` - казва дали потребителя е влязъл в профила си

Има следните методи:

- `User()` – конструктор по подразбиране
- `User(String, String, int)` – конструктор с параметри
- `bool operator==(const User&) const` – оператор ==
- `bool isLoggedIn()const` – проверява дали потребителя е влязъл в профила си
- `bool isAdmin()const` – проверява дали е администратор или не
- `void setLogged(bool)` – променя член-данната `logged` при влизане и излизане на потребител от профила си
- `void setLevel(int l)` – задава нивото на достъп
- `String getUsername()const` – връща потребителското име на потребителя
- `friend void saveFromFile(std::istream&, User&)` – запазва информацията за потребителя от файл
- `friend void saveToFile(std::ostream&, const User&)` - запазва информацията за потребителя във файл

❖ **UsersBase** – функционира като динамичен масив за потребители

Има следните член- данни:

- `User* users` – динамичен масив от потребители
- `int capacity`- капацитет на масива
- `int numberOfUsers` – брой на потребителите в масива

Има следните private методи:

- `void copyData(const UsersBase&)` – копира информация за масива
- `void deleteData()` – изтрива динамично заделената памет
- `void resize()` – удвоява капацитета на масива

Има следните методи:

- UsersBase(int = 1) - конструктор
- UsersBase(const UsersBase&) – копи-конструктор
- UsersBase& operator=(const UsersBase&) – оператор =
- ~UsersBase() - деструктор
- void userAdd(const User&) – добавя потребител в масива и записва новия потребител във файла с потребители
- void userRemove(const User&) – изтрива потребител в масива и записва промените във файла с потребители
- int getNumberOfUsers()const – връща броя потребители в масива
- bool isThereLoggedUser()const – проверява дали има влязъл в профила си потребител
- User& operator[](const int) - – оператор [], позволяващ редактирането на потребител на дадена позиция
- const User& operator[](const int index)const - оператор [], даващ само достъп до потребител на дадена позиция

❖ Library – информационна система, представяща библиотека

Има следните член-данни:

- UsersBase registeredUsers – регистрирани потребители
- BooksStorage library – налични книги

Има следните методи:

- void logIn(const User&) – позволява на потребител да си влезе в профила
- void logOut() – вече влязъл потребител, излиза от профила си
- void booksAll()const – извежда основна информация за всички книги
- void booksInfo(int) const – извежда подробна информация за книга с определен уникален номер
- void booksFind(String, String) const – намира и извежда информация за книга според заглавие, автор или ключова дума
- void booksSort(String) - сортира книгите във възходящ/низходящ ред спрямо заглавието, автора, рейтинга и годината на издаване
- void booksAdd(const Book&) – добавя нова книга
- void booksRemove(int) – изтрива книга
- void usersAdd(const User&) – регистрира потребител
- void usersRemove(const String&) – занулява регистрацията на потребител
- void open(const char*) – отваря файл и прочита данните от него
- void save(const char*) – запазва данни в последно, отворения файл
- void saveas(const char*) – запазва данните в нов файл
- void close() – изтрива запазената информация досега

3. Реализация

- Въвеждане и четене от файл на книга: характеристиките на книгата са разделени с '|', като се чете до нов ред
- Въвеждане и четене от файл на потребител: потребителското име и паролата са на два отделни реда(във файла между всеки потребител има един ред разстояние). В началото всеки регистрирал се потребител няма статус на администратор, освен единственият, записан във файла "users.txt"
- Информацията за потребителите са съхранява в точно един файл("users.txt"), докато информацията за книгите може да бъде презаписвана в различни файлове
- Преди всяко действие спрямо книгите и потребителите се правят нужните проверки: дали е регистриран потребител и дали е администратор.