Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет среднего профессионального образования

Санкт-Петербург 2019

Специальность: 09.02.07 Информационные системы и программирование

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

по теме: Стандартная библиотека шаблонов

**Цель работы:** Написать программу для учёта книг в библиотеке. Сведенья о книгах содержат: фамилию и инициалы автора, название, год издания, количество экземпляров данной книги в библиотеке. Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций: добавление данных о книгах, вновь поступающих в библиотеку; удаление данных о списываемых книгах; выдача сведений о всех книгах, упорядоченных по фамилиям авторов; выдача сведений о всех книгах, упорядоченных о годам издания. Хранение данных организовать с применением контейнерного класса vector.

Выполнил:

Студент группы Y2333

\_\_\_\_\_\_\_\_\_\_\_\_\_Харченко Д. И.

Проверил:

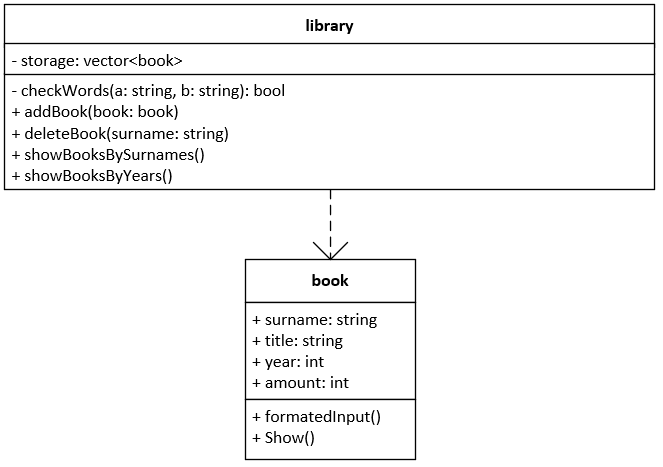
\_\_\_\_\_\_\_\_ Антонов М. Б.

Дата: “\_\_” \_\_\_\_\_\_ 2019г.

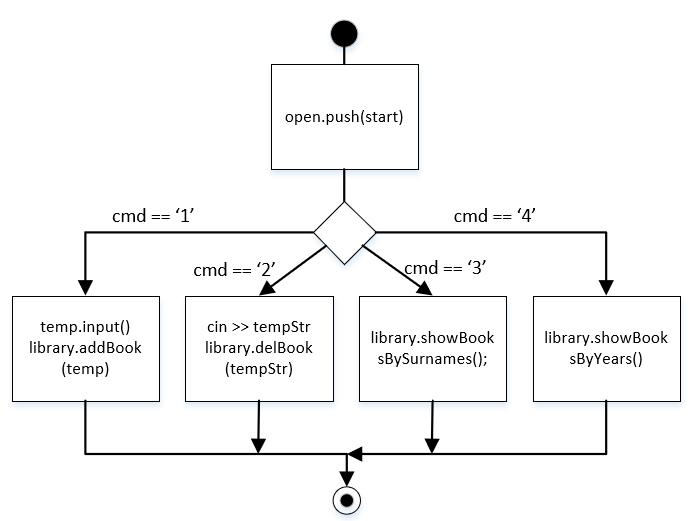
Оценка:\_\_\_\_\_\_\_\_\_\_\_

**UML-диаграммы:**

На рисунках 1 представлена UML-диаграмма классов. На рисунке 2 представлена диаграмма функции main.

**

*Рисунок 1. Диаграмма классов.*

**

*Рисунок 2. Диаграмма функции main.*

**Код программы:**

Файл main.cpp:

/\*\*

\* @file main.cpp

\* @author Popovskaya Angelina

\* @brief Лабораторная работа 8

\*/

#include <iostream>

#include <vector>

#include <conio.h>

#include "book.hpp"

#include "library.hpp"

using namespace std;

int main()

{

library library;

setlocale(LC\_ALL, "rus");

string tempStr;

book tempBook;

while(true){

system("cls");

cout << "1 - Добавить книгу\n2 - удалить книгу\n3 - Показать книги, упорядоченные по фамилиям авторов\n4 - Показать книги, упорядоченные по годам издания\n";

char cmd = getch();

switch(cmd){

case '1':

tempBook.formatedInput();

library.addBook(tempBook);

break;

case '2':

cout << "Введите название: ";

cin >> tempStr;

library.deleteBook(tempStr);

break;

case '3':

library.showBooksBySurnames();

break;

case '4':

library.showBooksByYears();

break;

}

system("pause");

}

return 0;

}

Файл library.hpp:

/\*\*

\* @file library.hpp

\* @author Popovskaya Angelina

\* @brief Лабораторная работа 8

\*/

class library

{

private:

vector<book> storage;

bool checkWords(string a, string b)

{

/\*\*

\* @brief Сравнить две строки по алфавиту

\* @param a Первая строка

\* @param b Вторая строка

\* @return true в случае

\*/

for(int i = 0; i < min(a.length(), b.length()); i++){

if(tolower(a[i]) > tolower(b[i]))

return true;

if(tolower(a[i]) < tolower(b[i]))

return false;

}

return a.length() > b.length();

}

public:

void addBook(book book)

{

/\*\*

\* @brief Добавить книгу в библиотеку

\* @param book Книга, которая будет добавлена

\*/

storage.push\_back(book);

}

void deleteBook(string title)

{

/\*\*

\* @brief Удалить книгу из библиотеки

\* @param title Название книги, котоаря будет удалена

\*/

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

if(storage[i].title == title){

storage.erase(storage.begin() + i);

break;

}

}

void showBooksBySurnames()

{

/\*\*

\* @brief Вывести список книг, сортированных по фамилиям авторов

\*/

int minindex;

book temp;

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

minindex=i;

for(int j = i; j < storage.size(); j++)

if(checkWords(storage[minindex].surname, storage[j].surname))

minindex = j;

temp = storage[minindex];

storage[minindex]=storage[i];

storage[i] = temp;

storage[i].Show();

}

}

void showBooksByYears()

{

/\*\*

\* @brief Вывести список книг, сортированных по дате издания

\*/

int minindex;

book temp;

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

minindex=i;

for(int j = i; j < storage.size(); j++)

if(storage[j].year < storage[minindex].year)

minindex = j;

temp = storage[minindex];

storage[minindex]=storage[i];

storage[i] = temp;

storage[i].Show();

}

}

};

Файл book.hpp:

/\*\*

\* @file book.hpp

\* @author Popovskaya Angelina

\* @brief Класс книги

\*/

#include <iostream>

#include <limits>

using namespace std;

class book

{

public:

string surname, title;

int year, amount;

void formatedInput()

{

/\*\*

\* @brief Считать книгу с клавиатуры

\*/

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Введите название: ";

getline(cin, title);

cout << "Введите фамилию автора: ";

getline(cin, surname);

cout << "Введите год издания: ";

cin >> year;

cout << "Введите количество штук: ";

cin >> amount;

}

void Show()

{

/\*\*

\* @brief Вывод информации о книге

\*/

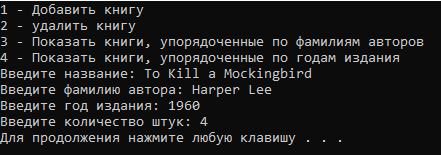
cout << "Название: " << title << "\nАвтор: " << surname << "\nГод издания: " << year << "\nКоличество: " << amount << endl << endl;

}

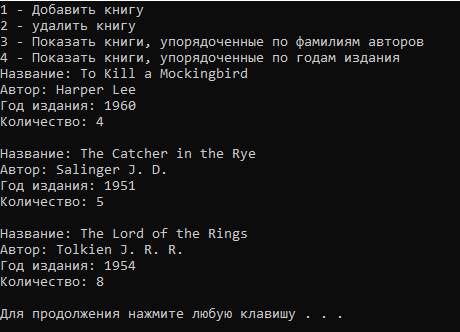
};

**Протокол программы:**

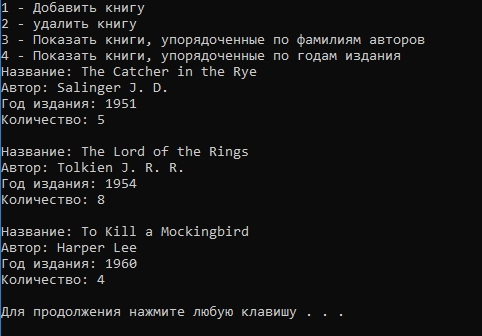
На рисунке 3-6 представлены скриншоты работы прогаммы.



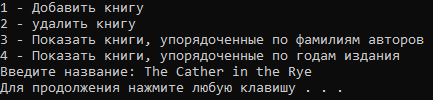
*Рисунок 3 – Добавление книги.*

**

*Рисунок 4 – Вывод книг, упорядоченных по фамилиям авторов.*

**

*Рисунок 5 – Вывод книг, упорядоченных по дате издания.*

**

*Рисунок 6 – Удаление книги.*

**Вывод:** В ходе лабораторной работы были изучены классы стандартной библиотеки шаблонов.