Тема 2

Задача: Да се създаде програма която поддържа складирането на книги и потребители. Потребителите се делят на два вида съответни админи и не-администрации. Обикновенните потребители имат достъп до книгите read only type. Те имат опции да извеждат книгите на база различни критерии включително да търсят на база ключови думи. Администрациите имат същите права ,но имат и опцията да добавят и премахваат книги.

Когато се влиза в програмата потребителят има няколко опции за извеждане на книгите. Той може да получи или пълен списък с книгите сортирани по даден критерии ,или да набере ключова дума с която ще се изведат само книгите ,които съдържат съответната дума. Ако се избере списък със всички книги потребителя ще му излезе опция дали да се подредят по азбучен ред спрямо заглавието или рейтинга им. Всяка една от опциите с избор дали да е отпред назад или обратнио. След като е селектирал книга потребителя има избор дали да е отвори и порчете отново по няколко различни начина

Командите които само администрациите имат са да добавят нова книга или да премахват вече съществуващща. Начинат на селектиране на книга за изтриване е на същя метод като на потребителя да чете.

В заглавното меню на администрациите им излиза опцията да добавят книга където имат избор дали да направят import от вече същесвуващ фаил ,или да въведят инфирмацията за книгата ръчно.

Програмата се дели на четири части. 3 класа и "source".

КЛАС BOOK

Първият клас (съответно първият ,който създадох) е book. В него се сладира всичката информация за една книга. В него. Всеки параметър има сетъри и гетъри.

ISBN и rating имат валидация. При При ISBN трябва подаденият string е да максимално 13 символа също дата да съдържа само цифри.

Rating трябва да е double между 0 и 5.

Всички string-ове трябва да са максимално 100 симбола дълги. Това може да се промени ако се промени статичната константа MAX_SYMBOLS_IN_PARAM в booh.h.

Ако стинг не попада в критериите става nullptr.

Ако rating е над 5 става 5 ,ако е под 0 става 0.

фукнции за четене:

По условие на задачата има няколко опции за извеждате на съдържанието на една книга.

Нормално (с функция printBook())

По изречения(printBookBySentence()) където търси определени „приключващи“ символи. Има проверка (isEndSymbol) която може да се редакрита с цел добавяне на още символи. Дефолтните са ***'.' '!' '?'*** **.** Има проверка за потенциални многоточия или други комбинации от „приключвашите“ символи. Ако се окаже че има ще изведе новата страница след края на посления символ.

По редове (printBookByPage(N)) принтира книгата с N – наброй реда на страница

По символи(printBookByNcharacters(N)) принтира N-наброй символа на страница.

Помощна функция newoageSpacer(). Оставя определен брой редове между страниците. Може да се промени от статичната константа LINES_BETWEEN_PAGES в book.h. Функцията се извиква между всеки две страници.

CreateBookInfoFile() - създава информационен txt фаил в папка data името на фаила се определя от location член данната.

RemoveFile() изтрива информационен фаил за книгата от data папката.

Възможности за подобрения: конструктор който създава книга от инфирационен фаил.

Клас USER

Той се сладират 3 член данни , username, password,isAdmin като isAdmin е булева променлива която пази дали user-а е админ.

Username и Password имат проверка да не са над 100символа. Това може да се промени с променя на статичната константа MAX_PARAMETER_LEN в user.h. Ако са над стават nullptr

Username има проверка дали няма разтояния в него. Ако има става nullptr.

Дефолтната стойност на isAdmin е false

Всички член днни имат сетъри и гетъри.

Username и password имат по една функция която по зададен sting връща дали не са идентични. Функцииите връщат true/false

chechUsername(char*)

checkPassword(char*)

MAX_PARAMETER_LEN има гетър.

creareUserFile() създава txt фаил с информацията за потребителят. Фаилът се скалдира в data/users а името му е username на потребителя.

getuserFileLocation() е гетър който връща стринг, кйто сочи автоматично към името на фаила на потребителят.

InportFile(char* location) задава стойности на user-а от неговият txt фаил. В data/users/ и и име на фаила /location/ ,

Съществува конструктор който използва importFile(char*) функцията.

КЛАС LIBRARY

Този клас съдържа множоство от книги. В него се и поддържа менюто на библиотеката.

Функция addBook(book) добавя нова книга към масивът с книги. Ако маивът надвишава лимита си се извиква функцията resize() ,която го удвоява. Началната големина на масива е 16 но може да се промени от DEFAULT_INPUT_SIZE в library.h

Фубкция importBook(char* location). Вариант на addbook където вкарваш книга иот фаил с даден location().

SwapIndex(size_t A, size_t B); функция която разменя индекса на две книги(A ,B)

rankingSort() , titleSort() , authorSort() - функции ,които сортитат книгите спрямо даденият критерии. (Сортирането става с swapIndex()). Понастоящен алгоритмите за сортирания са бавни и могат да се подобрят.

Bool isLowerWord(str ,str) сравнява два стинга по азбучният им ред. Изловзва се от titleSort() и authorSort().

Всички член данни имат гетъри.

ListBooks() извежда книгите по ред на индекса. Изполва се обикновенно непосредствено след сортиране.

getInput() функция която се извиква когато искаме да вземем integet от потребителят

getInputString() функция която се изивква когати искаме да вземем string от потребителя.

ValidInputCOnverter(input, commands) излозва се в менюто когато имаме commands на брой операции за извършване. Функцията проверява дали input<commands и връща input ако е true и -1 ако е false.

МЕНЮ

Много от функциите на менюто се взимат с параметър user за да се порверява дали има администраторски права.

menu(user) главната функция за за отваряне на menu в конзолата.

PrintListMenu() , printReadingOptions() , printBookProfileUser(),printBookProfileAdmin() , PrintCommandsUser, printCommandsAdmin().

Са помощни фукнции които едиственио принтират различините команди в различни етапи на менюто.

listMenu(user) – меню за различните опции за сортиране на книгите;

addBookMenu() - admin-only меню за добавяне на книга(с избор дали да е ръчно или с import на txt)

Функция removeBook(user) премахва книгата от смисъкът с книги и преподрежда тези с индекс след нея. При повикването на тази функция се извиква и removeFileDecision() където адмна може да избере дали да изтирие е txt документа на книгата или само да е премхне от листа.

searchMenu(user) функцията очаква input string . След това проверява за идентично съпадение на author, title и isbn. Също така и за частично на description. Ако намери съвпадения извежда само намерените книги

selectSearchedMenu(user , input) продължение на serachMenu за избиране на само list-нати книги.

SelectBookIndex(user) фукнкция за избирането на книга от list-натите след сортиране.

ReadingMenu(user) нюото за четена на книга. Към него се навръзват фукнциите readBookBySentences(). ReadBook() , readBookByCharactersIPage() , readBookByPage() ,които се навръзват със съответните команди в book класа.

Source.cpp

стартовото cpp. В него има функция login() ,която дава опцията за влизане в библиотеката. Ако не се намери потребител или паролата е грешна автоматично се влиза като гост (който няма администратоски права).

Възможности за осърваъшенстване:

Добре е да се облисливарианта users да е масив и членнданнта в library().

Може да се направи допълнителен клас interface вместо да се държи на едно място с library.

Използване на string класове(еквиваленти на string библиотеката)