|  |  |
| --- | --- |
| C:\Documents and Settings\admin\Рабочий стол\МИРЭА_ЭМБЛЕМА_приказ.JPG | |
| МИНОБРАНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  «**МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  **МИРЭА** | |
| Институт информационных технологий  Кафедра математического обеспечения и стандартизации информационных технологий | |
| **Отчет по лабораторной работе №4** **Тема работы «****КомбиНированный тип - Структура»** | |
| Выполнил: |  |
| Студент группы ИКБО-07-17  Отчет принял:  Преподаватель кафедры МОСИТ | Данилов Д.Е.  Миронов Антон Николаевич |
| Москва 2017 | |

# **Задание 1**

# **Условие задания**

# Учет выдачи книг пользователям библиотеки. Карточка пользователя библиотеки содержит сведения, о выданной книге: Инвентарный номер, Автор, Название, Дата выдачи, Дата возврата

* Отсортировать по дате выдачи методом прямого выбора
* Сформировать список книг, которые не возвращены вовремя.

**Декомпозиция**

* Создание структур
* Ввод отдельной структуры
* Вывод отдельной структуры
* Добавление структуры в динамический массив структур
* Вывод динамического массива структур
* Сортировка динамического массива структур
* Создание второго динамического массива структур со структурами, удовлетворяющими условию задачи варианта

# **Реализация функций**

bool digitcheck(unsigned int a)

{

cin.clear();

cin.sync();

string s;

getline(cin, s, '\n');

for (auto x : s)

{

if (!isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

bool spellcheck(string a)

{

cin.clear();

cin.sync();

for (auto x : a)

{

if (isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

void inp(LibraryUser &a) //Ввод структуры

{

cout << "Введиите номер книги" << endl;

cin >> a.iNumber;

while ((!digitcheck(a.iNumber)) || a.iNumber < 1 || a.iNumber > 65535)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.iNumber;

}

///////////////////////////////////

cout << "Введите автора книги (Без пробелов!!!)" << endl;

cin >> a.Author;

while (!spellcheck(a.Author))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Author;

}

///////////////////////////////////

cout << "Введите название книги (Без пробелов!!!)" << endl;

cin >> a.Title;

while (!spellcheck(a.Title))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Title;

}

///////////////////////////////////

cout << "Введите дату выдачи книги" << endl;

cout << "День: ";

cin >> a.Give.day;

while ((!digitcheck(a.Give.day)) || a.Give.day < 1 || a.Give.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.day;

}

cout << "Месяц: ";

cin >> a.Give.month;

while ((!digitcheck(a.Give.month)) || a.Give.month < 1 || a.Give.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.month;

}

cout << "Год: ";

cin >> a.Give.year;

while ((!digitcheck(a.Give.year)) || a.Give.year < 1 || a.Give.year > 2018)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.year;

}

///////////////////////////////////

cout << "Введите дату возврата книги" << endl;

cout << "День: ";

cin >> a.Back.day;

while ((!digitcheck(a.Back.day)) || a.Back.day < 1 || a.Back.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.day;

}

cout << "Месяц: ";

cin >> a.Back.month;

while ((!digitcheck(a.Back.month)) || a.Back.month < 1 || a.Back.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.month;

}

cout << "Год: ";

cin >> a.Back.year;

while ((!digitcheck(a.Back.year)) || a.Back.year < 1 || a.Back.year > 2018 || (a.Back.day < a.Give.day && a.Back.month <= a.Give.month && a.Back.year <= a.Give.year) || (a.Back.month < a.Give.month && a.Back.year <= a.Give.year) || (a.Back.year < a.Give.year))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.year;

}

cout << endl;

}

void out(LibraryUser a)

{

cout << "Номер книги - " << a.iNumber << endl;

cout << "Автор книги - " << a.Author << endl;

cout << "Название книги - " << a.Title << endl;

cout << "Дата выдачи - " << a.Give.day << "." << a.Give.month << "." << a.Give.year << endl;

cout << "Дата возврата - " << a.Back.day << "." << a.Back.month << "." << a.Back.year << endl;

}

void inpV(LibraryUser a, vector<LibraryUser> &b)

{

b.push\_back(a);

}

void sort(vector<LibraryUser> &a)

{

int min;

for (unsigned int i = 0; i < (a.size() - 1); i++)

{

min = i;

for (unsigned int j = i + 1; j < a.size(); j++)

{

if (((a[j].Give.day < a[min].Give.day) && (a[j].Give.month == a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.month < a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.year < a[min].Give.year)))

min = j;

}

swap(a[i], a[min]);

}

}

void outV(vector<LibraryUser> a)

{

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

{

cout << (i + 1) << ")" << endl;

cout << "Номер книги - " << a[i].iNumber << endl;

cout << "Автор книги - " << a[i].Author << endl;

cout << "Название книги - " << a[i].Title << endl;

cout << "Дата выдачи - " << a[i].Give.day << "." << a[i].Give.month << "." << a[i].Give.year << endl;

cout << "Дата возврата - " << a[i].Back.day << "." << a[i].Back.month << "." << a[i].Back.year << endl << endl;

}

}

void Not\_back(vector<LibraryUser> a, vector<LibraryUser> &b)

{

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

{

if (!((a[i].Back.day >= a[i].Give.day && a[i].Back.month == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && (a[i].Back.month - 1) == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && a[i].Back.month == 1 && a[i].Give.month == 12 && (a[i].Back.year - 1) == a[i].Give.year)))

b.push\_back(a[i]);

}

}

# **Кодирование алгоритма программы**

#include "stdafx.h"

#include <iostream>

#include <string>

#include <vector>

#include <Windows.h>

using namespace std;

struct Date

{

unsigned short day;

unsigned short month;

unsigned short year;

};

struct LibraryUser

{

unsigned int iNumber;

string Author;

string Title;

Date Give;

Date Back;

};

bool digitcheck(unsigned int a)

{

cin.clear();

cin.sync();

string s;

getline(cin, s, '\n');

for (auto x : s)

{

if (!isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

bool spellcheck(string a)

{

cin.clear();

cin.sync();

for (auto x : a)

{

if (isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

void inp(LibraryUser &a) //Ввод структуры

{

cout << "Введиите номер книги" << endl;

cin >> a.iNumber;

while ((!digitcheck(a.iNumber)) || a.iNumber < 1 || a.iNumber > 65535)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.iNumber;

}

///////////////////////////////////

cout << "Введите автора книги (Без пробелов!!!)" << endl;

cin >> a.Author;

while (!spellcheck(a.Author))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Author;

}

///////////////////////////////////

cout << "Введите название книги (Без пробелов!!!)" << endl;

cin >> a.Title;

while (!spellcheck(a.Title))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Title;

}

///////////////////////////////////

cout << "Введите дату выдачи книги" << endl;

cout << "День: ";

cin >> a.Give.day;

while ((!digitcheck(a.Give.day)) || a.Give.day < 1 || a.Give.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.day;

}

cout << "Месяц: ";

cin >> a.Give.month;

while ((!digitcheck(a.Give.month)) || a.Give.month < 1 || a.Give.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.month;

}

cout << "Год: ";

cin >> a.Give.year;

while ((!digitcheck(a.Give.year)) || a.Give.year < 1 || a.Give.year > 2018)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.year;

}

///////////////////////////////////

cout << "Введите дату возврата книги" << endl;

cout << "День: ";

cin >> a.Back.day;

while ((!digitcheck(a.Back.day)) || a.Back.day < 1 || a.Back.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.day;

}

cout << "Месяц: ";

cin >> a.Back.month;

while ((!digitcheck(a.Back.month)) || a.Back.month < 1 || a.Back.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.month;

}

cout << "Год: ";

cin >> a.Back.year;

while ((!digitcheck(a.Back.year)) || a.Back.year < 1 || a.Back.year > 2018 || (a.Back.day < a.Give.day && a.Back.month <= a.Give.month && a.Back.year <= a.Give.year) || (a.Back.month < a.Give.month && a.Back.year <= a.Give.year) || (a.Back.year < a.Give.year))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.year;

}

cout << endl;

}

void out(LibraryUser a)

{

cout << "Номер книги - " << a.iNumber << endl;

cout << "Автор книги - " << a.Author << endl;

cout << "Название книги - " << a.Title << endl;

cout << "Дата выдачи - " << a.Give.day << "." << a.Give.month << "." << a.Give.year << endl;

cout << "Дата возврата - " << a.Back.day << "." << a.Back.month << "." << a.Back.year << endl;

}

void inpV(LibraryUser a, vector<LibraryUser> &b)

{

b.push\_back(a);

}

void sort(vector<LibraryUser> &a)

{

int min;

for (unsigned int i = 0; i < (a.size() - 1); i++)

{

min = i;

for (unsigned int j = i + 1; j < a.size(); j++)

{

if (((a[j].Give.day < a[min].Give.day) && (a[j].Give.month == a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.month < a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.year < a[min].Give.year)))

min = j;

}

swap(a[i], a[min]);

}

}

void outV(vector<LibraryUser> a)

{

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

{

cout << (i + 1) << ")" << endl;

cout << "Номер книги - " << a[i].iNumber << endl;

cout << "Автор книги - " << a[i].Author << endl;

cout << "Название книги - " << a[i].Title << endl;

cout << "Дата выдачи - " << a[i].Give.day << "." << a[i].Give.month << "." << a[i].Give.year << endl;

cout << "Дата возврата - " << a[i].Back.day << "." << a[i].Back.month << "." << a[i].Back.year << endl << endl;

}

}

void Not\_back(vector<LibraryUser> a, vector<LibraryUser> &b)

{

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

{

if (!((a[i].Back.day >= a[i].Give.day && a[i].Back.month == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && (a[i].Back.month - 1) == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && a[i].Back.month == 1 && a[i].Give.month == 12 && (a[i].Back.year - 1) == a[i].Give.year)))

b.push\_back(a[i]);

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

LibraryUser TUser;

vector<LibraryUser> Users;

vector<LibraryUser> BadUsers;

bool flag = false;

bool flagV = false;

int answer = 100;

while (answer != 0)

{

system("cls");

cout << "Лабораторная работа №4 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Меню" << endl;

cout << "1) Запись отдельной книги" << endl;

cout << "2) Вывод отдельной книги" << endl;

cout << "///////////////////////////////////" << endl;

cout << "3) Добавление книги в общий список" << endl;

cout << "4) Сортировка книг по дате выдачи" << endl;

cout << "5) Вывести общий список книг" << endl;

cout << "6) Список книг, которые не возвращены в месячный срок" << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> answer;

system("cls");

cout << "Лабораторная работа №4 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (answer)

{

case 1: //1 Задание

{

inp(TUser);

flag = true;

system("pause");

break;

}

case 2: //2 Задание

{

if (flag)

out(TUser);

else

cout << "Нечего выводить \n Введите книгу";

system("pause");

break;

}

case 3: //3 Задание

{

if (flag)

{

inpV(TUser, Users);

flag = false;

flagV = true;

}

else

cout << "Нечего добавлять \n Введите книгу";

system("pause");

break;

}

case 4: //4 Задание

{

if (Users.size() > 1)

sort(Users);

else

cout << "Слишком мало элементов для сортировки \n Добавьте хотя бы еще одну книгу" << endl;

system("pause");

break;

}

case 5: //5 Задание

{

cout << "Книги в списке: " << endl;

outV(Users);

system("pause");

break;

}

case 6: //6 Задание

{

if (flagV)

{

Not\_back(Users, BadUsers);

if (BadUsers.size() >= 1)

outV(BadUsers);

else

cout << "Все вернули вовремя" << endl;

}

else

{

cout << "Список пуст" << endl;

}

BadUsers.clear();

system("pause");

break;

}

default:

break;

}

}

system("pause");

return 0;

}

# **Задание 2**

# **Условие задания**

Разбить первое задание на модули.

# **Реализация модулей**

**Unit\_Def.h**

#pragma once

#include <iostream>

#include <string>

#include <vector>

#include <Windows.h>

using namespace std;

struct Date

{

unsigned short day;

unsigned short month;

unsigned short year;

};

struct LibraryUser

{

unsigned int iNumber;

string Author;

string Title;

Date Give;

Date Back;

};

**Unit\_Operation.h**

#pragma once

#include "Unit\_Def.h"

void inpV(LibraryUser a, vector<LibraryUser> &b)

{

b.push\_back(a);

}

void sort(vector<LibraryUser> &a)

{

int min;

for (unsigned int i = 0; i < (a.size() - 1); i++)

{

min = i;

for (unsigned int j = i + 1; j < a.size(); j++)

{

if (((a[j].Give.day < a[min].Give.day) && (a[j].Give.month == a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.month < a[min].Give.month) && (a[j].Give.year == a[min].Give.year)) || ((a[j].Give.year < a[min].Give.year)))

min = j;

}

swap(a[i], a[min]);

}

}

void Not\_back(vector<LibraryUser> a, vector<LibraryUser> &b)

{

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

{

if (!((a[i].Back.day >= a[i].Give.day && a[i].Back.month == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && (a[i].Back.month - 1) == a[i].Give.month && a[i].Back.year == a[i].Give.year) || (a[i].Back.day <= a[i].Give.day && a[i].Back.month == 1 && a[i].Give.month == 12 && (a[i].Back.year - 1) == a[i].Give.year)))

b.push\_back(a[i]);

}

}

**Unit\_Input\_Output.h**

#pragma once

#include "Unit\_Def.h"

bool digitcheck(unsigned int a)

{

cin.clear();

cin.sync();

string s;

getline(cin, s, '\n');

for (auto x : s)

{

if (!isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

bool spellcheck(string a)

{

cin.clear();

cin.sync();

for (auto x : a)

{

if (isdigit(x))

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

void inp(LibraryUser &a) //Ввод структуры

{

cout << "Введиите номер книги" << endl;

cin >> a.iNumber;

while ((!digitcheck(a.iNumber)) || a.iNumber < 1 || a.iNumber > 65535)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.iNumber;

}

///////////////////////////////////

cout << "Введите автора книги (Без пробелов!!!)" << endl;

cin >> a.Author;

while (!spellcheck(a.Author))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Author;

}

///////////////////////////////////

cout << "Введите название книги (Без пробелов!!!)" << endl;

cin >> a.Title;

while (!spellcheck(a.Title))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Title;

}

///////////////////////////////////

cout << "Введите дату выдачи книги" << endl;

cout << "День: ";

cin >> a.Give.day;

while ((!digitcheck(a.Give.day)) || a.Give.day < 1 || a.Give.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.day;

}

cout << "Месяц: ";

cin >> a.Give.month;

while ((!digitcheck(a.Give.month)) || a.Give.month < 1 || a.Give.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.month;

}

cout << "Год: ";

cin >> a.Give.year;

while ((!digitcheck(a.Give.year)) || a.Give.year < 1 || a.Give.year > 2018)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Give.year;

}

///////////////////////////////////

cout << "Введите дату возврата книги" << endl;

cout << "День: ";

cin >> a.Back.day;

while ((!digitcheck(a.Back.day)) || a.Back.day < 1 || a.Back.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.day;

}

cout << "Месяц: ";

cin >> a.Back.month;

while ((!digitcheck(a.Back.month)) || a.Back.month < 1 || a.Back.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.month;

}

cout << "Год: ";

cin >> a.Back.year;

while ((!digitcheck(a.Back.year)) || a.Back.year < 1 || a.Back.year > 2018 || (a.Back.day < a.Give.day && a.Back.month <= a.Give.month && a.Back.year <= a.Give.year) || (a.Back.month < a.Give.month && a.Back.year <= a.Give.year) || (a.Back.year < a.Give.year))

{

cout << "Попробуйте еще раз" << endl;

cin >> a.Back.year;

}

cout << endl;

}

void out(LibraryUser a)

{

cout << "Номер книги - " << a.iNumber << endl;

cout << "Автор книги - " << a.Author << endl;

cout << "Название книги - " << a.Title << endl;

cout << "Дата выдачи - " << a.Give.day << "." << a.Give.month << "." << a.Give.year << endl;

cout << "Дата возврата - " << a.Back.day << "." << a.Back.month << "." << a.Back.year << endl;

}

void outV(vector<LibraryUser> a)

{

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

{

cout << (i + 1) << ")" << endl;

cout << "Номер книги - " << a[i].iNumber << endl;

cout << "Автор книги - " << a[i].Author << endl;

cout << "Название книги - " << a[i].Title << endl;

cout << "Дата выдачи - " << a[i].Give.day << "." << a[i].Give.month << "." << a[i].Give.year << endl;

cout << "Дата возврата - " << a[i].Back.day << "." << a[i].Back.month << "." << a[i].Back.year << endl << endl;

}

}

# **Кодирование алгоритма программы**

#include "stdafx.h"

#include "Unit\_Def.h"

#include "Unit\_Operation.h"

#include "Unit\_Input\_Output.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

LibraryUser TUser;

vector<LibraryUser> Users;

vector<LibraryUser> BadUsers;

bool flag = false;

bool flagV = false;

int answer = 100;

while (answer != 0)

{

system("cls");

cout << "Лабораторная работа №4 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Меню" << endl;

cout << "1) Запись отдельной книги" << endl;

cout << "2) Вывод отдельной книги" << endl;

cout << "///////////////////////////////////" << endl;

cout << "3) Добавление книги в общий список" << endl;

cout << "4) Сортировка книг по дате выдачи" << endl;

cout << "5) Вывести общий список книг" << endl;

cout << "6) Список книг, которые не возвращены в месячный срок" << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> answer;

system("cls");

cout << "Лабораторная работа №4 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (answer)

{

case 1: //1 Задание

{

inp(TUser);

flag = true;

system("pause");

break;

}

case 2: //2 Задание

{

if (flag)

out(TUser);

else

cout << "Нечего выводить \n Введите книгу";

system("pause");

break;

}

case 3: //3 Задание

{

if (flag)

{

inpV(TUser, Users);

flag = false;

flagV = true;

}

else

cout << "Нечего добавлять \n Введите книгу";

system("pause");

break;

}

case 4: //4 Задание

{

if (Users.size() > 1)

sort(Users);

else

cout << "Слишком мало элементов для сортировки \n Добавьте хотя бы еще одну книгу" << endl;

system("pause");

break;

}

case 5: //5 Задание

{

cout << "Книги в списке: " << endl;

outV(Users);

system("pause");

break;

}

case 6: //6 Задание

{

if (flagV)

{

Not\_back(Users, BadUsers);

if (BadUsers.size() >= 1)

outV(BadUsers);

else

cout << "Все вернули вовремя" << endl;

}

else

{

cout << "Список пуст" << endl;

}

BadUsers.clear();

system("pause");

break;

}

default:

break;

}

}

system("pause");

return 0;

}