|  |  |
| --- | --- |
| 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 №5** **Тема работы «Создание простого класса, его реализация и применение»** | |
| Выполнил: |  |
| Студент группы ИКБО-07-17  Отчет принял:  Преподаватель кафедры МОСИТ | Данилов Д.Е.  Миронов Антон Николаевич |
| Москва 2017 | |

**Вариант 7**

**Задание 1. Создание базового класса**

# Класс - Запись в Карточке пользователя библиотеки

# Поля: Инвентарный номер, Автор, Название, Дата выдачи, Дата возврата

# Дополнительные операции:

# 1) По инвентарному номеру вернуть название и дату возврата книги.

# 2) Определить, один ли автор у двух книг.

**Задание 2. Создание класса наследника и его тестирование**

# Класс – запись о книге вида

# Поля: Вид книги – художественная, жанр (параметр); Вид книги – техническая, справочник или словарь (параметр).

# Дополнительные операции:

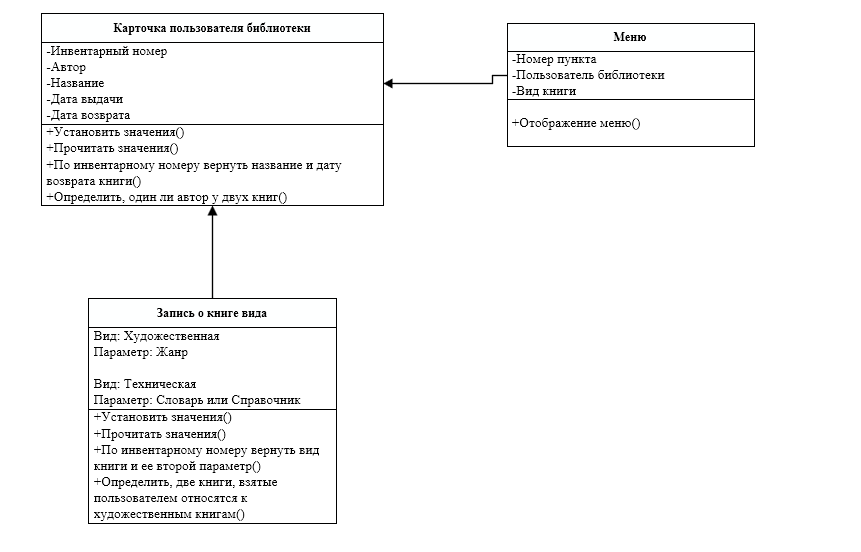
# 3) По инвентарному номеру вернуть вид книги и ее второй параметр.

# 4) Определить, две книги, взятые пользователем, относятся к художественным книгам.

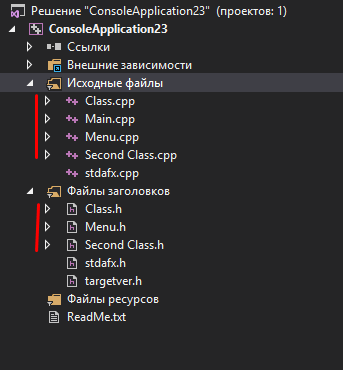
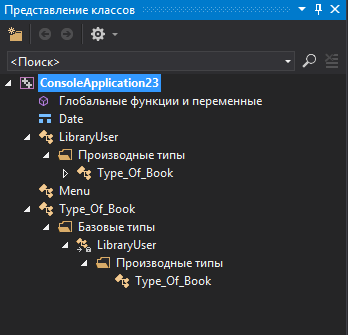
**Задание 3. Разработать класс – Меню, для демонстрации работы с объектами обоих классов и выполнением всех дополнительных операций над ними.**

**Задание 4. Разработать диаграмму классов с указанием полей, методов, и отношений между классами**

**Диаграмма Классов**

****

**Структура проекта**

****

**Реализация класса (Пользователь библиотеки)**

**Заголовочный файл**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

bool digitcheck(unsigned int a);

bool spellcheck(string a);

struct Date

{

unsigned short day;

unsigned short month;

unsigned short year;

};

class LibraryUser

{

public:

LibraryUser();

LibraryUser(int N, string A, string T, unsigned short day1, unsigned short month1, unsigned short year1, unsigned short day2, unsigned short month2, unsigned short year2);

~LibraryUser();

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

void set\_Number();

void set\_Author();

void set\_Title();

void set\_Give();

void set\_Back();

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

int get\_Number();

string get\_Author();

string get\_Title();

Date get\_Give();

Date get\_Back();

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

void Out\_Title\_Back(vector <LibraryUser> Array);

void Inp\_v(vector <LibraryUser> &Array);

void Out\_v(LibraryUser m);

bool Is\_Num\_in\_vector(int Num, vector <LibraryUser> Array);

void solo\_Author(vector <LibraryUser> Array);

private:

int a;

string s;

string s1;

Date d;

Date d1;

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

int Num;

string Author;

string Title;

Date Give;

Date Back;

};

**Реализация методов этого класса**

#include "stdafx.h"

#include "Class.h"

#include <vector>

using namespace std;

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 ((x < 'A' || x > 'Z') && (x < 'a' || x > 'z') && (x < 'А' || x > 'Я') && (x < 'а' || x > 'я') && x != 'Ё'&& x != 'ё')

{

cin.clear();

cin.sync();

return false;

}

}

return true;

}

LibraryUser :: LibraryUser()

{

}

LibraryUser :: LibraryUser(int N, string A, string T, unsigned short day1, unsigned short month1, unsigned short year1, unsigned short day2, unsigned short month2, unsigned short year2)

{

Num = N;

Author = A;

Title = T;

Give.day = day1;

Give.month = month1;

Give.year = year1;

Back.day = day2;

Back.month = month2;

Back.year = year2;

}

LibraryUser :: ~LibraryUser()

{

}

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

void LibraryUser::set\_Number()

{

cout << "Введите номер книги" << endl;

cin >> a;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> a;

}

Num = a;

}

void LibraryUser:: set\_Author()

{

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

cin >> s;

while (!spellcheck(s))

{

cout << "Попробуйте еще раз" << endl;

cin >> s;

}

Author = s;

}

void LibraryUser:: set\_Title()

{

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

cin >> s1;

while (!spellcheck(s1))

{

cout << "Попробуйте еще раз" << endl;

cin >> s1;

}

Title = s1;

}

void LibraryUser::set\_Give()

{

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

cout << "День: ";

cin >> d.day;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> d.day;

}

cout << "Месяц: ";

cin >> d.month;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> d.month;

}

cout << "Год: ";

cin >> d.year;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> d.year;

}

Give = d;

}

void LibraryUser::set\_Back()

{

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

cout << "День: ";

cin >> d1.day;

while ((!digitcheck(d1.day)) || d1.day < 1 || d1.day > 31)

{

cout << "Попробуйте еще раз" << endl;

cin >> d1.day;

}

cout << "Месяц: ";

cin >> d1.month;

while ((!digitcheck(d1.month)) || d1.month < 1 || d1.month > 12)

{

cout << "Попробуйте еще раз" << endl;

cin >> d1.month;

}

cout << "Год: ";

cin >> d1.year;

while ((!digitcheck(d1.year)) || d1.year < 1 || d1.year > 2018 || (d1.day < d.day && d1.month <= d.month && d1.year <= d.year) || (d1.month < d.month && d1.year <= d.year) || (d1.year < d.year))

{

cout << "Попробуйте еще раз" << endl;

cin >> d1.year;

}

cout << endl;

Back = d1;

}

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

int LibraryUser::get\_Number()

{

return Num;

}

string LibraryUser::get\_Author()

{

return Author;

}

string LibraryUser::get\_Title()

{

return Title;

}

Date LibraryUser::get\_Give()

{

return Give;

}

Date LibraryUser::get\_Back()

{

return Back;

}

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

void LibraryUser::Inp\_v(vector <LibraryUser> &Array)

{

LibraryUser a;

a.set\_Number();

a.set\_Author();

a.set\_Title();

a.set\_Give();

a.set\_Back();

Array.push\_back(a);

}

void LibraryUser :: Out\_v(LibraryUser m)

{

cout << m.get\_Number() << ", " << m.get\_Author() << ", " << m.get\_Title() << endl;

cout << m.get\_Give().day << "." << m.get\_Give().month << "." << m.get\_Give().year << endl;

cout << m.get\_Back().day << "." << m.get\_Back().month << "." << m.get\_Back().year << endl << endl;

}

void LibraryUser::Out\_Title\_Back(vector <LibraryUser> Array)

{

int a;

cout << "Введите инвентарный номер книги" << endl;

cin >> a;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> a;

}

for each (LibraryUser i in Array)

{

if (i.get\_Number() == a)

{

cout << "Автор этой книги: ";

cout << i.get\_Author() << endl;

cout << "Дата возврата этой книги: ";

cout << i.get\_Back().day << "." << i.get\_Back().month << "." << i.get\_Back().year << endl;

}

}

}

bool LibraryUser::Is\_Num\_in\_vector(int Num, vector <LibraryUser> Array)

{

for each (LibraryUser i in Array)

{

if (Num == i.get\_Number())

return true;

}

return false;

}

void LibraryUser::solo\_Author(vector <LibraryUser> Array)

{

int a, i1;

string s;

for (i1 = 0; i1 < 2; i1++)

{

cout << "Введите инвентарный номер книги" << endl;

cin >> a;

while ((!digitcheck(a)) || a < 1 || a > 65535 || (!Is\_Num\_in\_vector(a, Array)))

{

cout << "Попробуйте еще раз" << endl;

cin >> a;

}

for each (LibraryUser i in Array)

{

if (i.get\_Number() == a)

if (i1 == 1)

if (s == i.get\_Author())

cout << "У этих книг один автор" << endl;

else

cout << "У этих книг разные авторы" << endl;

else

s = i.get\_Author();

}

}

}

**Реализация класса (Вид книги – наследник класса Пользователь библиотеки)**

**Заголовочный файл**

#include "Class.h"

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Type\_Of\_Book : LibraryUser

{

public:

Type\_Of\_Book();

Type\_Of\_Book(int N, string A, string T, unsigned short day1, unsigned short month1, unsigned short year1, unsigned short day2, unsigned short month2, unsigned short year2);

~Type\_Of\_Book();

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

void set\_Type();

void set\_Option();

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

string get\_Type();

string get\_Option();

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

void Out\_Type\_Option(vector <Type\_Of\_Book> &Array);

bool Is\_Num\_in\_vector\_v(int Num, vector <Type\_Of\_Book> Array);

void solo\_Art(vector <Type\_Of\_Book> &Array);

void Inp\_w(vector <Type\_Of\_Book> &Array);

void Out\_w(Type\_Of\_Book m);

private:

string r;

string t;

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

string Type;

string Option;

};

**Реализация методов этого класса**

#include "stdafx.h"

#include "Class.h"

#include "Second Class.h"

#include <vector>

using namespace std;

Type\_Of\_Book::Type\_Of\_Book()

{

Type = "";

Option = "";

}

Type\_Of\_Book::Type\_Of\_Book(int N, string A, string T, unsigned short day1, unsigned short month1, unsigned short year1, unsigned short day2, unsigned short month2, unsigned short year2) : LibraryUser(N,A,T,day1,month1,year1,day2,month2,year2)

{

Type = "";

Option = "";

}

Type\_Of\_Book::~Type\_Of\_Book()

{

}

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

void Type\_Of\_Book::set\_Type()

{

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

cin >> r;

while (!spellcheck(r) || (r!="Художественная" && r != "Техническая" && r != "художественная" && r != "техническая"))

{

cout << "Попробуйте еще раз" << endl;

cin >> r;

}

Type = r;

}

void Type\_Of\_Book::set\_Option()

{

if (Type == "Художественная" || Type == "художественная")

{

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

cin >> t;

while (!spellcheck(t))

{

cout << "Попробуйте еще раз" << endl;

cin >> t;

}

Option = t;

}

else

{

cout << "Словарь или Справочник?" << endl;

cin >> t;

while (!spellcheck(t) || (t != "Словарь" && t != "Справочник" && t != "cловарь" && t != "cправочник"))

{

cout << "Попробуйте еще раз" << endl;

cin >> t;

}

Option = t;

}

}

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

string Type\_Of\_Book::get\_Type()

{

return Type;

}

string Type\_Of\_Book::get\_Option()

{

return Option;

}

void Type\_Of\_Book::Out\_Type\_Option(vector <Type\_Of\_Book> &Array)

{

int a;

cout << "Введите инвентарный номер книги" << endl;

cin >> a;

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

{

cout << "Попробуйте еще раз" << endl;

cin >> a;

}

for each (Type\_Of\_Book i in Array)

{

if (i.get\_Number() == a)

{

cout << "Вид этой книги: ";

cout << i.get\_Type() << endl;

cout << "Второй параметр этой книги: ";

cout << i.get\_Option() << endl;

}

}

}

bool Type\_Of\_Book::Is\_Num\_in\_vector\_v(int Num, vector <Type\_Of\_Book> Array)

{

for each (Type\_Of\_Book i in Array)

{

if (Num == i.get\_Number())

return true;

}

return false;

}

void Type\_Of\_Book::solo\_Art(vector <Type\_Of\_Book> &Array)

{

int a, i1;

string s;

for (i1 = 0; i1 < 2; i1++)

{

cout << "Введите инвентарный номер книги" << endl;

cin >> a;

while ((!digitcheck(a)) || a < 1 || a > 65535 || (!Is\_Num\_in\_vector\_v(a, Array)))

{

cout << "Попробуйте еще раз" << endl;

cin >> a;

}

for each (Type\_Of\_Book i in Array)

{

if (i.get\_Number() == a)

if (i1 == 1)

if ((i.get\_Type() == "Художественная" || i.get\_Type() == "художественная") && s == i.get\_Type())

cout << "Обе эти книги художественные" << endl;

else

cout << "Одна или две из этих книг не художественные" << endl;

else

s = i.get\_Type();

}

}

}

void Type\_Of\_Book::Inp\_w(vector <Type\_Of\_Book> &Array)

{

Type\_Of\_Book a;

a.set\_Number();

a.set\_Author();

a.set\_Title();

a.set\_Give();

a.set\_Back();

a.set\_Type();

a.set\_Option();

Array.push\_back(a);

}

void Type\_Of\_Book::Out\_w(Type\_Of\_Book m)

{

cout << m.get\_Number() << ", " << m.get\_Author() << ", " << m.get\_Title() << endl;

cout << m.get\_Give().day << "." << m.get\_Give().month << "." << m.get\_Give().year << endl;

cout << m.get\_Back().day << "." << m.get\_Back().month << "." << m.get\_Back().year << endl;

cout << m.get\_Type() << ", " << m.get\_Option() << endl << endl;

}

**Реализация класса (Меню – управляющая программа)**

**Заголовочный файл**

#include "Class.h"

#include "Second Class.h"

using namespace std;

class Menu

{

public:

Menu();

~Menu();

void show\_Menu();

private:

};

**Реализация методов этого класса**

#include "stdafx.h"

#include "Class.h"

#include "Second Class.h"

#include "Menu.h"

#include <vector>

#include <Windows.h>

using namespace std;

Menu::Menu()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

}

Menu::~Menu()

{

}

void Menu::show\_Menu()

{

LibraryUser User\_3{ 25, "B.B.Bee", "Honey", 25, 1, 1999, 24, 2, 1999 };

LibraryUser w;

Type\_Of\_Book v;

vector<Type\_Of\_Book> Array1;

vector <LibraryUser> Array;

int ans = 10;

while (ans != 0)

{

system("cls");

cout << "Лабораторная работа №5 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Задание 1:" << endl;

cout << "Меню" << endl;

cout << "1) Добавить объект" << endl;

cout << "2) Вывести заранее созданный объект(с помощью конструктора с параметрами)" << endl;

cout << "3) Вывести все добавленные объекты" << endl;

cout << "Дополнительные операции:" << endl;

cout << "4) По инвентарному номеру вернуть название и дату возврата книги" << endl;

cout << "5) Определить, один ли автор у двух книг" << endl << endl;

cout << "Задание 2:" << endl;

cout << "Меню" << endl;

cout << "6) Добавить объект" << endl;

cout << "7) Вывести все добавленные объекты" << endl;

cout << "Дополнительные операции:" << endl;

cout << "8) По инвентарному номеру вернуть вид и второй параметр книги" << endl;

cout << "9) Определить, относятся две выбранные книги к художественному виду" << endl << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> ans;

system("cls");

cout << "Лабораторная работа №5 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (ans)

{

case 1:

{

w.Inp\_v(Array);

system("pause");

break;

}

case 2:

{

w.Out\_v(User\_3);

system("pause");

break;

}

case 3:

{

for each (LibraryUser i in Array)

{

w.Out\_v(i);

}

system("pause");

break;

}

case 4:

{

w.Out\_Title\_Back(Array);

system("pause");

break;

}

case 5:

{

w.solo\_Author(Array);

system("pause");

break;

}

case 6:

{

v.Inp\_w(Array1);

system("pause");

break;

}

case 7:

{

for each (Type\_Of\_Book i in Array1)

{

v.Out\_w(i);

}

system("pause");

break;

}

case 8:

{

v.Out\_Type\_Option(Array1);

system("pause");

break;

}

case 9:

{

v.solo\_Art(Array1);

system("pause");

break;

}

default:

break;

}

}

system("pause");

}

**Основная программа**

#include "stdafx.h"

#include "Class.h"

#include "Second Class.h"

#include "Menu.h"

#include <vector>

int main()

{

Menu m;

m.show\_Menu();

return 0;

}