Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

По лабораторной работе №1

по дисциплине: «Объектно - ориентированное программирование»

на тему «Работа с классами и объектами на языке C++»

Выполнили студенты группы 19ВВ2:

Муромский Д.А.

Кобзев М.И.

Горбунов Д.А.

Приняли:

Евсеева Ю.И.

Гудков А.А.

Пенза 2021

**Цель работы:** Изучить основы работы с классами и объектами на языке C++.

**Задания**

Разработать классы для описанных ниже объектов. Включить в класс конструктор по умолчанию, конструктор с параметрами, конструктор копирования и деструктор. Поля класса определить с модификатором доступа private, для доступа к ним реализовать методы Set- и Get- с модификатором доступа public.

8. **Patient**: Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив объектов. Вывести:

а) список пациентов, имеющих данный диагноз;

б) список пациентов, номер медицинской карты которых находится в заданном интервале.

**Практическая часть**

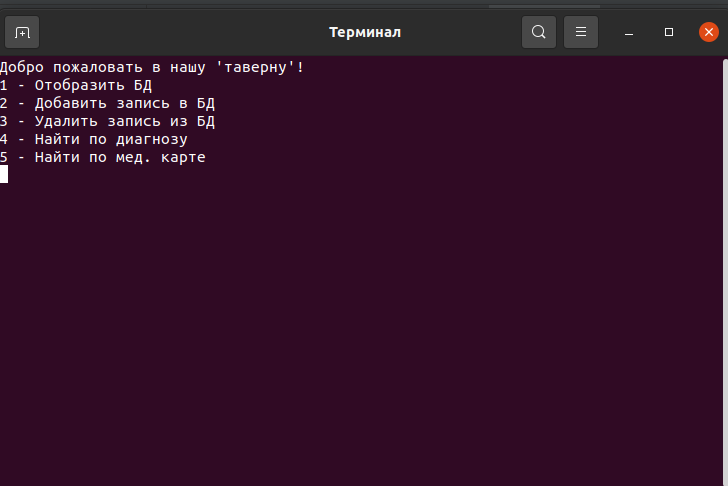


Image 1 - При запуске программы выводится консольное меню.

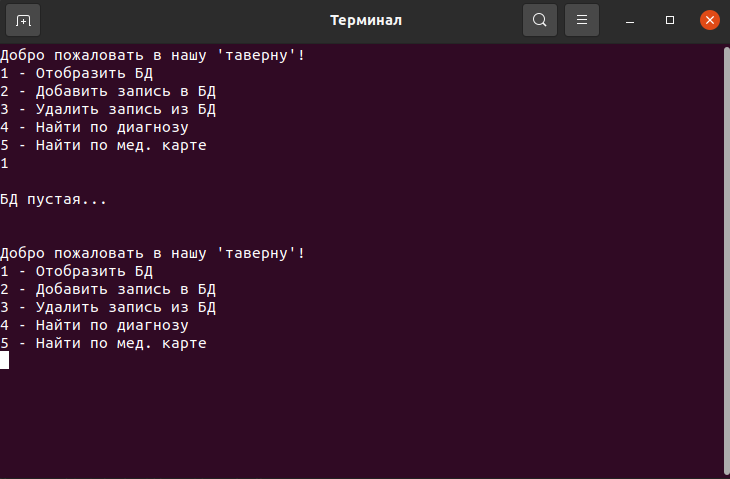


Image 2 - Отображение в консоль при пустой БД.

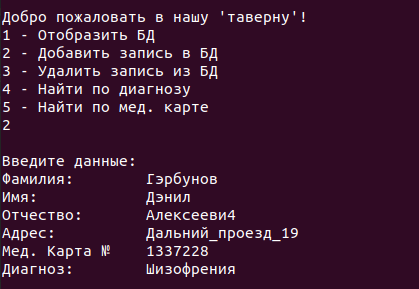


Image 3 - Добавление записи в БД.

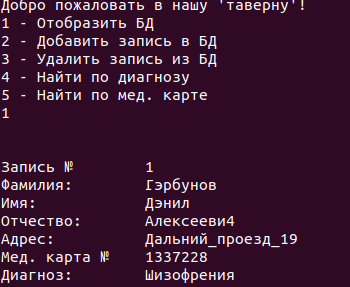


Image 4 - Отображение в консоли при имеющихся в БД записях.

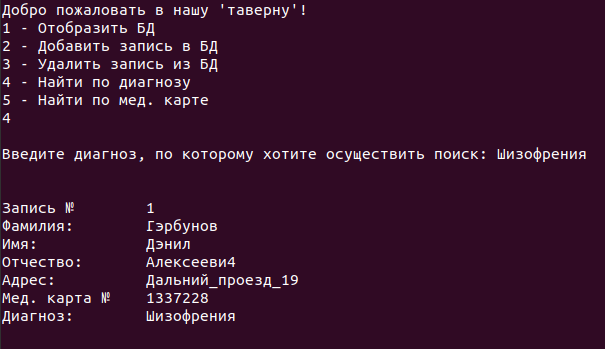


Image 5 - Поиск по диагнозу.

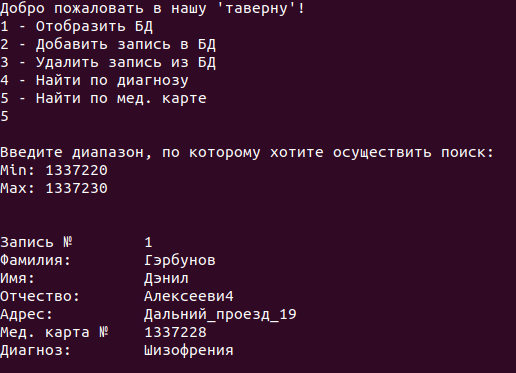


Image 6 - Поиск по мед. карте.

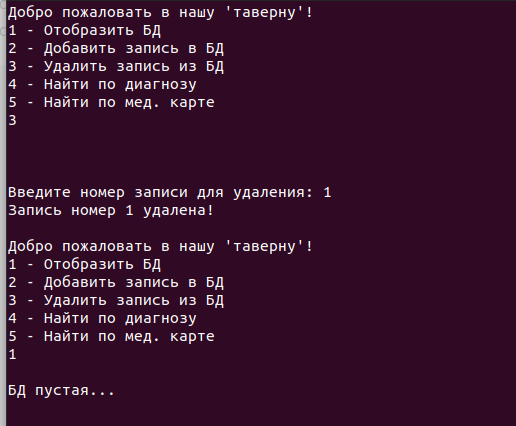


Image 7 - Удаление записи.

**Листинг class.h**

//

// Created by w3sti on 17.02.2021.

//

#ifndef LAB1\_CLASS\_H

#define LAB1\_CLASS\_H

#endif //LAB1\_CLASS\_H

#include <vector>

#include <string>

using namespace std;

class patient {

public:

string get\_last\_name() { return last\_name; }

string get\_name() { return name; }

string get\_papich() { return papich; }

string get\_address() { return address; }

long long int get\_number() { return number; }

string get\_shiza() { return shiza; }

void set\_last\_name(string a) { last\_name = a;}

void set\_name(string a) { name = a; }

void set\_papich(string a) { papich = a; }

void set\_address(string a) { address = a; }

void set\_number(long long int a) { number = a; }

void set\_shiza(string a) { shiza = a; }

patient()

{

}

patient(string a, string b, string c, string d, long long int e, string f)

: last\_name(a), name(b), papich(c), address(d), number(e), shiza(f)

{

}

patient(const patient& object)

: last\_name(object.last\_name),

name(object.name),

papich(object.papich),

address(object.address),

number(object.number),

shiza(object.shiza)

{

}

~patient() {

}

private:

string last\_name;

string name;

string papich;

string address;

long long int number = 0;

string shiza;

};

**Листинг funk.cpp**

//

// Created by w3sti on 17.02.2021.

//

#include "func.h"

#include <iostream>

vector<patient> list;

void record(int id, patient& object) {

cout << endl;

cout << "Запись №\t" << id + 1 << endl;

cout << "Фамилия:\t" << object.get\_last\_name() << endl;

cout << "Имя:\t\t" << object.get\_name() << endl;

cout << "Отчество:\t" << object.get\_papich() << endl;

cout << "Адрес:\t\t" << object.get\_address() << endl;

cout << "Мед. карта №\t" << object.get\_number() << endl;

cout << "Диагноз:\t" << object.get\_shiza() << endl;

}

void vivod() {

if (list.size()) {

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

record(i, list[i]);

}

}

else {

cout << "БД пустая..." << endl;

}

cout << endl;

cout << endl;

mainmenu();

}

void add\_record() {

string tempstr;

patient temp;

cout << "Введите данные: " << endl;

cout << "Фамилия:\t";

cin >> tempstr;

temp.set\_last\_name(tempstr);

cout << "Имя:\t\t";

cin >> tempstr;

temp.set\_name(tempstr);

cout << "Отчество:\t";

cin >> tempstr;

temp.set\_papich(tempstr);

cout << "Адрес:\t\t";

cin >> tempstr;

temp.set\_address(tempstr);

cout << "Мед. Карта №\t";

int tempint;

cin >> tempint;

temp.set\_number(tempint);

cout << "Диагноз:\t";

cin >> tempstr;

temp.set\_shiza(tempstr);

//list.push\_back(temp->get\_last\_name(),temp->get\_name(),temp->get\_papich(),temp->get\_address(),temp->get\_number(),temp->get\_shiza());

list.push\_back(temp);

cout << endl;

cout << endl;

mainmenu();

}

void delete\_record() {

cout << endl;

cout << endl;

cout << "Введите номер записи для удаления: ";

int a;

cin >> a;

list.erase(list.begin() + a - 1);

cout << "Запись номер " << a << " удалена!";

cout << endl;

cout << endl;

mainmenu();

}

void seach\_shiza\_record() {

cout << "Введите диагноз, по которому хотите осуществить поиск: ";

string tempstr;

cin >> tempstr;

vector<int> abc;

if (!list.empty()) {

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

if (list[i].get\_shiza() == tempstr) {

abc.push\_back(i);

}

}

}

cout << endl;

if (abc.empty()) {

cout << "Записи не найдены!" << endl;

}

else {

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

record(abc[i], list[abc[i]]);

}

}

cout << endl;

cout << endl;

mainmenu();

}

void seach\_number\_record() {

cout << "Введите диапазон, по которому хотите осуществить поиск: " << endl;

cout << "Min: ";

long long int min, max;

cin >> min;

cout << "Max: ";

cin >> max;

vector<int> zxc;

if (!list.empty()) {

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

if ((list[i].get\_number() >= min) && (list[i].get\_number() <= max)) {

zxc.push\_back(i);

}

}

}

cout << endl;

if (zxc.empty()) {

cout << "Записи не найдены!" << endl;

}

else {

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

record(zxc[i], list[zxc[i]]);

}

}

cout << endl;

cout << endl;

mainmenu();

}

void mainmenu() {

//SetConsoleCP(1251);

//SetConsoleOutputCP(1251);

cout << "Добро пожаловать в нашу 'таверну'!" << endl;

cout << "1 - Отобразить БД" << endl;

cout << "2 - Добавить запись в БД" << endl;

cout << "3 - Удалить запись из БД" << endl;

cout << "4 - Найти по диагнозу" << endl;

cout << "5 - Найти по мед. карте" << endl;

int number;

cin >> number;

cout << endl;

switch (number) {

case 1:

vivod();

break;

case 2:

add\_record();

break;

case 3:

delete\_record();

break;

case 4:

seach\_shiza\_record();

break;

case 5:

seach\_number\_record();

break;

}

}

**Вывод:** Изучили основы работы с классами и объектами на языке C++.