Министерство образования и науки РФ

Севастопольский государственный университет

Институт информационных технологий и управления в технических сетях

Лабораторная работа №7

**Контейнеры**

Выполнил:

ст.гр.ИСб-22д

Воронин И.Ю.

Проверил:

Тимофеев И.С.

Севастополь

2015

1.Цель работы

Приобретение практических навыков в написании объектно-ориентированных программ с использованием контейнеров стандартной библиотеки шаблонов. Освоение особенностей отладки объектно-ориентированных программ.

.

2.Вариант задания

***Вариант 10***

Написать программу учета книг в библиотеке. Сведения 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 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 по годам издания.

Хранение данных организовать с применением контейнерного класса vector.

3.Текст программы

Класс Book:

#include <string>

#include <iostream>

typedef unsigned short us;

class Book{

private:

std::string author;

std::string title;

us year;

us amount;

public:

Book();

Book(std::string \_author, std::string \_title, int \_year, int \_amount);

std::string getAuthor();

std::string getTitle();

us getYear();

us getAmount();

Book Add();

~Book();

};

#include "Book.h"

Book::Book(){

author = "";

title = "";

year = 0;

amount = 0;

}

Book::Book(std::string \_author, std::string \_title, int \_year, int \_amount){

author = \_author;

title = \_title;

year = \_year;

amount = \_amount;

}

Book Book::Add(){

std::cout << "Auhtor: ", std::cin >> author;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cout << "Title: ", std::cin >> title;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cout << "Year: ", std::cin >> year;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cout << "Amount: ", std::cin >> amount;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

return \*this;

}

std::string Book::getAuthor(){

return author;

}

std::string Book::getTitle(){

return title;

}

us Book::getYear(){

return year;

}

us Book::getAmount(){

return amount;

}

Book::~Book(){}

Класс Librarian:

#pragma once

#include <vector>

#include "Book.h"

class Librarian

{

public:

Librarian();

void Swap(Book &a, Book &b){

Book temp;

temp = a;

a = b;

b = temp;

}

Book NewBook();

void DeleteBook(std::vector <Book>&);

void AuthorSort(std::vector <Book>&);

void YearSort(std::vector <Book>&);

~Librarian();

};

#include "Librarian.h"

Librarian::Librarian(){

}

Book Librarian::NewBook(){

Book \*Temp = new Book();

Temp->Add();

return \*Temp;

}

void Librarian::AuthorSort(std::vector <Book> &book){

for (int i = 0; i < book.size()-1; i++){

for (int j = i + 1; j < book.size(); j++){

if (book[i].getAuthor() > book[j].getAuthor()){

Swap(book[i], book[j]);

}

}

}

}

void Librarian::YearSort(std::vector <Book> &book){

for (int i = 0; i < book.size()-1; i++){

for (int j = i + 1; j < book.size(); j++){

if (book[i].getYear() > book[j].getYear()){

Swap(book[i], book[j]);

}

}

}

}

void Librarian::DeleteBook(std::vector <Book> &book){

std::string del\_flag = "";

std::cout << "Title element to delete: ", std::cin >> del\_flag;

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

if (book[i].getTitle() == del\_flag) {

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

std::cout << std::endl << "Element was deleted." << std::endl;

}

}

}

Librarian::~Librarian(){

}

Файл Main:

#include <iostream>

#include <vector>

#include "Librarian.h"

#include "Book.h"

int Menu(void);

int main(){

std::vector <Book> arr;

Librarian \*libr = new Librarian();

while (true){

system("CLS");

switch (Menu()){

case 1:{

arr.push\_back(libr->NewBook());

break;

}

case 2:{

libr->DeleteBook(arr);

break;

}

case 3:{

libr->AuthorSort(arr);

for (int i = 0; i < arr.size(); i++) std::cout << arr[i].getAuthor() << std::endl;

break;

}

case 4:{

libr->YearSort(arr);

for (int i = 0; i < arr.size(); i++) std::cout << arr[i].getYear() << std::endl;

break;

}

case 0:{

arr.clear();

delete libr;

return 0;

break;

}

}

system("PAUSE");

}

return 0;

}

int Menu(){

int menu\_number;

do{

std::cout << "1 - Add." << std::endl;

std::cout << "2 - Delete." << std::endl;

std::cout << "3 - Show FIO." << std::endl;

std::cout << "4 - Show Year. " << std::endl;

std::cout << "0 - Exit. " << std::endl;

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cin >> menu\_number;

}

while (menu\_number < -1 && menu\_number > 5);

return menu\_number;

}

4.Выполнения программы

Тест программы проводится при вводе неупорядоченных по полям данных книг. Вводятся следующие данные:

Gogol Dead\_Souls 1842 3

Tolkin LOTR 1954 10

Pratchet Mor 1980 1

Bylgakov White\_Guard 1920 2

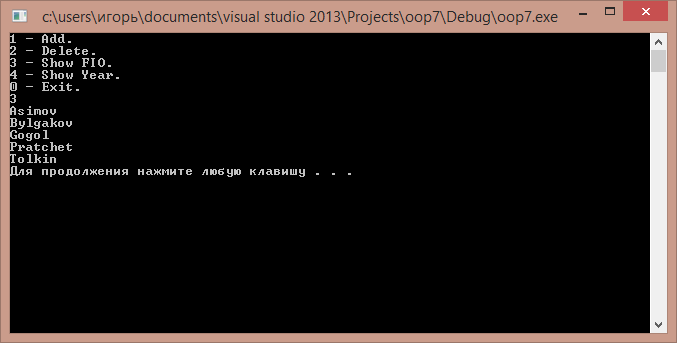


Рисунок 4.1 - Тест пункта №3.

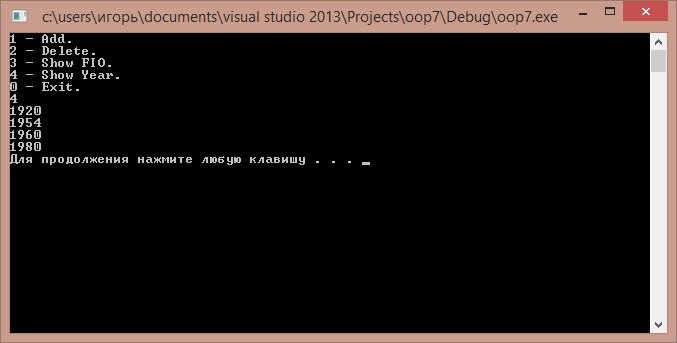


Рисунок 4.2 - Тест пункта №4 после удаления записи.

ВЫВОДЫ

В данной лабораторной работе были изучены классы STL типа контейнеры. Изучены виды контейнеров, итераторы. Была написана программа из двух классов, для реализации функции библиотеки.