МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ»

***Группа 82 ТП***

**Отчет**

**По лабораторной работе №22**

# “ИСПОЛЬЗОВАНИЕ КОНСТРУКТОРА И ДЕСТРУКТОРА ПРИ ПРОЕКТИРОВАНИИ ПОЛЬЗОВАТЕЛЬСКОГО КЛАССА”

учебный предмет

«Основы алгоритмизации и программирования»

**Исполнитель: А. К. Шатуха**

**Руководитель: Д. А. Федкевич**

Выполнение заданий:

1) Задание 1,2: Динамические данные.

Задайте динамический массив экземпляров класса в ЛР №20.

Листинг или ссылка на проект:

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Person {

private:

string surname;

string name;

string fathername;

string address;

string gender;

string education;

int year\_birth;

public:

Person() : surname(""), name(""), fathername(""), address(""), gender(""), education(""), year\_birth(0) {}

Person(string surname, string name, string fathername, string address, string gender, string education, int year\_birth) {

this->surname = surname;

this->name = name;

this->fathername = fathername;

this->address = address;

this->gender = gender;

this->education = education;

this->year\_birth = year\_birth;

}

Person(const Person& other) {

this->surname = other.surname;

this->name = other.name;

this->fathername = other.fathername;

this->address = other.address;

this->gender = other.gender;

this->education = other.education;

this->year\_birth = other.year\_birth;

}

~Person() {}

void setSurname(string surname) { this->surname = surname; }

void setName(string name) { this->name = name; }

void setFathername(string fathername) { this->fathername = fathername; }

void setAddress(string address) { this->address = address; }

void setGender(string gender) { this->gender = gender; }

void setEducation(string education) { this->education = education; }

void setYearBirth(int year\_birth) { this->year\_birth = year\_birth; }

string getSurname() const { return surname; }

string getName() const { return name; }

string getFathername() const { return fathername; }

string getAddress() const { return address; }

string getGender() const { return gender; }

string getEducation() const { return education; }

int getYearBirth() const { return year\_birth; }

void show() const {

cout << u8"Фамилия: " << surname << u8", Имя: " << name << u8", Отчество: " << fathername

<< u8", Адрес: " << address << u8", Пол: " << gender << u8", Образование: " << education

<< u8", Год рождения: " << year\_birth << endl;

}

};

void list\_older\_than(const vector<Person>& people, int age, int current\_year) {

cout << u8"Список граждан, возраст которых превышает " << age << u8" лет:" << endl;

for (const auto& person : people) {

if ((current\_year - person.getYearBirth()) > age) {

person.show();

}

}

}

void list\_higher\_education(const vector<Person>& people) {

cout << u8"\nСписок граждан с высшим образованием:" << endl;

for (const auto& person : people) {

if (person.getEducation() == u8"Высшее") {

person.show();

}

}

}

void list\_male(const vector<Person>& people) {

cout << u8"\nСписок граждан мужского пола:" << endl;

for (const auto& person : people) {

if (person.getGender() == u8"мужской") {

person.show();

}

}

}

int main() {

setlocale(LC\_ALL, "RU.UTF-8");

int size;

cout << u8"Введите количество граждан: ";

cin >> size;

vector<Person> people;

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

string surname, name, fathername, address, gender, education;

int year\_birth;

cout << u8"Введите данные для гражданина #" << (i + 1) << ":\n";

cout << u8"Фамилия: "; cin >> surname;

cout << u8"Имя: "; cin >> name;

cout << u8"Отчество: "; cin >> fathername;

cout << u8"Адрес: "; cin >> address;

cout << u8"Пол (мужской/женский): "; cin >> gender;

cout << u8"Образование: "; cin >> education;

cout << u8"Год рождения: "; cin >> year\_birth;

people.emplace\_back(surname, name, fathername, address, gender, education, year\_birth);

}

int age\_threshold = 20;

int current\_year = 2025;

list\_older\_than(people, age\_threshold, current\_year);

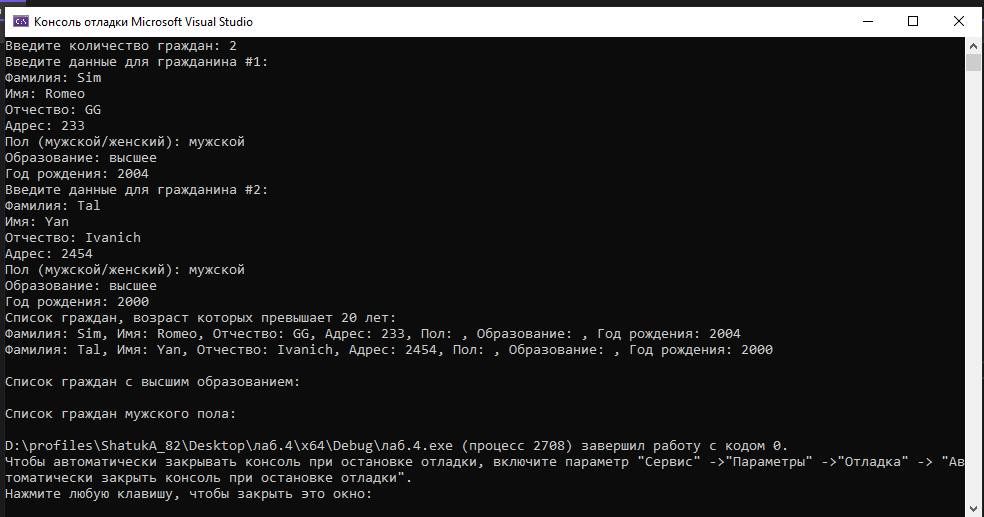
list\_higher\_education(people);

list\_male(people);

return 0;

}

Cкриншот:



(Из-за кодировки только на английском выводит корректно)