**КОЛО**

**#include "stdafx.h"**

**//MFC основные и стандартные компоненты**

**#include <afxwin.h>**

**//MFC расширения**

**#include <afxext.h>**

**//Идентификаторы ресурсов**

**#include "resource.h"**

**class CMyApp : public CWinApp**

**{**

**public:**

**//конструктор по умолчанию**

**CMyApp();**

**//стандартная инициализация**

**virtual BOOL InitInstance();**

**};**

**class CMainWnd : public CFrameWnd**

**{**

**public:**

**//Конструктор по умолчанию**

**CMainWnd();**

**private:**

**//Квадрат, в который вписан отображаемый круг**

**CRect newPlace;**

**//Квадрат, в который был вписан отображаемый**

**круг в предыдущий момент**

**CRect oldPlace;**

**CRect oldPlace2;**

**CRect oldPlace3;**

**CRect newPlace2;**

**CRect newPlace3;**

**//Таблица откликов окна**

**DECLARE\_MESSAGE\_MAP();**

**public:**

**afx\_msg void OnPaint();**

**};**

**//Таблица откликов на сообщения**

**BEGIN\_MESSAGE\_MAP(CMainWnd, CFrameWnd)**

**//Событие создания окна**

**ON\_WM\_CREATE()**

**ON\_WM\_PAINT()**

**END\_MESSAGE\_MAP()**

**// конструктор главного класса приложения**

**CMyApp::CMyApp(){}**

**// стандартная инициализация**

**CMyApp::InitInstance()**

**{**

**m\_pMainWnd=new CMainWnd();**

**ASSERT(m\_pMainWnd);**

**m\_pMainWnd->ShowWindow(SW\_SHOW);**

**m\_pMainWnd->UpdateWindow();**

**return TRUE;**

**};**

**CMainWnd::CMainWnd()**

**{**

**oldPlace.left = 150;**

**oldPlace.top = 150;**

**oldPlace.right = 190;**

**oldPlace.bottom = 190;**

**newPlace = oldPlace;**

**oldPlace2.left = 190;**

**oldPlace2.top = 190;**

**oldPlace2.right = 250;**

**oldPlace2.bottom = 250;**

**newPlace2 = oldPlace2;**

**oldPlace3.left = 600;**

**oldPlace3.top = 600;**

**oldPlace3.right = 580;**

**oldPlace3.bottom = 580;**

**newPlace3 = oldPlace3;**

**// создать окно программы**

**Create(NULL,"Окно приложения пользователя",**

**WS\_OVERLAPPEDWINDOW, rectDefault, NULL, NULL);**

**}**

**void CMainWnd::OnPaint()**

**{**

**CPaintDC dc(this);**

**CBrush bBrush(RGB(255,144,0));**

**dc.SelectObject(&bBrush);**

**dc.Ellipse(newPlace);**

**CBrush bbBrush(RGB(0,144,0));**

**dc.SelectObject(&bbBrush);**

**dc.Ellipse(newPlace2);**

**CBrush bb1Brush(RGB(100,0,100));**

**dc.SelectObject(&bb1Brush);**

**dc.Ellipse(newPlace3);**

**}**

**//Запуск приложения**

**CMyApp theApp;**

**СОМПЛЕКСНІ ЧИСЛА**

**#include "stdafx.h"**

**#include <iostream>**

**#include <sstream>**

**using namespace std;**

**class Comp**

**{**

**public:**

**double Re,Im;**

**};**

**Comp operator +(Comp x,Comp y)**

**{**

**Comp z;**

**z.Re=x.Re+y.Re;**

**z.Im=x.Im+y.Im;**

**return z;**

**}**

**Comp operator -(Comp x,Comp y)**

**{**

**Comp z;**

**z.Re=x.Re-y.Re;**

**z.Im=x.Im-y.Im;**

**return z;**

**}**

**int main()**

**{**

**Comp a,b,c;**

**int vub=-1;**

**cout<<"a.Re=";**

**cin>>a.Re;**

**cout<<"a.Im=";**

**cin>>a.Im;**

**cout<<"b.Re=";**

**cin>>b.Re;**

**cout<<"b.Im=";**

**cin>>b.Im;**

**while (vub!=0)**

**{**

**cout<<"1. Dodatu"<<endl;**

**cout<<"2. Vidnatu"<<endl;**

**cout<<"0. Vuxid"<<endl;**

**cin>>vub;**

**switch(vub)**

**{**

**case 1:**

**c=a+b;**

**case 2:**

**c=a-b;**

**}**

**cout<<"c.Re="<<c.Re<<endl;**

**cout<<"c.Im="<<c.Im<<endl;**

**}**

**system("pause");**

**return 0;**

**}**

**STL СТЕК**

**#include "stdafx.h"**

**#include <stack>**

**#include <vector>**

**#include <iostream>**

**int main()**

**{**

**std::stack<int> s;**

**int i;**

**for(i = 0; i < 10; ++i)**

**s.push(i);**

**std::stack<int> s1 (s);**

**while(!s1.empty()) {**

**std::cout << s1.top() << ' ';**

**s1.pop();**

**}**

**std::cout << "\n";**

**s.push(1);**

**s.pop();**

**s.pop();**

**std::stack<int> s2 (s);**

**while(!s2.empty()) {**

**std::cout << s2.top() << ' ';**

**s2.pop();**

**}**

**system("PAUSE");**

**return 0;**

**}**

**STL МАСИВ**

**#include "stdafx.h"**

**#include <iostream>**

**#include <array>**

**int main ()**

**{**

**std::array<int,5> myarray = { 2, 16, 77, 34, 50 };**

**int n=5;**

**std::cout << "myarray contains:";**

**for ( auto it = myarray.begin(); it != myarray.end(); ++it )**

**std::cout << ' ' << \*it;**

**std::cout << '\n';**

**int d=1;**

**for(int i=1;i<n;i=i+2)**

**{ d=d\*myarray[i]; }**

**std::cout << d << "\n";**

**//запис у файл**

**FileStream^ binaryStream = gcnew FileStream("binfile.bin", FileMode::Create);**

**UTF8Encoding^ utf8 = gcnew UTF8Encoding();**

**BinaryWriter^ binaryFile = gcnew BinaryWriter(binaryStream, utf8, true);**

**for(int i=0;i<n;i++) binaryFile -> Write(myarray[i]);**

**binaryStream -> Close();**

**//вивід з файлу**

**FileStream^ binaryStream = gcnew FileStream("binfile.bin", FileMode::Open);**

**UTF8Encoding^ utf8 = gcnew UTF8Encoding();**

**BinaryReader^ binaryFile = gcnew BinaryReader(binaryStream, utf8);**

**binaryStream -> Seek(0, SeekOrigin::Begin);**

**std::cout << (binaryFile -> ReadInt32());**

**binaryStream -> Close();**

**system("PAUSE");**

**return 0;**

**}**

**STL СПИСОК**

**#include "stdafx.h"**

**#include <iostream>**

**#include <list>**

**#include <time.h>**

**int main ()**

**{**

**std::list<int> mylist;**

**std::list<int>::iterator it;**

**mylist.push\_back (rand() % 100 + 1);**

**mylist.push\_back (rand() % 100 + 1);**

**mylist.push\_back (rand() % 100 + 1);**

**std::cout << "mylist contains:";**

**for (it=mylist.begin(); it!=mylist.end(); ++it)**

**std::cout << ' ' << \*it;**

**std::cout << '\n';**

**mylist.sort();**

**std::cout << mylist.front() << "\n";**

**std::cout << mylist.back() << "\n";**

**system("PAUSE");**

**return 0;**

**}**

**ТЕЛ ДОВІДНИК**

**r e c o r d . h**

**#include <string>**

**#include <cstring>**

**using namespace std;**

**class record {**

**public:**

**string adress;**

**string phone\_number;**

**virtual void Input() = 0;**

**virtual void Output() = 0;**

**};**

**p e r s o n . h**

**#include "record.h"**

**class person : public record {**

**public:**

**string surname;**

**void Input();**

**void Output();**

**void Search(string s);**

**}**

**p e r s o n . c p p**

**#include "person.h"**

**void person::Input() {**

**cout << "Input surname - ";**

**cin >> surname;**

**cout << "Input adress - ";**

**cin >> adress;**

**cout << "Input telephone number - ";**

**cin >> phone\_number;**

**}**

**void person::Output() {**

**cout << "Surname: " << surname << endl;**

**cout << "Adress: " << adress << endl;**

**cout << "Telephone number: " << phone\_number << endl;**

**}**

**void person::Search(string s) {**

**int p = 0;**

**if (s == surname)p++;**

**if (s == adress)p++;**

**if (s == phone\_number)p++;**

**if (p > 0) Output();**

**};**

**o r g a n i z a t i on n . h**

**#pragma once**

**#include "record.h"**

**class organization : public record {**

**public:**

**string name;**

**string fax;**

**string contact\_person;**

**void Input();**

**void Output();**

**void Search(string s);**

**};**

**o r g a n i z a t i on n . c p p**

**#include "organization.h"**

**void organization::Input() {**

**cout << "Input name (organization) - ";**

**cin >> name;**

**cout << "Input contact person - ";**

**cin >> contact\_person;**

**cout << "Input adress - ";**

**cin >> adress;**

**cout << "Input telephone number - ";**

**cin >> phone\_number;**

**cout << "Input fax - ";**

**cin >> fax;**

**};**

**void organization::Output() {**

**cout << "Name (organization): " << name;**

**cout << "Contact person: " << contact\_person << endl;**

**cout << "Adress: " << adress << endl;**

**cout << "Telephone number: " << phone\_number << endl;**

**cout << "Fax: " << fax << endl;**

**};**

**void organization::Search(string s) {**

**int p = 0;**

**if (s == name)p++;**

**if (s == contact\_person)p++;**

**if (s == adress)p++;**

**if (s == phone\_number)p++;**

**if (s == fax)p++;**

**if (p > 0) Output();**

**};**

**f r i e n d s . h**

**#include "person.h"**

**class friends : public person {**

**public:**

**string date;**

**void Input();**

**void Output();**

**void Search(string s);**

**};**

**f r i e n d s . c p p**

**#include "friends.h"**

**void friends::Input() {**

**Input();**

**cout << "Input data of birth - ";**

**}**

**void friends::Output() {**

**Output();**

**cout << "Data of birth: " << date << endl;;**

**}**

**void friends::Search(string s) {**

**int p = 0;**

**if (s == surname)p++;**

**if (s == adress)p++;**

**if (s == phone\_number)p++;**

**if (s == date)p++;**

**if (p > 0) Output();**

**}**

**t e l \_ b o o k . h**

**#include "person.h"**

**#include "organization.h"**

**#include "friends.h"**

**class tel\_book {**

**private:**

**int k, kq, kw, ke;**

**person per[15];**

**organization org[15];**

**friends fri[15];**

**public:**

**void Input();**

**void Output();**

**void Find();**

**};**

**t e l \_ b o o k . c p p**

**#include "tel\_book.h"**

**void tel\_book::Input() {**

**int j;**

**kq = 0; kw = 0; ke = 0;**

**cout << "Input N - ";**

**cin >> k; cout << endl;**

**for (int i = 0; i < k; i++) {**

**cout << "\nChoose the type:" << endl;**

**cout << "(1-person,2-organization,3-friend)" << endl;**

**cin >> j;**

**if (j == 1) {**

**per[kq].Input();**

**kq++;**

**} else**

**if (j == 2) {**

**org[kw].Input();**

**kw++;**

**} else {**

**fri[ke].Input();**

**ke++;**

**} }**

**};**

**void tel\_book::Output() {**

**if (kq != 0) {**

**cout << "Persons" << endl;**

**for (int i = 0; i < kq; i++) {**

**per[i].Output();**

**cout << endl;**

**}**

**}**

**if (kw != 0) {**

**cout << "Organizations" << endl;**

**for (int i = 0; i < kw; i++) {**

**org[i].Output();**

**cout << endl;**

**}**

**}**

**if (ke != 0) {**

**cout << "Friends" << endl;**

**for (int i = 0; i < ke; i++) {**

**fri[i].Output();**

**cout << endl;**

**} }**

**};**

**void tel\_book::Find()**

**{**

**bool tel;**

**string s;**

**cout << "Enter the data to search - ";**

**cin >> s;**

**cout << "\nResult of search" << endl;**

**for (int i = 0; i < kq; i++) {**

**per[i].Search(s);**

**cout << endl;**

**}**

**for (int i = 0; i < kw; i++) {**

**org[i].Search(s);**

**cout << endl;**

**}**

**for (int i = 0; i < ke; i++) {**

**fri[i].Search(s);**

**cout << endl;**

**}**

**};**

**M a i n . h**

**#include "person.h"**

**#include "organization.h"**

**#include "friends.h"**

**#include "tel\_book.h"**

**#include <iostream>**

**#include <string>**

**#include <cstring>**

**using namespace std;**

**int main() {**

**tel\_book a;**

**a.Input();**

**cout << "---------" << endl;**

**a.Find();**

**system("pause");**

**return 0;**

**}**