Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине ОАИП за 2 семестр

«динамические структуры данных»

Выполнил:

Студент группы ИИ-21

1-го курса

Корпач Д.Р.

Проверил :

Гирель Т.Н.

Брест 2022

Вариант 7

1. Функцию формирования динамического одномерного массива структур, значения которых вводятся с клавиатуры. Предусмотреть возможность заполнения одного поля структуры, используя известные значения других полей структуры. При вводе структур можно реализовать один из следующих механизмов: • ввод заранее заданного количества структур; • ввод до появления структуры с заданным признаком; • диалог с пользователем о необходимости продолжать ввод.

2. Функцию просмотра содержимого динамического массива структур.

3. Функцию дополнения уже существующего массива структур новыми структурами. Данная функция должна содержать параметр со значение по умолчанию (количество новых структур).

4. Функцию поиска и вывода на экран структуры (структур) с заданным значением элемента.

5. Функцию упорядочения массива структур по заданному полю (элементу).

По варианту

Рабочий»: фамилия; имя; отчество; домашний адрес ,национальность; дата рождения № цеха; табельный номер; образование; год поступления на работу.

#include<iostream>

#include <iomanip>

using namespace std;

struct Sponsor

{

    char name[32];

    char surname[32];

    char lastname[32];

    char home\_address[128];

    char narionality[32];

    char birthday[32];

    int numfactory;

    int tabnum;

    char education[128];

    int year\_work;

};

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

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

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

void redata(Sponsor\* Obj, const int amount);

void find(const Sponsor\* Obj, const int amount);

Sponsor\* sort(Sponsor\* Obj,  const int amount);

void sotf(Sponsor\* Obj,  const int amount);

int main()

{

    setlocale(LC\_ALL, "rus");

    Sponsor\* OurSponsors = 0;

    int sponsorAmount = 0;

    int YesOrNot = 0; //  продолжить или остановить ввод данных

    int num;

    for(int i = 1; i > 0;){

        for (int u = 0 ; u < 100; u++) {

            cout << endl;

        }

        cout << "1. input data\n";

        cout << "2. view data\n";

        cout << "3. changing data\n";

        cout << "4.sorted\n";

        cout << "5. search \n";

        cout << "6. Exit\n";

        cout << "input number: ";

        cin >> num;

        cin.get();

        for (int u = 0 ; u < 100; u++) {

            cout << endl;

        }

        if (num == 1) {

            do

            {

                OurSponsors = AddStruct(OurSponsors, sponsorAmount);

                setData(OurSponsors, sponsorAmount);

                sponsorAmount++;

                cout << "continue data entry (1 -yes, 0 - no): ";

                cin >> YesOrNot;

                cin.get();

            } while (YesOrNot != 0);

        }

        if (num == 2){

            showData(OurSponsors, sponsorAmount);

        }

        if (num == 3) {

            redata(OurSponsors, sponsorAmount);

        }

        if (num == 4){

            OurSponsors = sort(OurSponsors, sponsorAmount);

        }

        if (num == 5){

            find(OurSponsors, sponsorAmount);

        }

        if (num == 6) {

            break;

        }

    }

    return 0;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

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

{

    if (amount == 0)

    {

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

    }

    else

    {

        Sponsor\* tempObj = new Sponsor[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(Sponsor\* Obj, const int amount)

{

    cout << endl;

    cout << "surname: ";

    cin.getline(Obj[amount].surname, 32);

    cout << "name: ";

    cin.getline(Obj[amount].name, 32);

    cout << "patronymic: ";

    cin.getline(Obj[amount].lastname, 32);

    cout << "home adres: ";

    cin.getline(Obj[amount].home\_address, 128);

    cout << "nationality: ";

    cin.getline(Obj[amount].narionality, 32);

    cout << "date of berth: ";

    cin.getline(Obj[amount].birthday, 32);

    cout << "№ workshops: ";

    cin >> Obj[amount].numfactory;

    cout << "service number: ";

    cin >> Obj[amount].tabnum;

    cout << "edycation: ";

    cin.get();

    cin.getline(Obj[amount].education, 32);

    cout << "year of admission to work: ";

    cin >> Obj[amount].year\_work;

    cout << endl;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

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

{

    for (int i = 0; i < 100; i++) {

        cout << "\n";

    }

    cout << "# " << setw(9) << "surname  " << setw(7) << "  name  "<<setw(10)<< "  patronymic  "<<setw(9)<< "adres  "<<setw(10)<<"nationality  "<<setw(14)<< "   date of berth  "<<setw(13)<<"   # workshops: "<<setw(14)<<"  service number:  "<<setw(10)<<"  edycation: "<<setw(16)<<"  year go to work:"<< endl;

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

    for (int i = 0; i < amount; i++)

    {

        cout<<i+1<<setw(9)<<Obj[i].surname<<"  "<<setw(7)<<Obj[i].name<<"  "<<setw(10)<<Obj[i].lastname<<"  "<<setw(9)<< Obj[i].home\_address<<"  "<<setw(10)<< Obj[i].narionality<<"  "<<setw(14)<< Obj[i].birthday<<"  "<<setw(13)<< Obj[i].numfactory<<"  "<<setw(14)<<Obj[i].tabnum<<"  "<<setw(10)<<Obj[i].education<<"  "<<setw(16)<< Obj[i].year\_work<<endl;

    }

    int o;

    cout << "\n input 0 for exit: ";

    cin >> o;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void redata(Sponsor\* Obj, const int amount){

    for (int i = 0; i < 100; i++) {

        cout << "\n";

    }

    cout << "name\n";

    for (int i = 0; i < amount; i++)

    {

        cout << i + 1 << "  " << Obj[i].name << endl;

    }

    cout << "\nenter the number of name whose data you want to change: ";

    int o, i;

    cin >> o;

    o--;

    for (int j = 1; j > 0;) {

        for (int i = 0; i < 100; i++) {

            cout << "\n";

        }

        cout << "1. surname: " << Obj[o].surname << '\n' << "2.name: " << Obj[o].name << '\n' << "3.patronymic: "

             << Obj[o].lastname << '\n' << "4.adress: " << Obj[o].home\_address << '\n' << "5.nationlity: " << Obj[o].narionality

             << '\n' << "6. date berth: " << Obj[o].birthday<< '\n' << "7.№ Workshops : " << Obj[o].numfactory<< '\n' << "8. service number: " << Obj[o].tabnum<< '\n' << "9. edycation: " << Obj[o].education<< '\n' << "10.year of admission to work: " << Obj[o].year\_work << endl;

        cout << "input number: ";

        cin >> i;

        cout << "input new data: ";

        switch (i) {

            case 1:

                cin >> Obj[o].surname;

                break;

            case 2:

                cin >> Obj[o].name;

                break;

            case 3:

                cin >> Obj[o].lastname;

                break;

            case 4:

                cin >> Obj[o].home\_address;

                break;

            case 5:

                cin >> Obj[o].narionality;

                break;

            case 6:

                cin >> Obj[o].birthday;

                break;

            case 7:

                cin >> Obj[o].numfactory;

                break;

            case 8:

                cin >> Obj[o].tabnum;

                break;

            case 9:

                cin >> Obj[o].education;

                break;

            case 10:

                cin >> Obj[o].year\_work;

                break;

            default:

                cout << "wrong number";

                break;

        }

        cout << "something else to change?(1 - yes, 0 - no): ";

        cin >> i;

        cin.get();

        if (i == 1){

            continue;

        } else if(i == 0){

            break;

        }

    }

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void find(const Sponsor\* Obj, const int amount)

{

    char option;

    int a = 1;

    bool exit = false;

    cout << "instruction: b-base,n-name,p-patronymic,s-surname, o-edycation,h-home adres, t-№ workshops, l-service namber,j-edycation,y-year of admission to work,i-instruction,e-exit"<<endl;

    while (a > 0) {

        if (exit){

            break;

        }

        cout << "enter options: ";

        cin >> option;

        switch (option) {

            case 'b':

                cout << "# " << setw(9) << "surname  " << setw(7) << "  name  "<<setw(10)<< "  patronymic  "<<

                setw(9)<< "adres  "<<setw(10)<<"nationality  "<<setw(14)<< "   date of berth  "<<setw(13)<<

                "   # workshops: "<<setw(14)<<"  service number:  "<<setw(10)<<"  edycation: "<<setw(16)<<

                "  year go to work:"<< endl;

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

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

                {

                    cout<<i+1<<setw(9)<<Obj[i].surname<<"  "<<setw(7)<<Obj[i].name<<"  "<<setw(10)<<Obj[i].lastname<<"  "<<

                    setw(9)<< Obj[i].home\_address<<"  "<<setw(10)<< Obj[i].narionality<<"  "<<

                    setw(14)<< Obj[i].birthday<<"  "<<setw(13)<< Obj[i].numfactory<<"  "<<setw(14)<<Obj[i].tabnum

                    <<"  "<<setw(10)<<Obj[i].education<<"  "<<setw(16)<< Obj[i].year\_work<<endl;

                }

                cout << endl;

                break;

            case 'n':

                cout << "№  " << "name\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].name << endl;

                }

                cout << endl;

                break;

            case 's':

                cout << "№  " << "surname\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].surname << endl;

                }

                cout << endl;

                break;

            case 'p':

                cout << "№  " << "patronymic\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].lastname << endl;

                }

                cout << endl;

                break;

            case 'h':

                cout << "№  " << "home adres\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].home\_address << endl;

                }

                cout << endl;

                break;

            case 'g':

                cout << "№  " << "nationality\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].narionality << endl;

                }

                cout << endl;

                break;

            case 'r':

                cout << "№  " << "date berth\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].birthday << endl;

                }

                cout << endl;

                break;

            case 't':

                cout << "№  " << "№ workshops\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].numfactory << endl;

                }

                cout << endl;

                break;

            case 'l':

                cout << "№  " << "service number\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].tabnum << endl;

                }

                cout << endl;

                break;

            case 'j':

                cout << "№  " << "edycation\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].education << endl;

                }

                cout << endl;

                break;

            case 'y':

                cout << "№  " << "year of admission to work\t" << endl;

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

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

                {

                    cout << i + 1 << "  " << Obj[i].year\_work << endl;

                }

                cout << endl;

                break;

            case 'e':

                exit = true;

                break;

            case 'i':

                cout << "instruction: b-base,n-name,f-surname, o-edycation,d-home adres, g-groups, r-rating, t-№ workshops, l-service namber,j-edycation,u-year of admission to work,i-instruction,e-exit" << endl;

                cout << endl;

                break;

            default:

                cout << "Такой опции нет, если забыли, введите i\n";

        }

    }

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Sponsor\* sort(Sponsor\* Obj,  const int amount){

    int choise;

    cout << "1.surname\n";

    cout << "2.name\n";

    cout << "3.patronymic\n";

    cout << "4.home adres\n";

    cout << "5.nationality\n";

    cout << "6.date berth\n";

    cout << "7. № workshops\n";

    cout << "8.service number\n";

    cout << "9.edycation\n";

    cout << "10.year of admission to work\n";

    cout << "choise what you sorted : ";

    cin >> choise;

    int counter = 0;

    switch (choise) {

        case 1:

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

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

                        if (Obj[y].surname[0] < Obj[y + 1].surname[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].surname[0] > Obj[i + 1].surname[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 2:

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

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

                        if (Obj[y].name[0] < Obj[y + 1].name[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].name[0] > Obj[i + 1].name[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 3:

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

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

                        if (Obj[y].lastname[0] < Obj[y + 1].lastname[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].lastname[0] > Obj[i + 1].lastname[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 4:

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

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

                        if (Obj[y].home\_address[0] < Obj[y + 1].home\_address[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].home\_address[0] > Obj[i + 1].home\_address[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 5:

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

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

                        if (Obj[y].narionality[0] < Obj[y + 1].narionality[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].narionality[0] > Obj[i + 1].narionality[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 6:

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

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

                        if (Obj[y].birthday[0] < Obj[y + 1].birthday[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].birthday[0] > Obj[i + 1].birthday[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 7:

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

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

                        if (Obj[y].numfactory < Obj[y + 1].numfactory){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].numfactory > Obj[i + 1].numfactory){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 8:

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

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

                        if (Obj[y].tabnum < Obj[y + 1].tabnum){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].tabnum > Obj[i + 1].tabnum){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 9:

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

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

                        if (Obj[y].education[0] < Obj[y + 1].education[0]){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].education[0] > Obj[i + 1].education[0]){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

        case 10:

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

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

                        if (Obj[y].year\_work < Obj[y + 1].year\_work){

                            counter++;

                        }

                }

                if (counter == amount - 1) {

                    break;

                }

                counter = 0;

                if (Obj[i].year\_work > Obj[i + 1].year\_work){

                    if (i != amount - 1){

                        Sponsor\* tempObj = new Sponsor[amount];

                        tempObj[i] = Obj[i];

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

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

                        delete [] tempObj;

                    } else{

                        i = 0;

                        continue;

                    }

                }

            }

            break;

    }

    return Obj;

}















