

Чумин Владимир Владимирович

КЭ-142

Лабораторная № 7

Задание:

Ваша задача - создать простую систему управления библиотекой на C++. Система должна позволять добавлять новые книги в библиотеку, брать книги в займы, возвращать книги и выводить список всех книг в библиотеке.

Также система должна сохранять состояние библиотеки между запусками программы, сохраняя и загружая данные о книгах в файл и из файла. Например, если пользователь попытается взять в займы книгу, которая уже кем-то взята, система должна сообщить ему об этом. Когда книга возвращается, система должна позволить пользователю взять её в займы. Вам следует создать разные классы, например класс для библиотеки и класс для книги, а также использовать вектор.

Код:

```
#include <iostream>
#include <list>
#include "locale.h"
#include <fstream>
#include <vector>
#include <string>
using namespace std;

class Book {
protected:
    string title;
    string author;
    string genre;
    int year;
```

```

    bool isAvaible = true;

public:
    Book(string title, string author, string genre,
int year)
        : title(move(title)), author(move(author)),
genre(move(genre)), year(year) {}

    // Сериализация данных книги в строку
    string serialize() const {
        return title + ";" + author + ";" + genre +
";" + to_string(year) + ";" +
to_string(isAvaible);
    }

    Book(int year = -1, bool isAvaible = true) :
year(year), isAvaible(isAvaible) {}

    void takeBook() {
        if (!isAvaible) {
            cout << "Книга уже взята" << endl;
            return;
        }
        cout << "Книга взята" << endl;
        isAvaible = false;
    }

    string getTitle() const {
        return title;
    }

    void returnBook() {
        if (isAvaible) {
            cout << "Книга уже находится в
библиотеке" << endl;

```

```

        return;
    }
    cout << "Книга возвращена" << endl;
    isAvaible = true;
}

// Десериализация строки в данные книги
void deserialize(const string& data) {
    size_t pos = 0;
    size_t next_pos = data.find(';', pos);

    title = data.substr(pos, next_pos - pos);
    pos = next_pos + 1;
    next_pos = data.find(';', pos);

    author = data.substr(pos, next_pos - pos);
    pos = next_pos + 1;
    next_pos = data.find(';', pos);

    genre = data.substr(pos, next_pos - pos);
    pos = next_pos + 1;
    next_pos = data.find(';', pos);

    year = stoi(data.substr(pos, next_pos -
pos));
    pos = next_pos + 1;

    isAvaible = stoi(data.substr(pos));
}

void redactBook(string title = "", string
author = "", string genre = "", int year = -1) {
    if (!title.empty()) this->title = title;
    if (!author.empty()) this->author = author;
    if (!genre.empty()) this->genre = genre;
}

```

```

        if (year != -1) this->year = year;
    }

    void print() const {
        cout << "Название: " << title << endl;
        cout << "Автор: " << author << endl;
        cout << "Жанр: " << genre << endl;
        cout << "Год: " << year << endl;
        cout << "Доступность: " << (isAvaible ?
"Доступна" : "Недоступна") << endl;
    }

};

class Library {
protected:
    vector<Book> library;
    string filename;

    int findBook(const string& title) {
        for (int i = 0; i < library.size(); i++) {
            if (library[i].getTitle() == title) {
                return i;
            }
        }
        return -1;
    }

public:

    // Конструктор
    Library(const string& filename) :
filename(filename) {}

    void cliarLibrary() {

```

```

        library.clear();
    }

    void saveLibrary() {
        ofstream fileOut(filename);
        for (const auto& book : library) {
            fileOut << book.serialize() << endl;
        }
        fileOut.close();
    }

    void loadLibrary() {
        ifstream fileIn(filename);
        string line;
        while (getline(fileIn, line)) {
            Book book("", "", "", 0);
            book.deserialize(line);
            library.push_back(book);
        }
        fileIn.close();
    }

    void printLibrary() {
        for (const auto& book : library) {
            book.print();
            cout << endl;
        }
    }

    // Работа с книгами
    void takeBook(const string& title) {
        int index = findBook(title);
        if (index == -1) {
            cout << "Книга не найдена" << endl;
        }
    }

```

```
        return;
    } else {
        library[index].takeBook();
    }
}

void returnBook(const string& title) {
    int index = findBook(title);
    if (index == -1) {
        cout << "Книга не найдена" << endl;
        return;
    } else {
        library[index].returnBook();
    }
}

void printBook(const string& title) {
    int index = findBook(title);
    if (index == -1) {
        cout << "Книга не найдена" << endl;
        return;
    } else {
        library[index].print();
    }
}

void removeBook(const string &title) {
    int index = findBook(title);
    if (index == -1) {
        cout << "Книга не найдена" << endl;
        return;
    } else {
        library.erase(library.begin() + index);
    }
}
```

```

    void addBook(const string& title, const string&
author, const string& genre, int year) {
        library.emplace_back(title, author, genre,
year);
    }

    void redactBook(const string& title, const
string& author, const string& genre, int year) {
        int index = findBook(title);
        if (index == -1) {
            cout << "Книга не найдена" << endl;
            return;
        } else {
            library[index].redactBook(title,
author, genre, year);
        }
    }

    bool isBookHere(const string& title) {
        return findBook(title) != -1;
    }

    ~Library() {
        saveLibrary();
        cout << "Конец сеанса\n" << endl;
    }
};

class Interface {
private:
    string username = "admin";
    string password = "admin";
    string getInfo() {
        string info;

```

```
        getline(cin, info);
        return info;
    }
    void wait() {
        cout << "\nНажмите Enter для продолжения"
<< endl;
        cin.get();
    }
public:
    bool login() {
        cout << "Введите логин: ";
        string login = getInfo();
        cout << "Введите пароль: ";
        string password = getInfo();

        return login == username && password ==
this->password;
    }

    string getName() {
        cout << "Введите название книги: ";
        return getInfo();
    }

    string getAuthor() {
        cout << "Введите автора книги: ";
        return getInfo();
    }

    string getGenre() {
        cout << "Введите жанр книги: ";
        return getInfo();
    }

    int getYear() {
```



```
    int year;
    cout << "Введите год книги: ";
    cin >> year;
    cin.ignore();
    return year;
}
```

```
char getAction() {
    char action;
    wait();

    cout << "Выберите действие: " << endl;
    cout << "1. Взять книгу" << endl;
    cout << "2. Вернуть книгу" << endl;
    cout << "3. Редактировать книгу" << endl;
    cout << "4. Добавить книгу" << endl;
    cout << "5. Показать все книги" << endl;
    cout << "6. Показать книгу" << endl;
    cout << "7. Удалить книгу" << endl;
    cout << "8. Выйти" << endl;
    cout << "9. УДАЛИТЬ ВСЕ ДАННЫЕ" << endl;
    cin >> action;
    cin.ignore();

    return action;
}
};
```

```
int main () {
    setlocale(LC_ALL, "Russian");
    Interface interface;
```

```
    cout << "Добро пожаловать в библиотеку" <<
endl;

    if (!interface.login()) {
        cout << "Неверный логин или пароль" <<
endl;
        return 0;
    }

    Library current_library("library.txt");
    current_library.loadLibrary();

    while (true) {
        char action = interface.getAction();

        switch (action) {
            case '1': {
                string title = interface.getName();
                current_library.takeBook(title);
                break;
            }

            case '2': {
                string title = interface.getName();
                current_library.returnBook(title);
                break;
            }

            case '3': {
                string title = interface.getName();
                if
(!current_library.isBookHere(title)) {
                    cout << "Книга не найдена" <<
endl;
                    break;
                }
            }
        }
    }
}
```

```

    }

    cout << "Если не хотите изменять
значение, оставьте поле пустым" << endl;
    string author =
interface.getAuthor();
    string genre =
interface.getGenre();
    int year = interface.getYear();
    current_library.redactBook(title,
author, genre, year);
    break;
}

    case '4': {
        string title = interface.getName();
        string author =
interface.getAuthor();
        string genre =
interface.getGenre();
        int year = interface.getYear();
        current_library.addBook(title,
author, genre, year);
        break;
    }

    case '5': {
        current_library.printLibrary();
        break;
    }

    case '6': {
        string title = interface.getName();
        current_library.printBook(title);
        break;
    }
}

```

```

    }

    case '7': {
        string title = interface.getName();
        current_library.removeBook(title);
        break;
    }

    case '8': {
        return 0;
    }

    case '9': {
        current_library.cliarLibrary();
        return 0;
    }

    default: {
        cout << "Неверное действие" <<
endl;

        break;
    }
}
}
}
}
}

```

Результат:

```
vol@vol-All-Series:~/CLionProjects/C++_Labs$ g++ Lab7/lab_7.cpp && ./a.out
Добро пожаловать в библиотеку
Введите логин: admin
Введите пароль: admin

Нажмите Enter для продолжения

Выберите действие:
1. Взять книгу
2. Вернуть книгу
3. Редактировать книгу
4. Добавить книгу
5. Показать все книги
6. Показать книгу
7. Удалить книгу
8. Выйти
9. УДАЛИТЬ ВСЕ ДАННЫЕ
4
Введите название книги: Учебник по C++
Введите автора книги: Кафедра информационных технологий ЮУрГУ
Введите жанр книги: Учебник
Введите год книги: 2022
```

```
Введите название книги: Учебник по Python
Введите автора книги: Кафедра информационных технологий ЮУрГУ
Введите жанр книги: 2024
Введите год книги: 2024

Нажмите Enter для продолжения

Выберите действие:
1. Взять книгу
2. Вернуть книгу
3. Редактировать книгу
4. Добавить книгу
5. Показать все книги
6. Показать книгу
7. Удалить книгу
8. Выйти
9. УДАЛИТЬ ВСЕ ДАННЫЕ
3
Введите название книги: Учебник по Python
Если не хотите изменять значение, оставьте поле пустым
Введите автора книги:
Введите жанр книги: учебник
Введите год книги:

2016
```

```
vol@vol-All-Series:~/CLionProjects/C++_Labs$ cat library.txt
Учебник по C++;Кафедра информационных технологий ЮУрГУ;Учебник;2022;1
Учебник по Python;Кафедра информационных технологий ЮУрГУ;учебник;2016;1
```