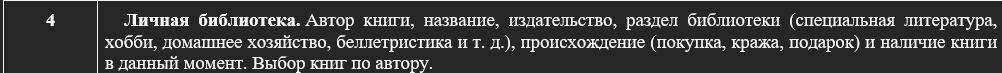
Лабораторная работа 4

По дисциплине «Основы алгоритмизации и программирования»

На тему «Представление информации в виде структуры»

***Задание#1***



Macros.h

#include <iostream>

#include <vector>

#include <ctime>

#include <cmath>

#include <iomanip>

#include <stack>

#include <map>

#include <string>

#include <chrono>

#include <fstream>

#include <Windows.h>

#define vi cout <<

#define vv cin >>

#define nl cout << "\n";

#define nw cout << "\t";

#define SCOCP1251 SetConsoleOutputCP(1251);

#define SCCP1251 SetConsoleCP(1251);

#define ret return

#define pause system("pause");

using namespace :: std;

.cpp

#include "Macros.h"

// Структура для хранения информации о книге

struct Book {

string author;

string title;

string publisher;

string section;

string origin;

bool isAvailable;

};

// Вектор для хранения всех книг

vector<Book> books;

// Функция для добавления новой книги

void addBook() {

Book newBook;

vi "Автор книги: ";

getline(cin, newBook.author);

vi "Название: ";

getline(cin, newBook.title);

vi "Издательство: ";

getline(cin, newBook.publisher);

vi "Раздел библиотеки: ";

getline(cin, newBook.section);

vi "Происхождение (покупка, кража, подарок): ";

getline(cin, newBook.origin);

vi "Наличие книги (1 - есть в наличии, 0 - нет): ";

vv newBook.isAvailable;

cin.ignore();

books.push\_back(newBook);

}

// Функция для отображения всех книг

void displayBooks() {

for (const auto& book : books) {

vi "Автор: " << book.author << ", Название: " << book.title

<< ", Издательство: " << book.publisher << ", Раздел: " << book.section

<< ", Происхождение: " << book.origin

<< ", Наличие: " << (book.isAvailable ? "Есть" : "Нет");

nl

}

}

// Функция для удаления книги по названию

void deleteBook(const string& titleToDelete) {

int foundIndex = -1;

for (int i = 0; i < books.size(); i++) {

if (books[i].title == titleToDelete) {

foundIndex = i;

break;

}

}

if (foundIndex != -1) {

books.erase(books.begin() + foundIndex);

vi "Книга \"" << titleToDelete << "\" удалена.\n";

}

else {

vi "Книга \"" << titleToDelete << "\" не найдена.\n";

}

}

// Функция для поиска книги по автору

void searchByAuthor(const string& authorToFind) {

for (const auto& book : books) {

if (book.author == authorToFind) {

vi "Найдена книга: " << book.title << endl;

}

}

}

// Функция для сохранения всех книг в файл

void saveToFile(const string& filename) {

ofstream outFile(filename);

if (!outFile) {

cerr << "Ошибка открытия файла для записи.\n";

ret;

}

for (const auto& book : books) {

outFile << book.author << ","

<< book.title << ","

<< book.publisher << ","

<< book.section << ","

<< book.origin << ","

<< book.isAvailable << "\n";

}

outFile.close();

}

// Функция для загрузки всех книг из файла

void loadFromFile(const string& filename) {

ifstream inFile(filename);

if (!inFile) {

cerr << "Ошибка открытия файла для чтения.\n";

ret;

}

Book newBook;

int isAvailableInt;

while (getline(inFile, newBook.author, ',')) {

getline(inFile, newBook.title, ',');

getline(inFile, newBook.publisher, ',');

getline(inFile, newBook.section, ',');

getline(inFile, newBook.origin, ',');

inFile >> isAvailableInt;

inFile.ignore();

newBook.isAvailable = isAvailableInt != 0;

books.push\_back(newBook);

}

inFile.close();

}

int main() {

SCOCP1251

SCCP1251

int choice;

do {

// Вывод меню управления библиотекой

vi "\nУправление личной библиотекой\n";

vi "1. Добавить книгу\n";

vi "2. Показать книги\n";

vi "3. Удалить книгу\n";

vi "4. Поиск по автору\n";

vi "5. Сохранить в файл\n";

vi "6. Загрузить из файла\n";

vi "0. Выход\n";

vi "Введите ваш выбор: ";

vv choice;

cin.ignore();

// Обработка выбора пользователя

switch (choice) {

case 1:

// Добавление новой книги

addBook();

break;

case 2:

// Отображение всех книг

displayBooks();

break;

case 3: {

// Удаление книги по названию

string titleToDelete;

vi "Введите название книги для удаления: ";

getline(cin, titleToDelete);

deleteBook(titleToDelete);

break;

}

case 4: {

// Поиск книги по автору

string authorToFind;

vi "Введите автора для поиска: ";

getline(cin, authorToFind);

searchByAuthor(authorToFind);

break;

}

case 5: {

// Сохранение всех книг в файл

string filename;

vi "Введите имя файла для сохранения: ";

getline(cin, filename);

saveToFile(filename);

break;

}

case 6: {

// Загрузка всех книг из файла

string filename;

vi "Введите имя файла для загрузки: ";

getline(cin, filename);

loadFromFile(filename);

break;

}

case 0:

// Выход из программы

vi "Выход из программы...\n";

break;

default:

// Некорректный выбор

vi "Некорректный выбор\n";

}

} while (choice != 0);

pause

ret 0;

}