**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНЖИНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ**

**Институт компьютерных технологий и информационной безопасности**

**Кафедра Информационно-аналитических систем безопасности**

**Отчёт по лабораторной работе №2**по курсу «Объектно-ориентированное программирование»

Выполнил:  
Студент гр. КТбо2-9  
t.me/youmustdefend

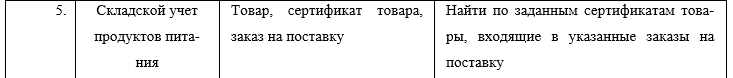
Проверил:  
Профессор, д. т. н. ИКТИБ  
Беляков С. Л.

**Вариант №5**

**Цель работы:**

Разработать приложение с графическим интерфейсом для Windows, используя библиотеку .NET.

**Вариант индивидуального задания:**

****

**Программный код после отладки**

MyForm.h:

#pragma once

#include "Certificate.h"

#include "ClassFind.h"

#include "Order.h"

#include "Product.h"

namespace LabV2 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

private:

ArrayList^ Products = gcnew ArrayList;

ArrayList^ Orders = gcnew ArrayList;

ArrayList^ Certificates = gcnew ArrayList;

ClassFind^ test = gcnew ClassFind();

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

Product^ prod1 = gcnew Product();

prod1->setID(1);

prod1->setName("Cheese");

Product^ prod2 = gcnew Product();

prod2->setID(2);

prod2->setName("Meat");

Product^ prod3 = gcnew Product();

prod3->setID(3);

prod3->setName("Fish");

Products->Add(prod1);

Products->Add(prod2);

Products->Add(prod3);

Certificate^ cert1 = gcnew Certificate;

cert1->setCertNum(1);

cert1->setID(prod1->getID());

prod1->setCertNum(cert1->getCertNum());

Certificate^ cert2 = gcnew Certificate;

cert2->setCertNum(2);

cert2->setID(cert2->getID());

prod2->setCertNum(cert2->getCertNum());

Certificate^ cert3 = gcnew Certificate;

cert3->setCertNum(3);

cert3->setID(cert1->getID());

prod3->setCertNum(cert3->getCertNum());

Certificates->Add(cert1);

Certificates->Add(cert2);

Certificates->Add(cert3);

Order^ order1 = gcnew Order();

order1->setAddress("per. Dobrolyubovskiy 15");

order1->setID(prod1->getID());

order1->setQuantity(4);

order1->setOrderNum(1);

prod1->addOrder(order1);

Order^ order2 = gcnew Order();

order2->setAddress("st. Aleksandrovskaya 31");

order2->setID(prod2->getID());

order2->setQuantity(5);

order2->setOrderNum(2);

prod2->addOrder(order2);

Order^ order3 = gcnew Order();

order3->setAddress("st. Petrovskaya 1");

order3->setID(prod3->getID());

order3->setQuantity(6);

order3->setOrderNum(3);

prod3->addOrder(order3);

Orders->Add(order1);

Orders->Add(order2);

Orders->Add(order3);

test->setClassFind(Products, Certificates, Orders);

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::DataGridView^ dataGridView1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column3;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column4;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->button1 = (gcnew System::Windows::Forms::Button());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->Column1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column4 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

this->SuspendLayout();

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(12, 29);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(199, 22);

this->textBox1->TabIndex = 0;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(12, 107);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(199, 22);

this->textBox2->TabIndex = 1;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 9);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(199, 17);

this->label1->TabIndex = 2;

this->label1->Text = L"Введите номер сертификата";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 87);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(157, 17);

this->label2->TabIndex = 3;

this->label2->Text = L"Введите номер заказа";

//

// button1

//

this->button1->Location = System::Drawing::Point(12, 193);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(97, 27);

this->button1->TabIndex = 4;

this->button1->Text = L"Найти";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// dataGridView1

//

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(4) {

this->Column1,

this->Column2, this->Column3, this->Column4

});

this->dataGridView1->Location = System::Drawing::Point(273, 12);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->RowHeadersWidth = 51;

this->dataGridView1->RowTemplate->Height = 24;

this->dataGridView1->Size = System::Drawing::Size(714, 391);

this->dataGridView1->TabIndex = 6;

//

// Column1

//

this->Column1->HeaderText = L"Номер сертификата";

this->Column1->MinimumWidth = 6;

this->Column1->Name = L"Column1";

this->Column1->Width = 125;

//

// Column2

//

this->Column2->HeaderText = L"Номер заказа";

this->Column2->MinimumWidth = 6;

this->Column2->Name = L"Column2";

this->Column2->Width = 125;

//

// Column3

//

this->Column3->HeaderText = L"Товар";

this->Column3->MinimumWidth = 6;

this->Column3->Name = L"Column3";

this->Column3->Width = 125;

//

// Column4

//

this->Column4->HeaderText = L"Адрес поставки";

this->Column4->MinimumWidth = 6;

this->Column4->Name = L"Column4";

this->Column4->Width = 125;

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(999, 415);

this->Controls->Add(this->dataGridView1);

this->Controls->Add(this->button1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Name = L"MyForm";

this->Text = L"MyForm";

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (this->textBox1->Text != "" && this->textBox2->Text != "")

{

ArrayList^ testCertificates = gcnew ArrayList;

ArrayList^ testOrders = gcnew ArrayList;

int cert = Convert::ToInt32(this->textBox1->Text);

int ord = Convert::ToInt32(this->textBox2->Text);

Certificate^ testCert1 = gcnew Certificate;

testCert1->setCertNum(cert);

testCertificates->Add(testCert1);

Order^ testOrd1 = gcnew Order;

testOrd1->setOrderNum(ord);

testOrders->Add(testOrd1);

ArrayList^ getProducts = test->findProducts(testCertificates, testOrders);

for (int i = 0; i < getProducts->Count; i++)

{

int productID = static\_cast<Product^>(getProducts[i])->getID();

String^ productName = static\_cast<Product^>(getProducts[i])->getName();

this->dataGridView1->Rows->Add(cert, ord, productName);

}

}

}

};

}

Order.h:

#pragma once

using namespace System;

using namespace System::Collections;

ref class Order

{

private:

String^ Address = "";

int productID = -1;

int quantity = -1;

int orderNum = -1;

public:

void setAddress(String^ addr);

String^ getAddress();

void setID(int pID);

int getID();

void setQuantity(int q);

int getQuantity();

int getOrderNum();

void setOrderNum(int oN);

};

Order.cpp:

#include "Order.h"

void Order::setID(int pID)

{

if (pID > 0)

{

this->productID = pID;

}

else

{

return;

}

}

int Order::getID()

{

return this->productID;

}

void Order::setAddress(String^ addr)

{

if (addr != "")

{

this->Address = addr;

}

else

{

return;

}

}

String^ Order::getAddress()

{

return this->Address;

}

void Order::setQuantity(int q)

{

if (q > 0)

{

this->quantity = q;

}

else

{

return;

}

}

int Order::getQuantity()

{

return this->quantity;

}

int Order::getOrderNum()

{

return this->orderNum;

}

void Order::setOrderNum(int oN)

{

if (oN > 0)

{

this->orderNum = oN;

}

else

{

return;

}

}

Product.h:

#pragma once

#include "Order.h"

using namespace System;

using namespace System::Collections;

ref class Product

{

private:

String^ name = "";

int productID = -1;

int certNumber = -1;

ArrayList^ orders = gcnew ArrayList();

public:

void setName(String^ n);

String^ getName();

void setID(int pID);

int getID();

void setCertNum(int cN);

int getCertNum();

void addOrder(Order^ o);

ArrayList^ getOrders();

};

Product.cpp:

#include "Product.h"

using namespace System::Collections;

void Product::setName(String^ n)

{

if (n != "")

{

this->name = n;

}

else

{

return;

}

}

String^ Product::getName()

{

return this->name;

}

void Product::setID(int pID)

{

if (pID > 0)

{

this->productID = pID;

}

else

{

return;

}

}

int Product::getID()

{

return this->productID;

}

void Product::setCertNum(int cN)

{

if (cN > 0)

{

this->certNumber = cN;

}

else

{

return;

}

}

int Product::getCertNum()

{

return this->certNumber;

}

void Product::addOrder(Order^ o)

{

this->orders->Add(o);

}

ArrayList^ Product::getOrders()

{

return this->orders;

}

Certificate.h:

#pragma once

using namespace System;

using namespace System::Collections;

ref class Certificate

{

private:

int certNumber = -1;

int productID = -1;

public:

void setCertNum(int cN);

int getCertNum();

void setID(int pID);

int getID();

};

Certificate.cpp

#include "Certificate.h"

void Certificate::setCertNum(int cN)

{

if (cN > 0)

{

this->certNumber = cN;

}

else

{

return;

}

}

int Certificate::getCertNum()

{

return this->certNumber;

}

void Certificate::setID(int pID)

{

if (pID > 0)

{

this->productID = pID;

}

else

{

return;

}

}

int Certificate::getID()

{

return this->productID;

}

ClassFind.h:

#pragma once

#include "Certificate.h"

#include "Product.h"

#include "Order.h"

using namespace System;

using namespace System::Collections;

ref class ClassFind

{

private:

ArrayList^ Products = gcnew ArrayList;

ArrayList^ Certificates = gcnew ArrayList;

ArrayList^ Orders = gcnew ArrayList;

public:

ClassFind();

ArrayList^ findProducts(ArrayList^ certificatesArray, ArrayList^ ordersArray);

void setClassFind(ArrayList^ productsArray, ArrayList^ certificatesArray, ArrayList^ ordersArray);

};

ClassFind.cpp:

#include "ClassFind.h"

void ClassFind::setClassFind(ArrayList^ productsArray, ArrayList^ certificatesArray, ArrayList^ ordersArray)

{

if (productsArray->Count > 0 && certificatesArray->Count > 0 && ordersArray->Count > 0)

{

this->Certificates = certificatesArray;

this->Products = productsArray;

this->Orders= ordersArray;

}

else

{

return;

}

}

ArrayList^ ClassFind::findProducts(ArrayList^ certificatesArray, ArrayList^ ordersArray)

{

ArrayList^ productsArray = gcnew ArrayList();

ArrayList^ claimOrders = gcnew ArrayList();

for (int i = 0; i < ordersArray->Count; i++)

{

int Num = static\_cast<Order^>(ordersArray[i])->getOrderNum();

for (int j = 0; j < this->Orders->Count; j++)

{

int thisNum= static\_cast<Order^>(this->Orders[j])->getOrderNum();

if (Num == thisNum)

{

claimOrders->Add(this->Orders[j]);

}

}

}

ArrayList^ suitOrders = gcnew ArrayList();

for (int i = 0; i < claimOrders->Count; i++)

{

int oPID = static\_cast<Order^>(claimOrders[i])->getID();

for (int j = 0; j < this->Products->Count; j++)

{

int PID = static\_cast<Product^>(this->Products[j])->getID();

int pCertNum = static\_cast<Product^>(this->Products[j])->getCertNum();

for (int k = 0; k < certificatesArray->Count; k++)

{

int CertNum = static\_cast<Certificate^>(certificatesArray[k])->getCertNum();

if (oPID == PID && CertNum == pCertNum)

{

suitOrders->Add(static\_cast<Order^>(claimOrders[i]));

}

}

}

}

for (int i = 0; i < suitOrders->Count; i++)

{

int sOrdNum = static\_cast<Order^>(suitOrders[i])->getOrderNum();

for (int j = 0; j < this->Products->Count; j++)

{

ArrayList^ pOrdres = (static\_cast<Product^>(this->Products[j])->getOrders());

for (int k = 0; k < pOrdres->Count; k++)

{

int pOrdNum = static\_cast<Order^>(pOrdres[k])->getOrderNum();

if (pOrdNum == sOrdNum)

{

productsArray->Add(this->Products[j]);

}

}

}

}

return productsArray;

}

ClassFind::ClassFind()

{

}

**Тест**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Номера сертификата и заказа: 2, 2. | Продукты, номера их заказов и сертификатов: 2, 2, Meat. |

**Вывод**

В процессе выполнения лабораторной работы я разработал приложение с графическим интерфейсом для Windows, используя библиотеку .NET. Расширил и укрепил свои знания в среде программирования С++.