Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине «Основы алгоритмизации и программирования»

За 2 семестр

Тема: «Структуры, объединения, перечисления»

Выполнили:

Студенты 1 курса

Группы ИИ-21(1)

Кирилович А.А.

Литвинюк Т. В

Проверил:

Гирель Т. Н.

Брест 2022

**Цель:** научиться работать со структурными переменными, изучить особенности различных видов структур, а также сферы их применения.

**Ход работы**

#include<iostream>

#include <string>

using namespace std;

struct Human

{

    string surname;

    string name;

    string lastName;

    string phoneNumber;

    string homeAddress;

    float age;

};

Human\* AddStruct(Human\* Obj, const int amount);

void setData(Human\* Obj, const int amount);

void sortStruct(Human\* Obj, const int amount);

void search(Human\* Obj, const int amount, string str);

void editData(Human\* Obj, const int amount, int str);

Human\* DeleteStruct(Human\* Obj, const int amount, int strDelete);

void showData(const Human\* Obj, const int amount);

int main()

{

    system("chcp 1251 > nul");

    Human\* Persons = 0;

    int personsAmount = 0;

    int YesOrNot = 0;

    while (true)

    {

        cout << "1. Ввод массива структур.\n";

        cout << "2. Сортировка массива структур по заданному параметру.\n";

        cout << "3. Поиск в массиве структур по заданному параметру.\n";

        cout << "4. Изменение заданной структуры.\n";

        cout << "5. Удаление структуры из массива.\n";

        cout << "6. Вывод на экран массива структур.\n";

        cout << "7. Выход.\n";

        int n;

        cout << "\nВведите число (ваш выбор): ";

        cin >> n;

        cin.get();

        system("cls");

        switch (n)

        {

        case 1:// ввод

        {

            do

            {

                Persons = AddStruct(Persons, personsAmount);

                setData(Persons, personsAmount);

                personsAmount++;

                cout << "Продолжить ввод данных (1 - да, 0 - нет): ";

                cin >> YesOrNot;

                cin.get();

            } while (YesOrNot != 0);

            system("cls");

        }

            break;

        case 2:// сортировка

            sortStruct(Persons, personsAmount);

            showData(Persons, personsAmount);

            break;

        case 3:// поиск

        {

            string str;

            cout << "Поиск: ";

            getline(cin, str);

            search(Persons, personsAmount, str);

        }

            break;

        case 4:// изменение

        {

            showData(Persons, personsAmount);

            cout << "Введите номер строки, которую хотите изменить: " << endl;

            int str;

            cin >> str;

            cin.get();

            cout << "1. Фамилия.\n";

            cout << "2. Имя.\n";

            cout << "3. Отчество.\n";

            cout << "4. Номер телефона.\n";

            cout << "5. Адрес.\n";

            cout << "6. Возраст.\n";

            cout << "Введите номер атрибута, который хотите изменить: " << endl;

            int atr;

            cin >> atr;

            cin.get();

            cout << "Теперь введите нужное вам значение:" << endl;

            editData(Persons, str, atr);

        }

            break;

        case 5://Удаление

        {

            showData(Persons, personsAmount);

            cout << "\nВведите номер строки, которую хотите удалить: " << endl;

            int str;

            cin >> str;

            cin.get();

            Persons = DeleteStruct(Persons, personsAmount, str);

            personsAmount--;

        }

            break;

        case 6: //вывод

            showData(Persons, personsAmount);

            break;

        case 7://выход

            exit(0);

            break;

        default:

            cout << "Такого пункта в меню нет. Попробуйте ещё. ";

            break;

        }

    }

    delete[] Persons;

    return 0;

}

Human\* AddStruct(Human\* Obj, const int amount)

{

    if (amount == 0)

    {

        Obj = new Human[amount + 1]; // выделение памяти для первой структуры

    }

    else

    {

        Human\* tempObj = new Human[amount + 1];

        for (int i = 0; i < amount; i++)

        {

            tempObj[i] = Obj[i]; // копируем во временный объект

        }

        delete[] Obj;

        Obj = tempObj;

    }

    return Obj;

}

void setData(Human\* Obj, const int amount)

{

    cout << "Фамилия: ";

    getline(cin, Obj[amount].surname);

    cout << "Имя: ";

    getline(cin, Obj[amount].name);

    cout << "Отчество: ";

    getline(cin, Obj[amount].lastName);

    cout << "Номер телефона: ";

    getline(cin, Obj[amount].phoneNumber);

    cout << "Адрес: ";

    getline(cin, Obj[amount].homeAddress);

    cout << "Возраст: ";

    cin >> Obj[amount].age;

    cin.get();

    cout << endl;

}

void sortStruct(Human\* Obj, const int amount)

{

    cout << "1. Фамилия.\n";

    cout << "2. Имя.\n";

    cout << "3. Отчество.\n";

    cout << "4. Номер телефона.\n";

    cout << "5. Адрес.\n";

    cout << "6. Возраст.\n";

    int atr;

    cin >> atr;

    cin.get();

    Human buffer;

    switch (atr)

    {

    case 1:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].surname > Obj[j + 1].surname) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

    break;

    case 2:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].name > Obj[j + 1].name) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

    break;

    case 3:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].lastName > Obj[j + 1].lastName) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

    break;

    case 4:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].homeAddress > Obj[j + 1].homeAddress) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

        break;

    case 5:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].phoneNumber > Obj[j + 1].phoneNumber) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

    break;

    case 6:

        for (int i = 0; i < amount - 1; i++) {

            for (int j = 0; j < amount - i - 1; j++) {

                if (Obj[j].age > Obj[j + 1].age) {

                    buffer = Obj[j];

                    Obj[j] = Obj[j + 1];

                    Obj[j + 1] = buffer;

                }

            }

        }

    break;

    default:

        cout << "Такого пункта в меню нет! ";

        break;

    }

}

void search(Human\* Obj, const int amount, string str)

{

    int k = 0;

    for (int i = 0; i < amount; i++)

    {

        if (str == Obj[i].surname || str == Obj[i].name || str == Obj[i].lastName || str == Obj[i].homeAddress || str == Obj[i].phoneNumber )

        {

            cout << "№  " << "Фамилия\t" << "Имя\t" << "Отчество\t" << "Адрес\t" << "Номер телефона\t" << "Возраст\t" << endl;

            cout << "===============================================================================================" << endl;

            cout << i + 1 << "  " << Obj[i].surname << '\t' << Obj[i].name << '\t' << Obj[i].lastName << '\t';

            cout << Obj[i].homeAddress << '\t' << Obj[i].phoneNumber << '\t' << Obj[i].age << '\t' << endl;

        }

        else

        {

            k += 1;

        }

    }

    if (k == amount)

    {

        cout << "Результат не найден.\n" << endl;

    }

}

void editData(Human\* Obj, const int amount, int atr)

{

    switch (atr)

    {

    case 1:

        cout << "Фамилия: ";

        getline(cin, \*&Obj[amount-1].surname);

        break;

    case 2:

        cout << "Имя: ";

        getline(cin, \*&Obj[amount-1].name);

        break;

    case 3:

        cout << "Отчество: ";

        getline(cin, \*&Obj[amount-1].lastName);

        break;

    case 4:

        cout << "Номер телефона: ";

        getline(cin, \*&Obj[amount-1].phoneNumber);

        break;

    case 5:

        cout << "Адрес: ";

        getline(cin, \*&Obj[amount-1].homeAddress);

        break;

    case 6:

        cout << "Возраст: ";

        cin >> Obj[amount-1].age;

        cin.get();

        break;

    default:

        cout << "Такого пункта в меню нет! ";

        break;

    }

    cout << endl;

}

Human\* DeleteStruct(Human\* Obj, const int amount, int str)

{

    if (amount == 0)

    {

        cout << "ERROR! Вы еще не создали хотя бы одну структуру!";

    }

    if (amount == 1)

    {

        delete[] Obj;

    }

    if (amount > 1)

    {

        Human\* tempObj = new Human[amount - 1];

        for (int i = 0; i < amount - 1; i++)

        {

            if (i < str - 1)

            {

                tempObj[i] = Obj[i];

            }

            else

            {

                tempObj[i] = Obj[i + 1];

            }

        }

        delete[] Obj;

        Obj = tempObj;

    }

    return Obj;

}

void showData(const Human\* Obj, const int amount)

{

    system("cls");

    cout << "№  " << "Фамилия\t" << "Имя\t" << "Отчество\t" << "Адрес\t" << "Номер телефона\t" << "Возраст\t" << endl;

    cout << "===============================================================================================" << endl;

    for (int i = 0; i < amount; i++)

    {

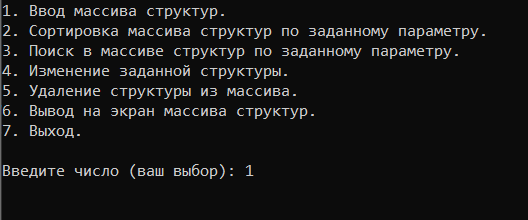
        cout << i + 1 << "  " << Obj[i].surname << '\t' << Obj[i].name << '\t' << Obj[i].lastName << '\t';

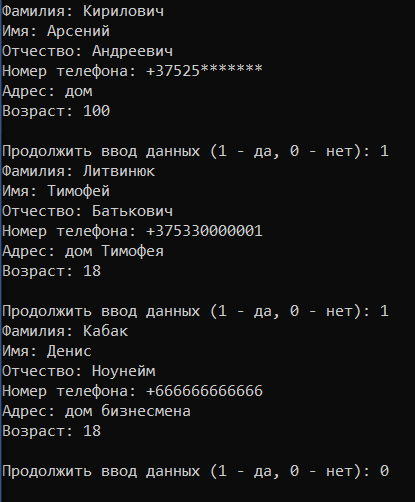
        cout << Obj[i].homeAddress << '\t' << Obj[i].phoneNumber << '\t' << Obj[i].age << '\t' << endl;

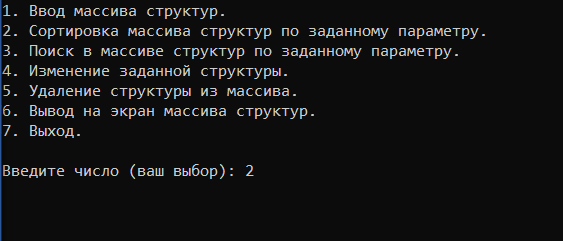
    }

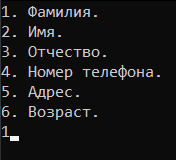
}

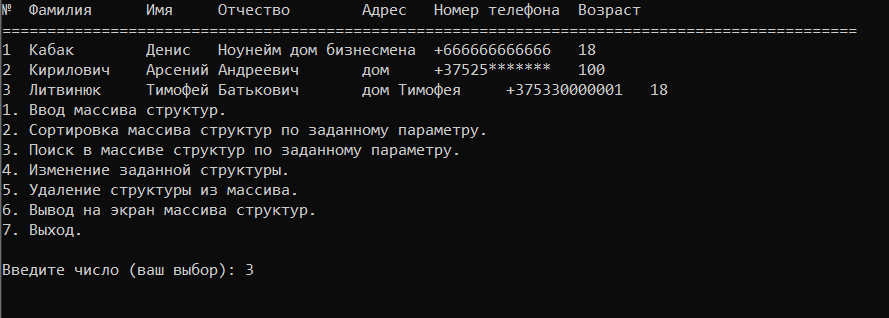
Вывод:



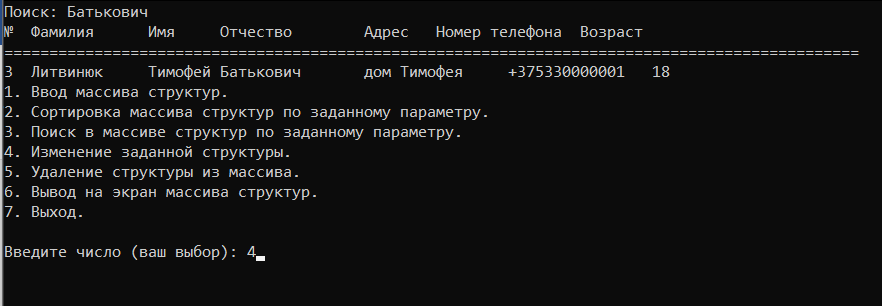


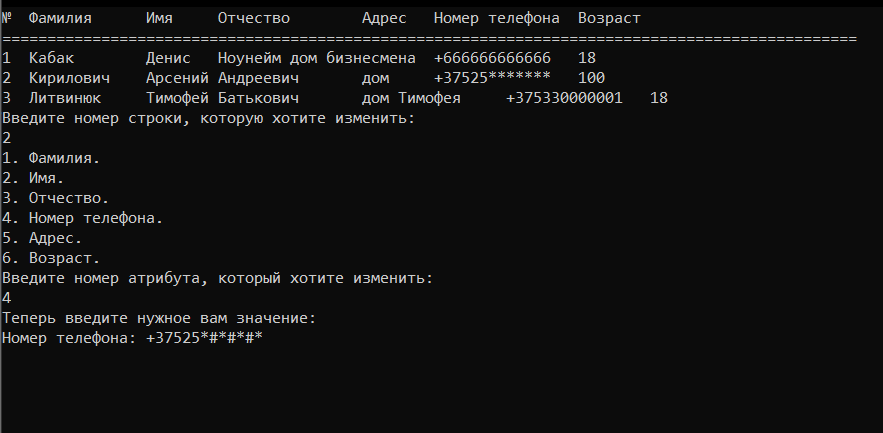


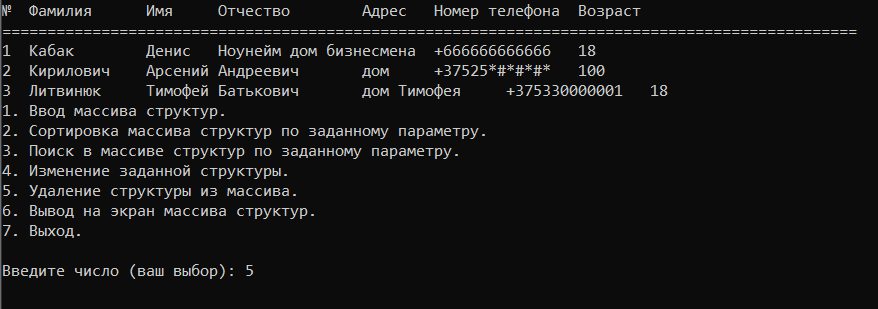


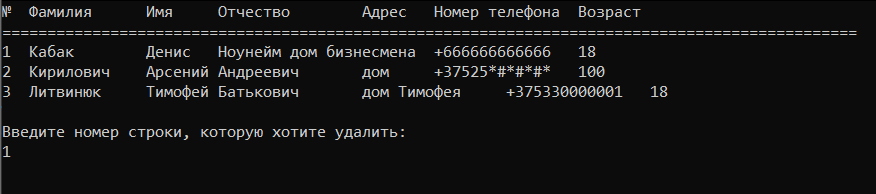


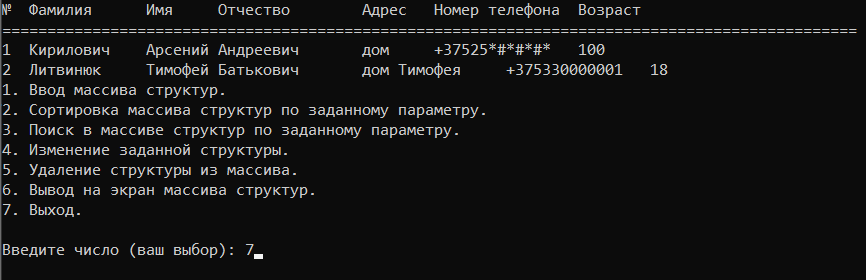


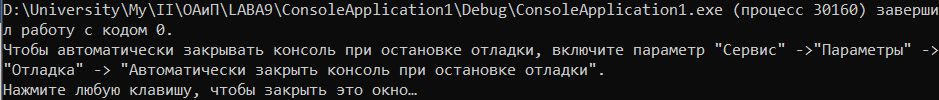












**Вывод:** научился работать со структурными переменными, изучил особенности различных видов структур, а также сферы их применения.