 Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Курсов проект по Обектно-ориентирано програмиране специалност Информационни системи

Тема №10 Библиотека

Съдържание

1. Увод……………………………………………………………………………………..…....3
   1. Описание и идея на проекта
   2. Цели и задачи на разработката
   3. Връзка към хранилището в Github

1. Общ поглед върху проекта……………………………………………………………….....4
   1. Основни дефиниции, концепции и алгоритми, използвани в проекта
   2. Подходи и методи за решаване на поставените проблеми

1. Проектиране…………………………………………………………………………………5
2. Заключение…………………………………………………………………………………..8

# 1. Увод

### 1.1. Описание и идея на проекта

Проектът „Библиотека“ реализира виртуална библиотека, в която потребителят може да разглежда информация за добавените в нея книги, а администраторът да добавя нови такива или да изтрива вече съществуващи.

### 1.2. Цели и задачи на разработката

Целта на проекта е да се създаде оптимална програма с максимално разнообразна функционалност, която да отговаря на поставената задача.

**1.3. Връзка към хранилището в Github**: https://github.com/mitache2000/Library

# 2. Общ поглед върху проекта

### 2.1. Основни дефиниции, концепции и алгоритми, използвани в проекта

Данните в проекта са разпределени в съответни класове. За по-добра и подредба проектът е съставен от header и cpp файлове, като във всеки header се съдържа отделен клас, обединяващ необхидимите за неговата имплементация член-данни и член-функции, които са разпределени в public и private секциите в зависимост от необходимия достъп до тях, а в cpp файла се намират дефинициите на съответните член-функции.

### 2.2. Подходи и методи за решаване на поставените проблеми

В класовете Vector и String използвам така наречената Big four, т.е. съдържат член-функциите Конструктор, Копи конструктор и оператор= и деструктор, за да може програмата да работи правилно.

Използвани са следните библиотеки:

* <fstream> // библиотека за работа с файлове
* <iostream> //стандартната библиотека за вход и изход

# 3. Проектиране

За имплементацията на проекта са създадени следните класове:

* User
* Date
* Book
* Checker
* Commands
* Клас **User** се характеризира с член-данните в private секцията :

o string name (името на потребителя)

o string password (паролата на потребителя)

В public секцията се намират необходимите конструктор, конструктор с параметри, копи конструктор, оператор=, оператор<<, гетъри, функцията bool isAdmin(), която проверява дали потребителят е админ

* Клас **Date** се характеризира със следните член-данни в секцията със спецификатор за достъп private:
  + - Int day, int month, int year.

Методите на този клас са съответно конструктор, копи конструктор, оператори< и >, както и оператор<<

* Клас **Book** се състои от член-данните в public секцията:
  + size\_t id;
  + String title;
  + String author;
  + String genre;
  + String description;
  + Date year;
  + String keyWords;
  + float rating;

Функции в public секцията са: конструктор, копи конструктор и функцията PrintBookInformation(), която ни изкарва информация за книгата.

* Клас **Checker** съдържа функциите: bool doesUserExist, bool doesUserLoginExist и bool DoesBookExist, които ги използваме за проверка когато искаме да изтрием книга или потребител или да влезем във вече съществуващ акаунт.
* Клас **Commands** няколко член-данни както и всички основни функции, които използваме в проекта.

Член данните, които използваме са:

* + Vector<User>allUsers – в нея се добавят всички потребители
  + Vector<Book>allBooks – в нея се добавят всички книги
  + User user – потребител от клас User
  + String filePath – ще запазва името на файлът, който ще отваряма
  + Checker checker – ще го използваме, за да правим проверките от клас Checker
  + Bool isUserLogedIn – показва дали потребителят е логнат или не

Функциите са:

* void addUser() – добавя нов потребител, достъпва се само от администратор
* void removeUser() – изтрива потребител, достъпва се само от администратор
* void logInUser() – логва потребителят
* void LogOutUser – отписва потребителят
* void AdminActions(String& path) – извършва командите, които са само за администратор – add и remove user
* void BookSplit() – тук са всички операции с книги – add, all, info, sort, find
* void SaveBooks(String& path) – записва книгата
* String enterUsername(String& username) – въвеждаме потребителско име
* String enterPassword(String& password) – въвеждаме парола
* Bool isLeap(int year) – показва дали годината е високосна
* Date enterYear(Date& date) – добавяме година
* Void PrintBooks(Vector<Book>& allBooksFromFile) – показва всички книги
* Void AddUserToFile(String& username, String& password)
* Void RemoveUserFromFile(String& usernameToRemove, Vector<User>& allUsers)
* **Main.cpp** съдържа външните функции:
  + Void PrintCommandsForLibrary() – извежда информация кои команди са достъпни и какво правят
  + Int Start(String& path) – първо проверява кой файл е отворен, а след това в зависимост коя команда въведен, извиква функция от класа Commands
  + Int main() – проверява дали отвореният файл от Start съществува, а после показва възможните команди. Когато се подаде „exit“, програмата се затваря.

-

# 4. Заключение

С подходяща графика и разширение на функционалността този проект може да се разрасне и да се превърне в база данни за библиотека. В последствие може да бъде направено и приложение за телефон, за да бъде още по-достъпно.