

Министерство образования и науки, молодежи и спорта Украины
Донбасская государственная машиностроительная академия
Кафедра Прикладной математики



С. Л. Загребельный, А. А. Костиков, В. Э. Миринский

ОСНОВЫ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ В СРЕДЕ VISUAL STUDIO 2010

**Методические указания и контрольные задания
по дисциплине
«Вычислительная техника и программирование»**

*для студентов специальности
8.05070204 «Электромеханические системы
автоматизации и электропривод»
заочной формы обучения*

Краматорск 2012

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ

Контрольная работа является средством проверки правильности усвоения студентами основных положений изучаемой дисциплины при самостоятельной подготовке.

На титульном листе контрольной работы необходимо указать: наименование кафедры, название дисциплины, курс, номер контрольной работы, свой шифр, шифр группы, фамилию, имя и отчество (полностью).

Контрольная работа выполняется на компьютере и распечатывается на принтере на листах формата А4, жестко скрепленных между собой.

При повторной сдаче контрольной работы на проверку на исправленных листах должно быть написано «Исправления». Первоначальный вариант контрольной работы и рецензии на нее должны прилагаться.

Вместе с контрольной работой каждый студент сдает диск (CD или DVD), на котором помещаются все результаты работы. Результат каждой лабораторной работы сохранять в отдельной папке (**Lab_1, lab_2, lab_3** и т.д.). Название проектов набирать английскими буквами.

Контрольная работа состоит из 6 лабораторных работ, т.е. всего 6 программ. **Номера заданий** контрольной работы выбирается в соответствии с двумя последними цифрами зачетной книжки из таблицы 1.

Например, последние 2 цифры зачетки 00 смотрим по таблице 1 (б/в, 25, 7, 10, 20, 6), тогда лабораторная работа 1 без варианта, лабораторная работа 2 – 25 вариант, лабораторная работа 3 – 7 вариант и т.д.

Таблица 1 – Выбор заданий в соответствии с номером зачетной книжки

Пред- по- след- няя цифра	Последняя цифра									
	0	1	2	3	4	5	6	7	8	9
0	б/в, 25,7, 10,20,6	б/в, 24,8, 11,19,5	б/в, 23,9, 12,18,4	б/в,22,10, 13,17,3	б/в,21,11, 14,16,2	б/в,20,12, 15,15,1	б/в,19,13, 16,14,25	б/в,18,14, 17,13,24	б/в,17,15, 18,12,23	б/в,16,16, 19,11,22
1	б/в,15,17, 20,10,21	б/в,14,18, 21,9,20	б/в,13,19, 22,8,19	б/в,12,20, 23,7,18	б/в,11,21, 24,6,17	б/в,10,22, 25,5,16	б/в,9,23, 1,4,15	б/в,8,24, 2,3,14	б/в,7,25, 3,2,13	б/в,6,1, 4,1,12
2	б/в,5,2, 5,25,11	б/в,4,3, 6,24,10	б/в,3,4, 7,23,9	б/в,2,5, 8,22,8	б/в,1,6, 9,21,7	б/в,15,8, 11,20,6	б/в,14,9, 12,19,5	б/в,13,10, 13,18,4	б/в,12,11, 14,17,3	б/в,11,12, 15,16,2
3	б/в,10,13, 16,15,1	б/в,9,14, 17,14,25	б/в,8,15, 18,13,24	б/в,7,16, 19,12,23	б/в,6,17, 20,11,22	б/в,5,18, 21,10,21	б/в,4,19, 22,9,20	б/в,3,20, 23,8,19	б/в,2,21, 24,7,18	б/в,1,22, 25,6,17
4	б/в,25,23, 1,5,16	б/в,24,24, 2,4,15	б/в,23,25, 3,3,14	б/в,22,1, 4,2,13	б/в,21,2, 5,1,12	б/в,20,3, 6,25,11	б/в,19,4, 7,24,10	б/в,18,5, 8,23,9	б/в,17,6, 9,22,8	б/в,16,7, 10,21,7
5	б/в,15,6, 12,13,2	б/в,14,7, 13,12,1	б/в,13,8, 14,11,25	б/в,12,9, 15,10,24	б/в,11,10, 16,9,23	б/в,10,11, 17,8,22	б/в,9,12, 18,7,21	б/в,8,13, 19,6,20	б/в,7,14, 20,5,19	б/в,6,15, 21,4,18
6	б/в,5,16 22,3,17	б/в,4,17, 23,2,16	б/в,3,18, 24,1,15	б/в,2,19, 25,24,14	б/в,1,20, 1,25,13	б/в,15,21, 2,23,12	б/в,14,22, 3,22,11	б/в,13,23, 4,21,10	б/в,12,24, 5,20,9	б/в,11,25, 6,19,8
7	б/в,10,1, 7,18,7	б/в,9,2, 8,17,6	б/в,8,3, 9,16,5	б/в,7,4, 10,15,4	б/в,6,5, 11,14,3	б/в,5,4, 9,13,25	б/в,4,5, 10,12,24	б/в,3,6, 11,10,23	б/в,2,7, 12,11,22	б/в,1,8, 13,9,21
8	б/в,25,9, 14,8,20	б/в,24,10, 15,7,19	б/в,23,11, 16,6,18	б/в,22,12, 17,5,17	б/в,21,13, 18,4,16	б/в,20,14, 19,3,15	б/в,19,15, 20,2,14	б/в,18,16, 21,1,13	б/в,17,17, 22,25,12	б/в,16,18, 23,24,11
9	б/в,25,19, 24,23,10	б/в,24,20, 25,22,9	б/в,23,21, 1,21,8	б/в,22,22, 2,20,7	б/в,21,23, 3,19,6	б/в,20,24, 4,18,5	б/в,19,25, 5,17,4	б/в,18,1, 6,16,3	б/в,17,2, 7,15,2	б/в,16,3, 8,14,1

ЛАБОРАТОРНАЯ РАБОТА 1

Создание простейшей программы на языке Visual C++

Цель: научиться использовать среду Visual Studio 2010 для создания простейших windows приложений по программированию на языке C++.

1.1 Задание к лабораторной работе

- 1 Проверить наличие личной папки. При необходимости создать ее.
- 2 В личной папке создать вложенную папку для лабораторной работойой № 1, присвоив этой папке имя **Lab1**
- 3 Войти в среду *Visual Studio 2010*.
- 4 После запуска **Microsoft Visual Studio 2010** появляется следующая стартовая страница, которая показана на рис. 1.1.

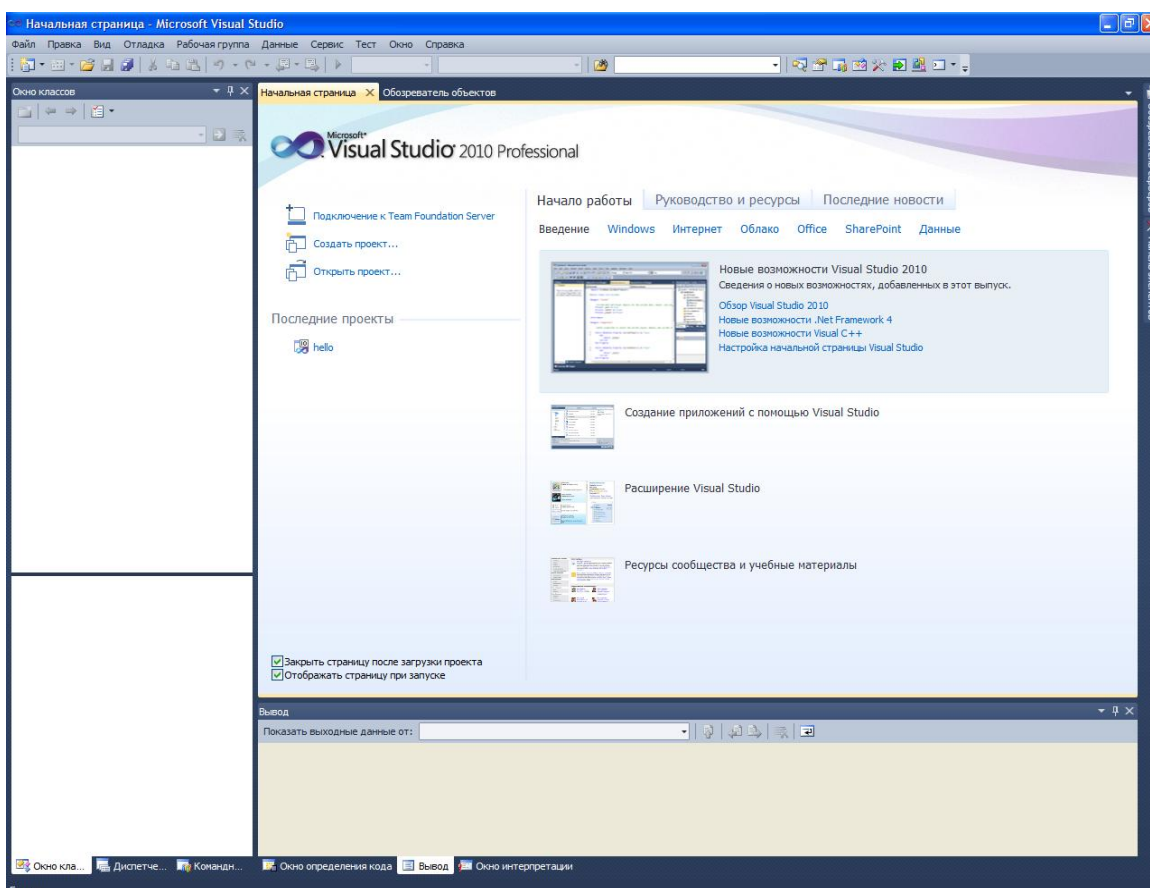


Рисунок 1.1 – Стартовая страница Visual Studio 2010

Следующим шагом является создание нового проекта. Для этого в меню **Файл** необходимо выбрать **Создать** → **Проект** (или комбинацию клавиш **Ctrl + Shift + N**). Результат выбора пунктов меню для создания нового проекта показан на рис. 1.2.

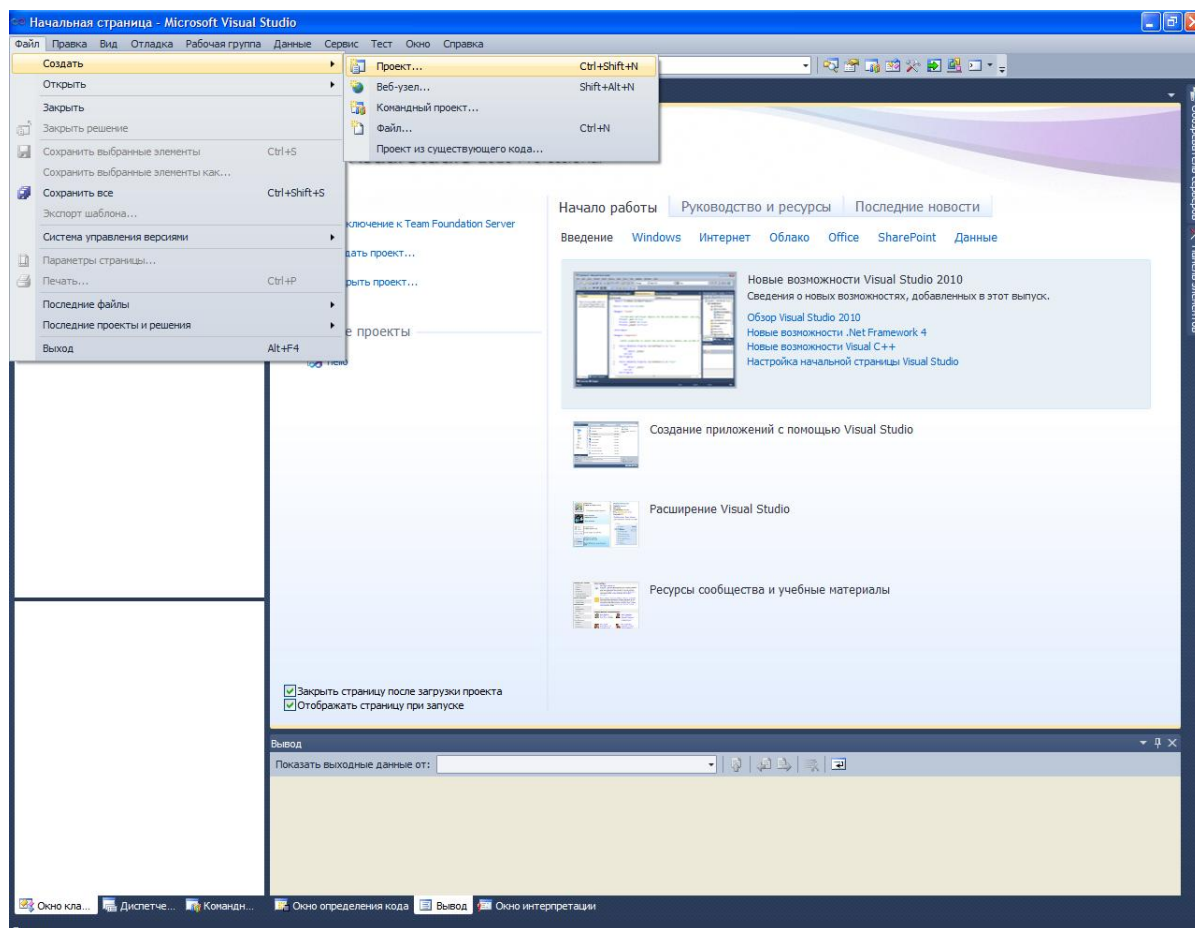


Рисунок 1.2 – Окно с выбором нового проекта

Среда Visual Studio отобразит окно **Создать Проект**, в котором необходимо выбрать тип создаваемого проекта. Проект используется в Visual Studio для логической группировки нескольких файлов, содержащих исходный код, на одном из поддерживаемых языков программирования, а также любых вспомогательных файлов. Обычно после сборки проекта (которая включает компиляцию всех входящих в проект файлов исходного кода) создается один исполняемый модуль.

В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.

Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, например, **Primer**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор**. По умолчанию проект сохраняется в специальной папке Projects. Пример выбора имени проекта показано на рис. 1.3.

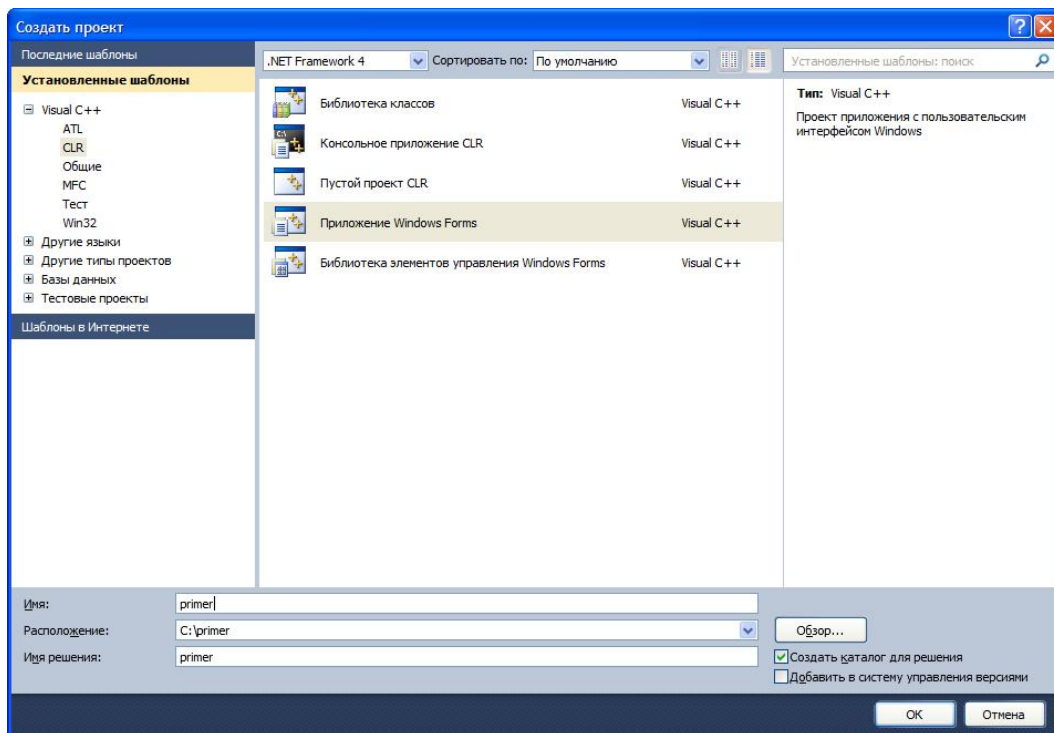


Рисунок 1.3 – Пример задания имени проекта

После нажатия кнопки **ОК** откроется окно

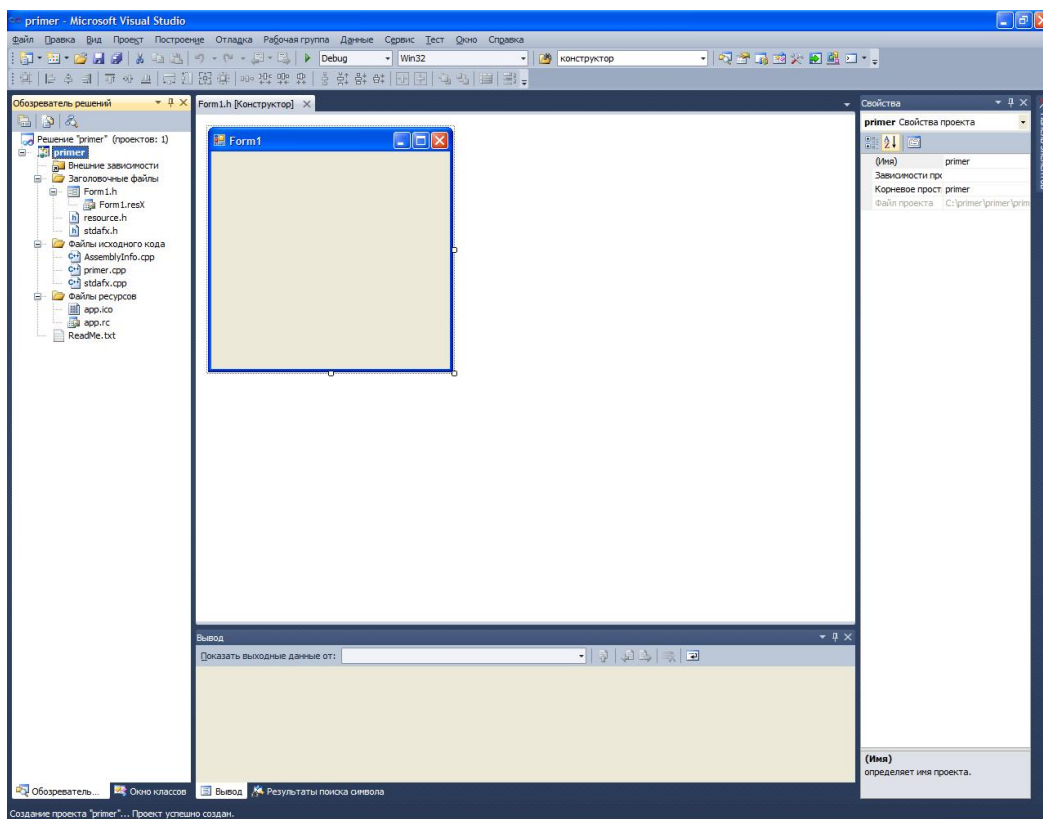


Рисунок 1.4 – Окно подготовленное к созданию приложения на языке C

Для запуска программы на ее исполнение из окна редактора в **Visual**

Studio 2010 можно нажать клавишу **F5**. рис. 1.5 показан результат исполнения первой программы.

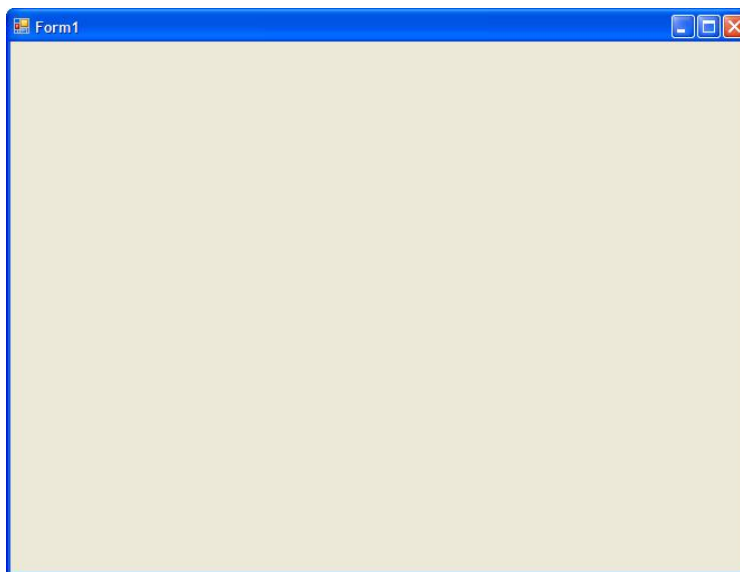



Рисунок 1.5 – Вывод формы программы на языке C

5 Легко убедиться, что появившееся на экране окно-форма обладает всеми свойствами окна *Windows*. Его можно перемещать по экрану, изменять размеры, сворачивать в значок и разворачивать на весь экран. Окно имеет стандартные интерфейсные элементы: заголовок, оконное меню, кнопки управления размером и кнопку закрытия. Однако на этом возможности окна и исчерпываются. Это естественно, ведь в проекте еще ничего, кроме формы, не создано.

6 Изменить значение свойства **Text** – занести текст, который будет показываться в заголовке формы при выполнении приложения. Например: «**Лаб 1. Выполнил Иванов И.И., группа ЭСА-10-1**».

7 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.

8 Вызвать панель элементов нажав на кнопку  или нажать комбинацию клавиш (Ctrl+Alt+X).

9 Поместить на форму элемент **Button1** (командная кнопка) и написать на ней «**Вывод текста**» (т.е. занести этот текст в свойство **Text**).

10 Аналогично создать кнопку **Button2** и написать на ней «**Запуск формы**» и кнопку **Button3** и написать на ней «**Выход**». С помощью свойства **Font** можно изменить размер шрифта и написание текста на кнопке. С помощью свойства **ForeColor** можно изменить цвет шрифта.

11 Поместить на форму компонент **listBox1** (многострочный редактор).

12 Поместить на форму компонент **GroupBox1** и в свойстве **Text** написать «**Стиль шрифта**». Это компонент-контейнер. Аналогично как и для кнопок с помощью свойств **Font** и **ForeColor** можно изменить размер шрифта, написание текста и цвет шрифта.

13 Разместить в контейнере **GroupBox1** четыре компонента типа **TcheckBox: CheckBox1, ... , CheckBox4**, подписав их в свойстве **Text** как: «**Полужирный**», «**Курсивный**», «**Подчеркнутый**» и «**Перечеркнутый**».

14 Поместить на форму компонент **TextBox1**.

15 Слева от компонента **TextBox1** поместить компонент **Label1** с подписью (свойство **Text**) «**Введите текст**». Также с помощью свойств **Font** и **ForeColor** можно изменить размер шрифта, написание текста и цвет шрифта.

16 Поместить на форму компонент **GroupBox2** и в свойстве **Text** этого компонента написать «**Цвет шрифта**». Это компонент-контейнер.

17 Разместить в контейнере **GroupBox2** пять компонентов-радиокнопок: **RadioButton1, ... , RadioButton5**, подписав их как «**Красный**», «**Синий**», «**Зеленый**», «**Желтый**» и «**Черный**». В соответствии с этими подписями установить цвет в свойстве **BackColor** (цвет фона) каждого из этих компонентов. Чтобы на этом фоне смотрелись надписи, для компонентов зеленого и черного цвета изменить цвет символов свойство **ForeColor** на белый.

18 Используя команду **Проект ⇒ Добавить новый элемент**, создать новую форму **Form2**, определив ее имя .

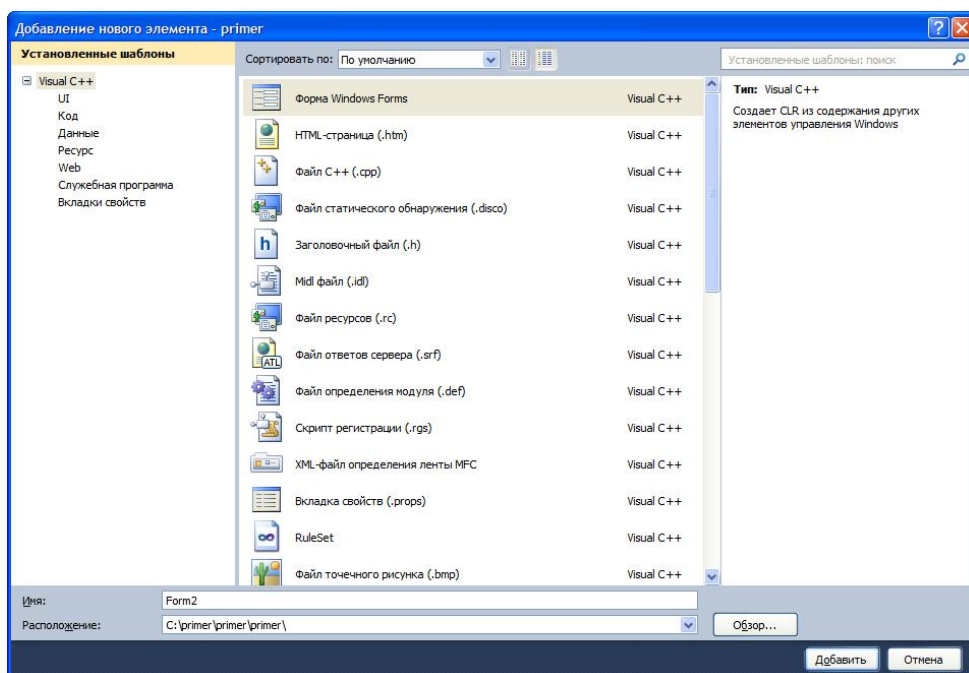


Рисунок 1.6 – Добавление новой формы в проект

19 В свойство **Text** этого компонента занести текст «*О программе*». Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение **FixedToolWindow**. Цветовое оформление выбрать по своему усмотрению (свойство **BackColor**).

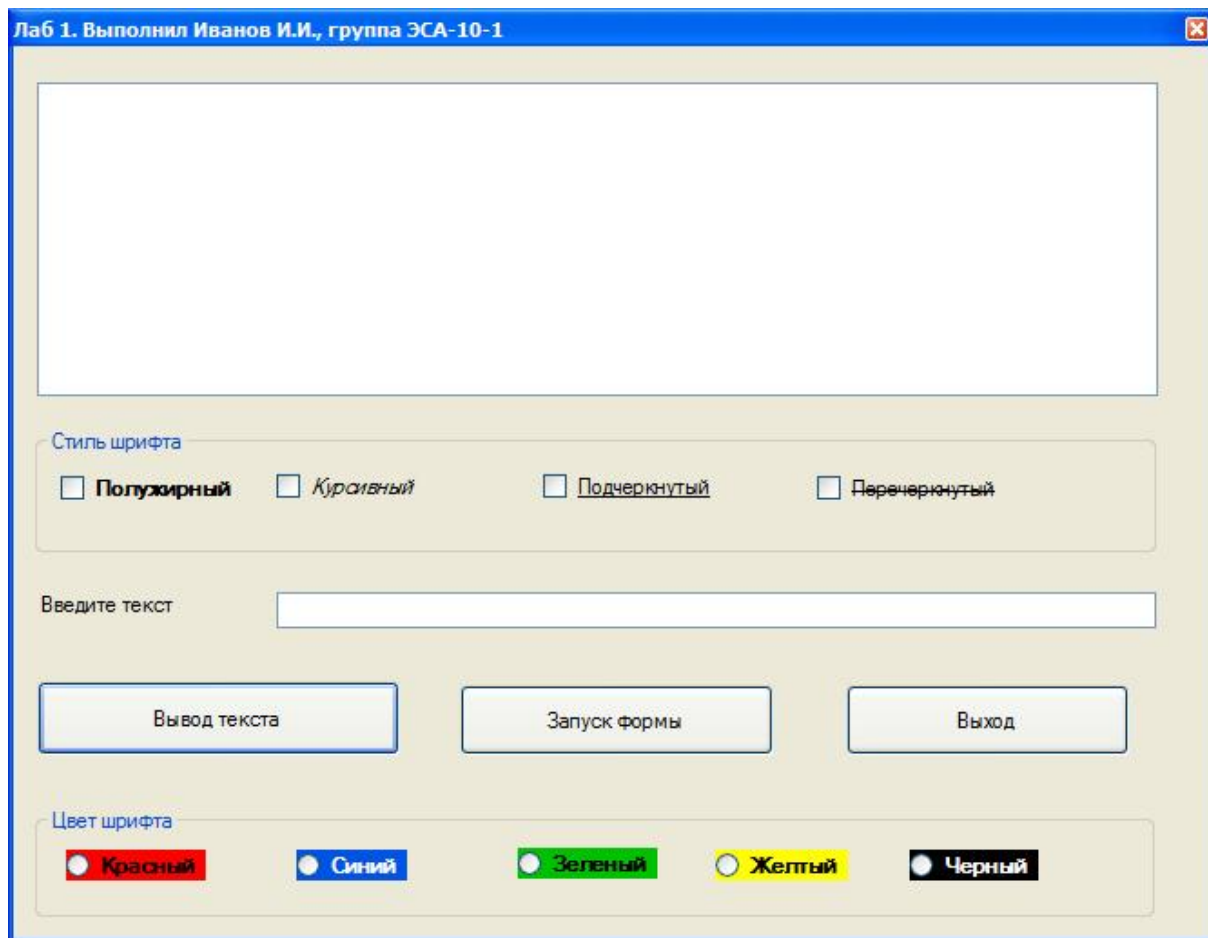


Рисунок 1.7 – Вид Form1 лабораторной работы

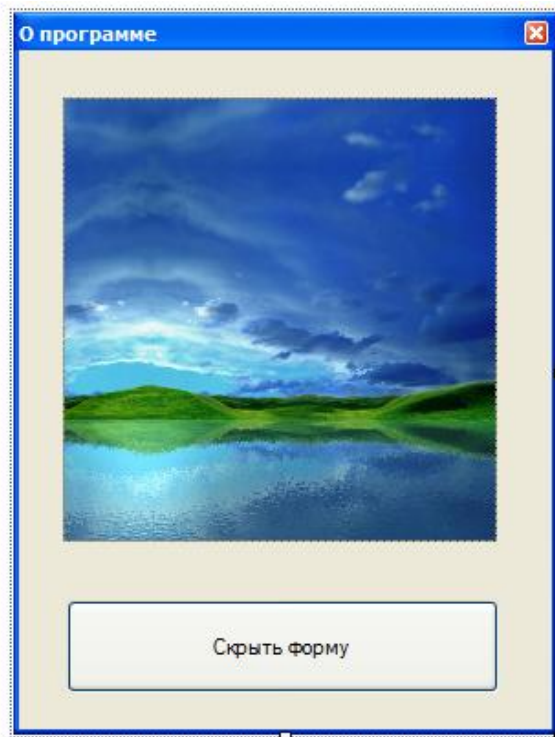


Рисунок 1.8 – Вид Form2 лабораторной работы


20 Поместить на новую форму компонент **Button1** (кнопка). На кнопке написать «**Скрыть форму**».

21 Поместить на форму **Form2** компонент **PictureBox1** (отображение картинок) и с помощью свойства **Image** загрузить в него соответствующий тематике рисунок. С помощью свойства **SizeMode** вписываем нашу картинку в блок картинки выбрав пункт **StretchImage**.

22 В результате проведенных операций должно получиться две формы примерно такого вида, как показано на рис. 1.7-1.8 (показан вид, который формы будут иметь на этапе выполнения).

Формы не должны обязательно иметь представленный выше вид, но общая идея лабораторной работы, т.е. знакомство с основными компонентами и их свойствами, должна быть соблюдена.

23 Запустить приложение на выполнение. Несмотря на наличие в форме всевозможных элементов управления, они не выполняют никаких действий. Например, можно выбрать любой стиль шрифта или цвет, но этот выбор нигде не отразится. Это естественно, т.к. в проекте еще не разработан ни один обработчик события, поэтому ни одно событие не обрабатывается, т.е. приложение не реагирует еще ни на одно событие. Форма **Form2** вообще никогда не сможет появиться на экране, несмотря на то, что она зарегистрирована в проекте. Наличие командной кнопки «**Запуск формы**» еще не обеспечивает ее показ на экране. Необходим соответствующий программный код, который бы в ответ на щелчок на этой кнопке показывал форму.

24 Сделать первую форму активной. Щелчком на компоненте **Button2** (командная кнопка «*Запуск формы*») выбрать (активизировать) его. В окне *Свойства* отображаются свойства и события выбранного компонента, т.е. при активизации **Button2** будут показаны свойства и события именно этого объекта. Перейти на вкладку **События** . Найти строку с именем события **Click** (Щелчок). Двойной щелчок в правой колонке этой строки (события) переводит в окно ввода кода обработчика этого события. Для объекта типа **Button** событие **button2_Click** является событием по умолчанию, код которого раскрывается двойным щелчком на объекте. Если обработчик события еще не разработан, *Visual Studio 2010* создает заготовку для его создания. Создать следующую процедуру обработки этого события (эта процедура показывает форму **Form2**, делая ее видимой). Для этого добавим строку **#include "Form2.h"** ссылку на **Form2** сразу после первой строки в коде программы **Form1**.

```
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    Form2^ f = gcnew Form2();
    this->Hide();
    f->ShowDialog();
    this->Show();
}
```

24 Сделать активной вторую форму (**Form2**). В окне *Свойства* перейти на вкладку **События**. Из списка компонентов, установленных на форме, выбрать **Button1**. Найти событие **Click**. Двойным щелчком по этому событию перейти в окно ввода кода. Создать процедуру обработки этого события, закрывающую (делающую невидимой) форму **Form2**:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {this->Hide();};
```

25 Запустить приложение на выполнение. Теперь при щелчке на командной кнопке **Button2** («*Запуск формы*») на экране появится форма **Form2**, т.к. событие Щелчок на этой кнопке будет обработано и этот обработчик покажет форму (сделает ее видимой). Аналогично, щелчок на соответствующей кнопке формы **Form2** уберет с экрана эту форму (сделает ее невидимой). Закрывать приложение.

26 Сделать окно первой формы активным. Щелчком на компоненте **Button1** (командная кнопка «*Вывод текста*») выбрать (активизировать) его. В окне *Свойства* перейти на вкладку **События**. Найти строку с именем события **Click** (Щелчок). Двойной щелчок в правой колонке этой строки (события) переводит в окно ввода кода обработчика этого события. Создать следующую процедуру обработки события:

```

private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
    { if (textBox1->Text!=""){
        this ->listBox1->Items->Clear();
        this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14, Sys-
tem::Drawing::FontStyle::Regular, Sys-
tem::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->listBox1->ForeColor = Sys-
tem::Drawing::Color::Black;
        this ->listBox1->Items->Add(this ->textBox1-
>Text);
            if (radioButton1->Checked == true){ this-
>listBox1->ForeColor = System::Drawing::Color::Red;}
            if (radioButton2->Checked == true){ this-
>listBox1->ForeColor = System::Drawing::Color::Blue;}
            if (radioButton3->Checked == true){ this-
>listBox1->ForeColor = System::Drawing::Color::Green;}
            if (radioButton4->Checked == true){ this-
>listBox1->ForeColor = System::Drawing::Color::Yellow;}
            if (radioButton5->Checked == true){ this-
>listBox1->ForeColor = System::Drawing::Color::Black;}
            if (checkBox1->Checked == true){this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14, Sys-
tem::Drawing::FontStyle:: Bold, Sys-
tem::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));}
            if (checkBox2->Checked == true){this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14, Sys-
tem::Drawing::FontStyle:: Italic, Sys-
tem::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));}
            if (checkBox3->Checked == true){this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14, Sys-
tem::Drawing::FontStyle:: Underline, Sys-
tem::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));}
            if (checkBox4->Checked == true){this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14, Sys-
tem::Drawing::FontStyle:: Strikeout, Sys-
tem::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));}
            if ((checkBox1->Checked == true)&&(checkBox2-
>Checked == true)){this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Bold | System::Drawing::FontStyle::Italic)),
        System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));}
            if ((checkBox1->Checked == true)&&(checkBox2-
>Checked == true)&&(checkBox3->Checked == true))this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14,

```

```

static_cast<System::Drawing::FontStyle>(((System::Drawing::FontS
type::Bold | System::Drawing::FontStyle::Italic)
    | System::Drawing::FontStyle::Underline)),
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox1->Checked == true)&&(checkBox2-
>Checked == true)&&(checkBox3->Checked == true)&&(checkBox4-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>(((System::Drawing::Font
Style::Bold | System::Drawing::FontStyle::Italic)
    | System::Drawing::FontStyle::Underline)
    | System::Drawing::FontStyle::Strikeout)),
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox2->Checked == true)&&(checkBox3-
>Checked == true)&&(checkBox4->Checked == true))this->listBox1-
>Font = (gcnew System::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>(((System::Drawing::FontS
type::Italic | System::Drawing::FontStyle::Underline)
    | System::Drawing::FontStyle::Strikeout))),
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox2->Checked == true)&&(checkBox3-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Italic | System::Drawing::FontStyle::Underline)),
    System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox2->Checked == true)&&(checkBox4-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Italic | System::Drawing::FontStyle::Strikeout))),
    System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox1->Checked == true)&&(checkBox3-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Bold | System::Drawing::FontStyle::Underline)),
    System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

        if ((checkBox3->Checked == true)&&(checkBox4-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Underline | System::Drawing::FontStyle::Strikeout))),
    System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));

```

```

        if ((checkBox1->Checked == true)&&(checkBox4-
>Checked == true))this->listBox1->Font = (gcnew Sys-
tem::Drawing::Font(L"Arial", 14,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontSt
yle::Bold | System::Drawing::FontStyle::Strikeout)),
        System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204));}
    else
{MessageBox::Show( "Заполните пожалуйста данные", "Ошибка ввода
данных",
        MessageBoxButtons::OK, MessageBoxIcon::Exclamation );}
}

```

27 Сделать окно первой формы активным. Щелчком на компоненте **Button3** (командная кнопка «*Выход*») выбрать (активизировать) его. В окне *Свойства* перейти на вкладку **События**. Найти строку с именем события *Click* (Щелчок). Двойной щелчок в правой колонке этой строки (события) переводит в окно ввода кода обработчика этого события. Создать следующую процедуру обработки события:

```

private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e)
    { Application::Exit();}

```

28 Нажав клавишу **F5**, запустить приложение на выполнение. В случае необходимости выполнить отладку. Объяснить логику функционирования приложения.

29 По результатам работы составить отчет.

ЛАБОРАТОРНАЯ РАБОТА 2

Условные операторы. Вычисление значения функции, заданной условно

Цель: Создание простого законченного Windows приложения, включающего ввод исходных данных, расчет и вывод результатов расчета на форму. Знакомство с возможностями *Visual Studio 2010*. Проектирование разветвляющегося вычислительного процесса.

2.1 Пример выполнения работы

Создать приложение для вычисления значения функции:

$$y = f(x) = \begin{cases} \sin(x^2 + ax), & \text{если } x \leq 0 \\ 1 - \frac{1 + \sqrt{(x^2 + ax)}}{e^{\sin(x)}(1+x)}, & \text{если } 0 < x \leq a \\ \frac{\cos(x^2 - a^2)}{\sqrt{1 - \sin(a-x)}} - \frac{1 - \sin(a-x)}{e^{\sin(x)}}, & \text{если } x > a \end{cases}$$

1 Войти в среду *Visual Studio 2010*.

2 В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.

3 Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, **Lab2**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор** (например N:\СИ\2_trim\Lab2).

4 Для формы изменим значение свойства **Text** занеся следующие данные – «*Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 2*».

5 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.

6 Поместить на форму компонент **Button1** (командная кнопка) и написать на ней «*Расчет*» (т.е. занести этот текст в свойство **Text**).

7 Аналогично создать кнопку **Button2** и написать на ней «*Выход*». С помощью свойства **Font** можно изменить размер шрифта и написание текста на кнопке. С помощью свойства **ForeColor** можна изменить цвет шрифта см. рис. 2.1.

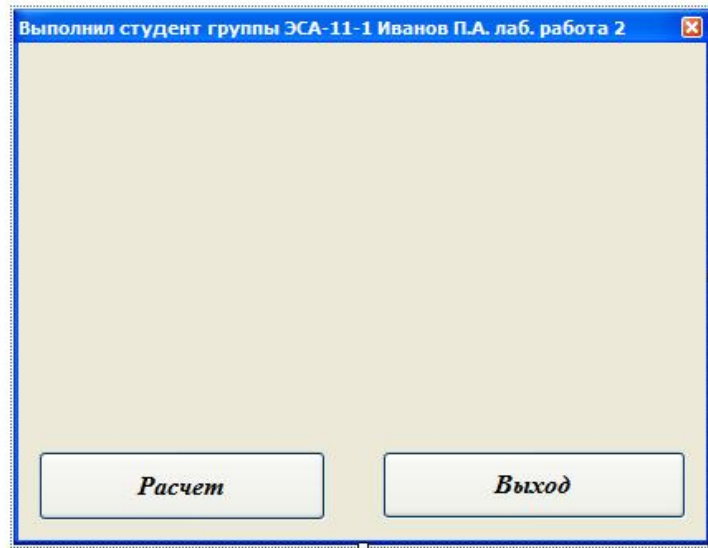


Рисунок 2.1 – Окно формы на этапе создания программы

8 Разместим в самом верху формы компонент **Label**. В свойство **Text** занесем текст «*Вычисление значения функции*» (также можно поменять параметры шрифта).

Для выполнения расчетов необходимо задать исходные данные: значение параметра **a** и значение переменной **x**. Эти данные будут вводиться с клавиатуры с помощью компонента однострочный редактор класса **TextBox**. Свойство этого компонента, в которое помещается результат ввода, имеет текстовый тип, т.е. допускает ввод любых символов. Таким образом, в процессе ввода пользователь может ввести (случайно или преднамеренно) недопустимые (нечисловые) символы. Такую ситуацию допустить нельзя, т.к. при вводе недопустимых символов произойдет сбой (аварийное завершение) программы во время преобразования неверных данных к вещественному типу. Поэтому пользователю нужно внимательно вводить данные в эти поля. Для этого выше кнопок разместим три элемента **TextBox** (можно один элемент **TextBox** вставить остальные два скопировать с первого). Первые два предназначены для ввода соответственно значения параметра **a**, и значения переменной **X**. В свойства **Text** этих компонентов ввести какие-либо значения – значения по умолчанию. Эти значения будут показываться при запуске приложения на выполнение. При выполнении приложения их можно будет заменить другими. Компонент **TextBox3** будет использован для вывода результата вычисления функции. Поэтому необходимо запретить ввод в него данных пользователем. Для этого свойству **ReadOnly** (только чтение) присвоить значение **true**, запрещающее пользователю заносить в компонент какие-либо данные.

9 Напротив первого элемента **TextBox1** разместим элемент **Label2** и в свойстве **Text** этого элемента занесем текст «*Введите значение A*», напротив **TextBox2** разместим элемент **Label3** и в свойстве **Text** занесем текст «*Введите значение X*» и напротив последнего **TextBox3** разместим элемент **La-**

bel4 и в свойстве *Text* занесем текст «Значение функции Y=». Получим следующую форму на рис. 2.2.

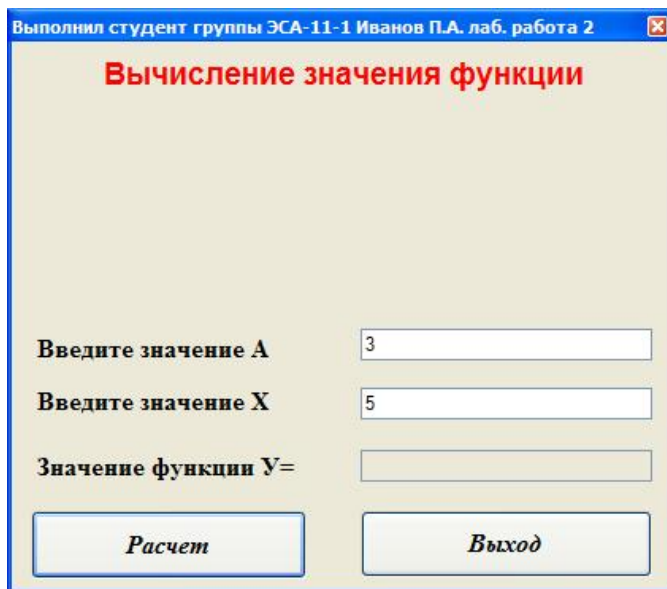


Рисунок 2.2 – Примерный вид проектируемой формы

10 Запустим текстовый процессор **Microsoft Word** и в нем наберем математическую формулу в соответствии с заданием с помощью редактора формул. Выделим набранную формулу и скопируем ее в буфер обмена.

11 Запустим графический редактор **Paint** (*Пуск* → *Программы* → *Стандартные* → *Paint*) и вставим из буфера обмена набранную формулу (рис. 2.3).

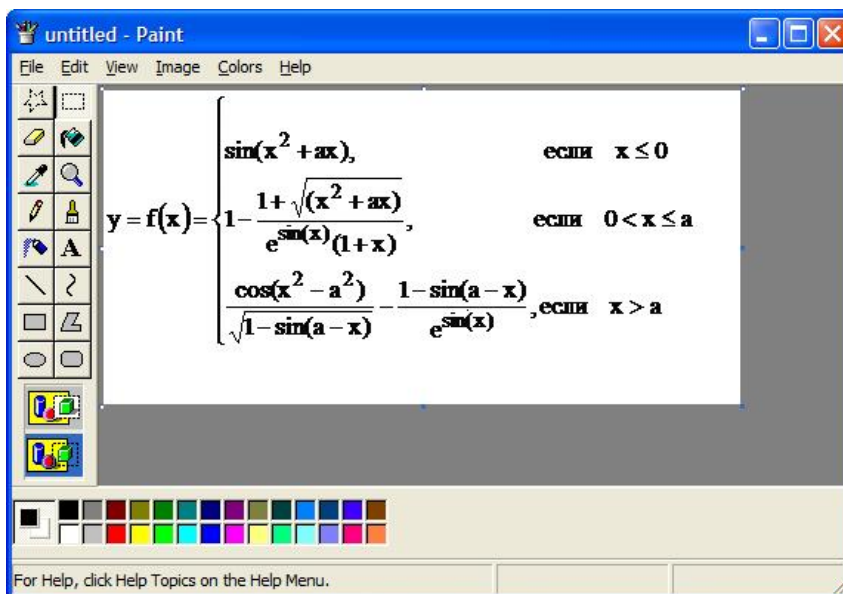


Рисунок 2.3 – Окно графического редактора Paint

12 Сохраним графический файл (на английском языке) в ту же папку что и проект программы (**Lab2**) .

13 Поместим на форму компонент **PictureBox1** и с помощью свойства **Image** загрузим в него созданный нами рисунок, отображающий функцию. Загрузка рисунка производится с помощью стандартного окна диалога. С помощью свойства **SizeMode** вписываем нашу картинку в блок **PictureBox1** выбрав пункт **StretchImage**.

14 В результате проведенных операций должна получиться форма примерно такого вида, как показано на рис. 2.4.

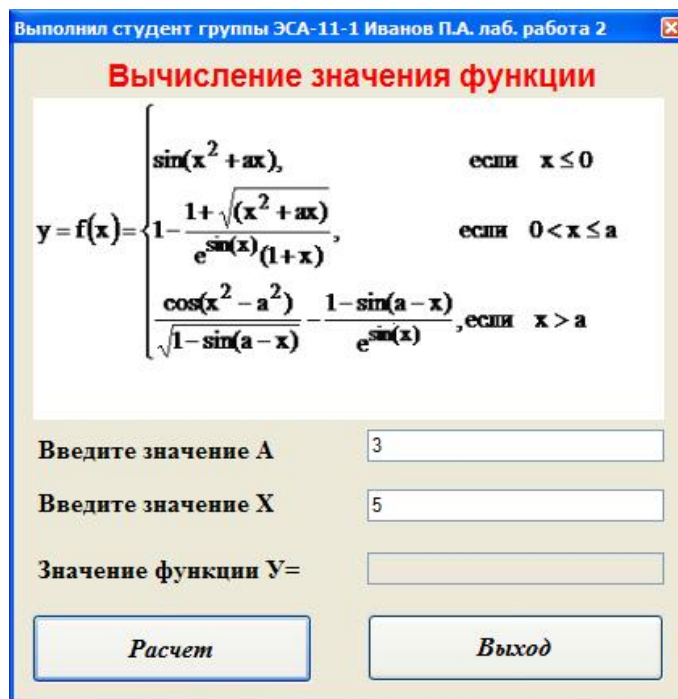


Рисунок 2.4 – Примерный вид формы приложения

15 Создадим событие **Click** для кнопки **Button1** с надписью «**Расчет**». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button1**. Получим заготовку подпрограммы:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
```

16 Между двумя фигурными скобками вставим следующий фрагмент программного кода:

```
double y, x, a;
//Проверка что данные для x и a введены
if ((textBox1->Text!="") && (textBox2->Text!=""))
{
//Преобразование из текста в дробное число x и a
a = Convert::ToDouble(textBox1->Text);
x = Convert::ToDouble(textBox2->Text);
if (x<=0){y=sin(x*x+a*x);}
else
```

```

if ((x>0)&&(x<=a)) {y=1-(1+sqrt(x*x+a*x))/(exp(sin(x))*(1+x));}
else
if (x>a) {y=(cos(x*x-a*a)/(sqrt(1-sin(a-x)))-(1-sin(a-x))/(exp(sin(x))))};}
//Вывод в компоненте TextBox3 преобразованного в текст значения y
textBox3->Text = Convert::ToString(y);
else
//Вывод окна с сообщением если не введены данные в компоненты TextBox1 и TextBox2
{MessageBox::Show("Введите пожалуйста данные А и Х", "Ошибка ввода данных",
MessageBoxButtons::OK, MessageBoxIcon::Exclamation); }

```

17 Создадим событие **Click** для кнопки **Button2** с надписью «**Выход**». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button2**. Получим заготовку подпрограммы:

```

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
}

```

18 Между двумя фигурными скобками вставим следующий фрагмент программного кода:

```

Application ::Exit();

```

19 Вверху программы после слов `#pragma once` добавим строку подключив библиотеку работы с математическими функциями `#include <math.h>`

20 Таким образом получим полноценную программу для вычисления значения функции используя условный оператор `if`.

21 Запустим программу на выполнение нажав на функциональную кнопку **F5** и получим следующий вид окна рис. 2.5.

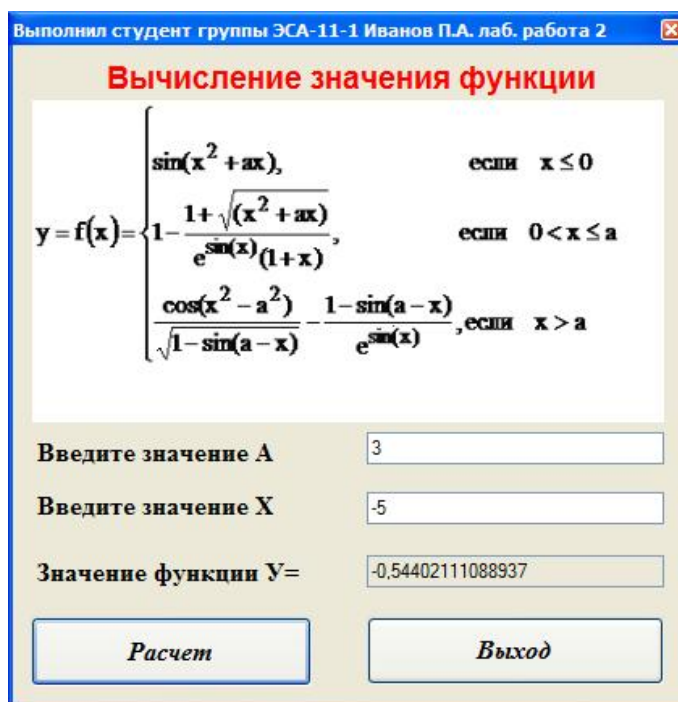


Рисунок 2.5 – Рабочий вид формы приложения

2.2 Рабочее задание

Создать приложение для вычисления и вывода на экран значения функции:

$$y = f(x) = \begin{cases} f_1(x), & \text{если } x \leq 0 \\ f_2(x), & \text{если } 0 < x \leq a \\ f_3(x), & \text{если } x > a \end{cases}$$

Выражения для функции $f_1(x)$, $f_2(x)$ и $f_3(x)$ выбрать из таблицы 2.1 в соответствии с номером своего варианта. В форме предусмотреть поля для ввода значения параметра a и переменной x , вывода результата вычисления y , а также командные кнопки для осуществления расчета и выхода из приложения.

Таблица 2.1 – Индивидуальные задания

Вариант	$f_1(x)$	$f_2(x)$	$f_3(x)$
1	$ x ^{2x+1}$	$\sin x^2$	$\ln^2 x + \sqrt{x}$
2	$\sin^2 x^3$	$\sqrt[5]{6x - x^2 + 1}$	$2 \sin(x - e^{-x})$
3	$2xe^{-x}$	$(x-1)^3 + \cos(x^3)$	$2\sqrt{x^3} \sin(x^3)$
4	$\ln(x^2 + 5)$	$\sin(e^x + 2)$	$\operatorname{tg}(5x+1)$
5	$2\sqrt{ x^3 } \sin(x^3)$	$(x+1)^2 \cos x^3$	$\sqrt{x^4 + 2} + \sin x^2$
6	$\cos x + x^3$	$\sqrt{x^3} \sin x$	$8 + \cos(3x)$
7	$x \sin(x+4)$	$\ln(4x^2 + 1)$	$\ln \sqrt[5]{5 + x^2}$
8	$x^4 + 2x^3 - x$	$1,3\sqrt{4 + x^2}$	$ x + 1 ^x$
9	$ x ^5 \operatorname{ctg} x $	$\ln(x^2 + 1)$	$e^{-2x} - \sqrt[3]{ x+1 }$
10	$x^5 \operatorname{ctg}(2x^3)$	$\sqrt[5]{x^4 + 3}$	$ \sin^2 x + 1 ^{2x}$
11	$\operatorname{ctg}(3x-1)^2$	$2 + xe^{-x}$	$\sin^3 x^2$
12	$x \sin(x-1)$	$(x-1)^3 + \cos x^3$	$\sqrt{ x ^3} \sin x^3$
13	$(x+1)/(x +2)^3$	$e^x + \cos(x+2)$	$3 \ln \sqrt[5]{\sin^2 x + 2}$
14	$3x^5 - \operatorname{ctg} x^3$	$\ln(\sin 4x + 1)^2$	$\sqrt[3]{2x^2 + x^4 + 1}$
15	$1,3\sqrt{4 + x^2}$	3^{x+3}	$x^{x+1} \sin(x+2)$

Продолжение таблицы 2.1

16	$e^{-3x} + \cos x$	$\sin^3 x^4$	$e^{-x} + \sqrt[3]{3x^2 + 1}$
17	$x^3 + (x + 1)^{0,1x}$	$(x-1)^3 + \cos(2x^3)$	$\sin(7x) + \operatorname{tg}(0,01x)$
18	$\operatorname{tg}(0,1\pi x^2) + x$	$e^{x+1} - \sin(x + \pi)$	$3\sqrt[5]{\sin^2 x + 2}$
19	$3x^5 - \operatorname{ctg}(\pi x^3)$	$(x+1)^{0,3} + \sin 2x^3$	$5x - x^2$
20	$ x ^{\sin(x)} + \sin(x)$	$3^{x+3} + 2x$	$2^x + \sin(\pi x)$
21	$x^2 + \sin(7x)$	$ x^3 + 10^x $	$\sqrt[7]{2x^4 + x^2 + 1}$
22	$\left \sqrt[3]{\frac{2x+5}{x^3+2}} \right $	$\frac{5x+x^2}{(x^2+3)^3}$	$\cos^2(x^3 + \sqrt{x})$
23	$\sqrt[5]{x^2 + x + 1}$	$\ln^2(\sqrt{x+5})$	$\sin(x^2) + x^{0,25}$
24	$\sqrt[3]{ x + 2} - 1$	$\sin(x^3) + x^{0,5}$	$\ln^2(x) + \sqrt{x}$
25	$\sqrt{\sin^2 x + \cos^4 x}$	$\ln^2(x) + \sqrt{x}$	$\operatorname{tg}^2(x) + \sqrt{x}$
26	$x^3 - \ln(x + 1)$	$\frac{2x+2}{(\operatorname{tg}(2x-1)+1)}$	$x^4 - x^x$
27	$x^4 + 2x^3 - x$	$e^{-x} + \sqrt[4]{x}$	$\ln(x^3 + x^2)$
28	$\frac{(3x-1)^2}{x^5}$	$\ln^2 \sqrt{x+5} $	$\cos(\sqrt{1+x^2})$
29	$x^5 \operatorname{ctg}(2x^3)$	$\frac{5}{\operatorname{tg}(2x+3)+1}$	$\operatorname{tg}(x^2+1)e^{-x}$
30	$ x \sin(3x)$	$x^3 \cos(x+2)$	$\sin x^2 + x^{0,25}$

ЛАБОРАТОРНАЯ РАБОТА 3

Циклический алгоритм. Табулирование функции и поиск экстремумов

Цель: изучение операторов цикла; создание циклических алгоритмов; приобретение навыков создания проекта вывода данных в виде таблицы в *Visual Studio 2010*.

3.1 Пример выполнения работы

Создать приложение для вычисления значения функции:

$$y = f(x) = \begin{cases} 2x + 2, & \text{если } x \leq 0 \\ \sqrt{x + 3}, & \text{если } 0 < x \leq a \\ \cos^2(x + 2), & \text{если } x > a \end{cases}$$

1 Войти в среду *Visual Studio 2010*.

2 В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.

3 Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, **Lab3**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор** (например N:\СИ\2_trim\Lab3).

4 Для формы изменим значение свойства **Text** занеся следующие данные – «*Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 3*».

5 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.

6 Поместим на форму два компонента **Button1** и **Button2** (кнопка), выше этих компонентов левее центра разместим четыре поля для ввода данных **TextBox1**, **TextBox2**, ..., **TextBox4**.

7 Напротив компонентов **TextBox** разместим четыре элемента **Label1**, **Label2**, ..., **Label4**.

8 В верхнем левом углу формы расположим компонент **PictureBox1** (блок для картинки).

9 Напротив компонента **PictureBox1** разместим компонент **DataGridView1** (таблица для просмотра данных).

10 Над компонентом **PictureBox1** разместим элемент **Label5**.

11 Под компонентом **PictureBox1** разместим два компонента **Label6** и **Label7**, а под ними два компонента для вывода экстремумов функции **TextBox5**, **TextBox6**. Получим следующую форму с размещенными на ней компонентами см. рис. 3.1.

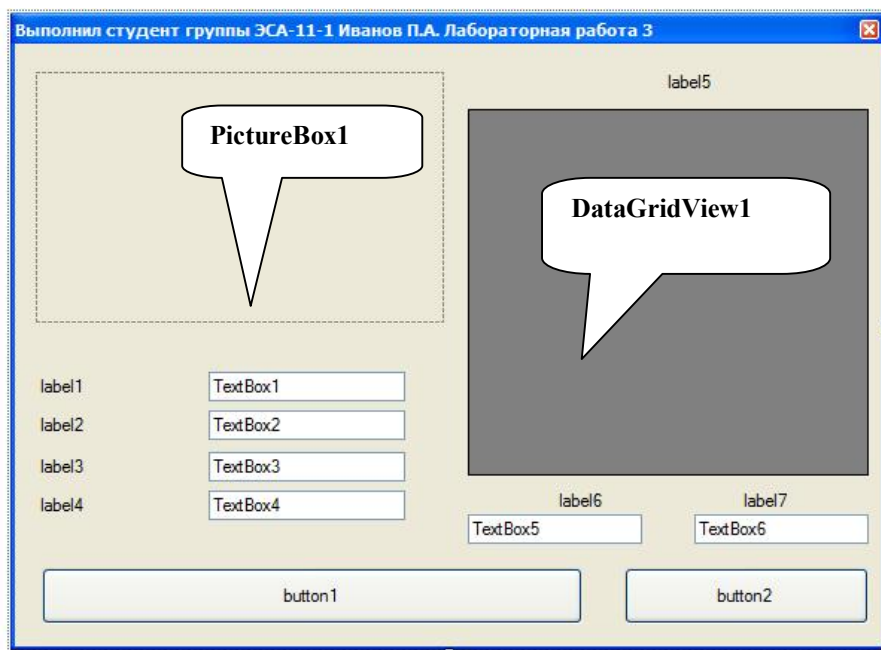


Рисунок 3.1 – Окно формы на этапе создания программы

12 Выделим командную кнопку **Button1** и напишем на ней «*Показать таблицу и экстремумы функции*» (т.е. занести этот текст в свойство **Text**), аналогично для кнопки **Button2** сделаем надпись «*Выход*».

13 Выделим компонент **Label1** и в свойство **Text** занесем текст «*Введите значение XN =*» (начальное значение интервала), аналогично для **Label2** в свойство **Text** занесем текст «*Введите значение XK =*» (конечное значение интервала), для **Label3** занесем текст «*Введите значение XH =*» (шаг изменения значения **X**), для **Label4** в свойство **Text** занесем текст «*Введите значение a =*» (положительное число меньше XK).

14 Выделим компонент **Label5** и в свойство **Text** занесем текст «*Таблица значений функции*»,

15 Выделим компонент **Label6** и в свойство **Text** занесем текст «*Максим. знач. функции*», аналогично для **Label7** в свойство **Text** занесем текст «*Миним. знач. функции*».

16 Для выполнения расчетов необходимо задать исходные данные: значение параметра **a** и значения переменных **XN**, **XK**, **XH**. Эти данные будут вводиться с клавиатуры с помощью компонента однострочный редактор класса **TextBox**. В свойства **Text** этих компонентов ввести какие-либо значения – значения по умолчанию. Эти значения будут показываться при запуске приложения на выполнение. При выполнении приложения их можно будет заменить другими.

17 Компоненты **TextBox5** и **TextBox6** будут использованы для вывода результатов нахождения экстремумов функции (максимального и минимального значения функции). Поэтому необходимо запретить ввод в него данных пользователем. Для этого свойству **ReadOnly** (только чтение) присвоить значение **true**, запрещающее пользователю заносить в компонент какие-либо данные. Получим следующую форму на рис. 3.2.

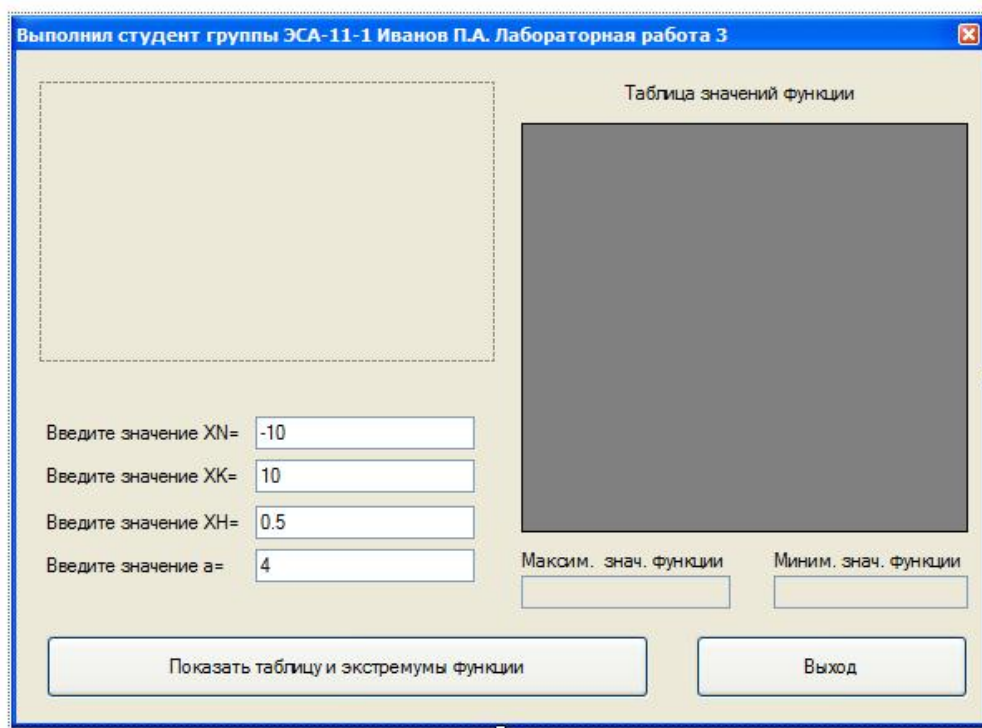


Рисунок 3.2 – Примерный вид проектируемой формы

18 Запустим текстовый процессор **Microsoft Word** и в нем наберем математическую формулу в соответствии с заданием с помощью редактора формул. Выделим набранную формулу и скопируем ее в буфер обмена.

19 Запустим графический редактор **Paint** (*Пуск → Программы → Стандартные → Paint*) и вставим из буфера обмена набранную формулу. Сохраним графический файл (на английском языке) в ту же папку что и проект программы (**Lab3**).

20 Выделим на форме компонент **PictureBox1** и с помощью свойства **Image** загрузим в него созданный нами рисунок, отображающий функцию. Загрузка рисунка производится с помощью стандартного окна диалога. С помощью свойства **SizeMode** вписываем нашу картинку в блок **PictureBox1** выбрав пункт **StretchImage**.

21 Выделим компонент **DataGridView1** и для свойств **RowHeadersVisible** (отображение заголовка строк) и **ColumnHeadersVisible** (отображение заголовка столбцов) выберем параметр **False**.

22 В результате проведенных операций должна получиться форма примерно такого вида, как показано на рис. 3.3.

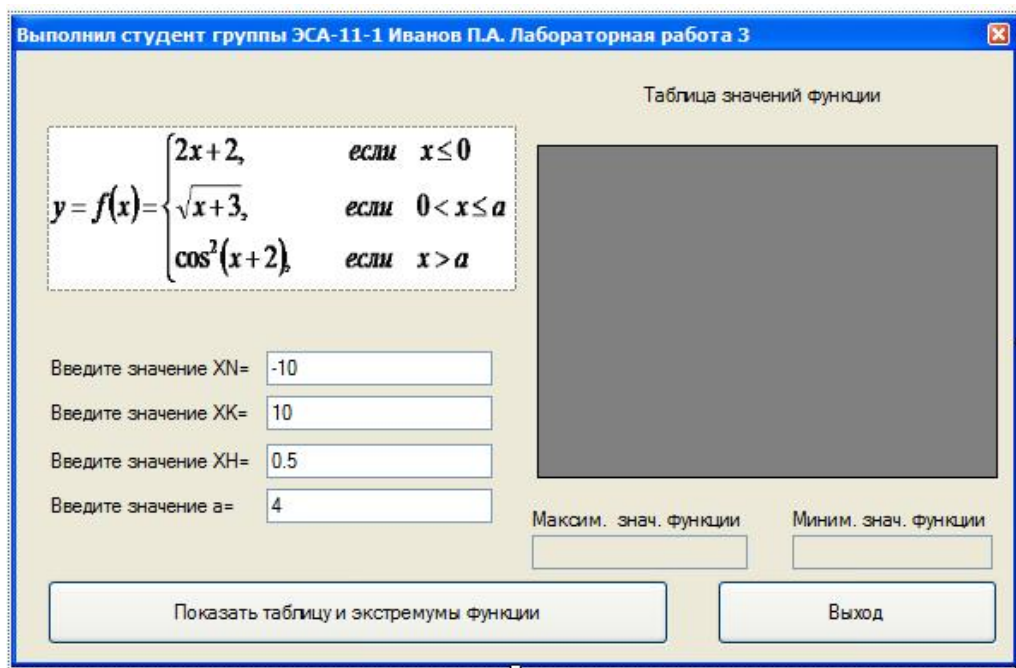


Рисунок 3.3 – Примерный вид формы приложения

22 Перейдем к коду программы и в самом верху второй строкой кода программы подключим библиотеку использования математических функций, для этого вставим

```
#include <math.h>
```

23 Создадим событие *Click* для кнопки **Button1** с надписью «*Показать таблицу и экстремумы функции*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button1**. Внесем изменения в код подпрограммы и получим следующий вид программы:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    double xn, xk, xh, x, y, a, ymax, ymin, yt;
    int n, i;
    //Проверка ввода данных в компоненты textBox
    if ((textBox1->Text!="") && (textBox2->Text!="") &&
        (textBox3->Text!="") && (textBox4->Text!="") )
    {
        //Преобразование введенных данных в тип double
        xn = Convert::ToDouble(textBox1->Text);
        xk = Convert::ToDouble(textBox2->Text);
        xh = Convert::ToDouble(textBox3->Text);
        a = Convert::ToDouble(textBox4->Text);
        //Создание таблицы
        int k =0;
```

```

    k = dataGridView1->ColumnCount;
    if (k != 0)
        for (int i = 0; i < k; i++)
            //Очистка столбцов таблицы
            dataGridView1->Columns->Clear();

    //Создание двух столбцов в таблице
    dataGridView1->ColumnCount = 2;

    //Создание в таблице строк
    dataGridView1->Rows->Add(ceil((xk-xn)/xh)+1);

    //Занесение в верхнюю строку таблицы первой ячейки текста X, в другую текста Y
    dataGridView1->Rows[0]->Cells[0]->Value = Convert::ToString("    x");
    dataGridView1->Rows[0]->Cells[1]->Value = Convert::ToString("    Y");
    i=1;
    x=xn;
    ymax=-1.8e307; ymin=1.8e307;
    while (x<=xk)
        {
            if (x<=0) { y=2*x+2; }
            else
                if (x<=a) {y=sqrt(x+3); }
                else
                    if (x>a) {y=pow(cos(x+2), 2); }

            //Занесения в первый столбец значений аргумента X
            dataGridView1->Rows[i]->Cells[0]->Value = Convert::ToString(x);

            //Переменной yt присваивает округленное до двух знаков после запятой значение y
            yt=ceil(y*100)/100;

            //Вывод во втором столбце таблицы значение функции Y
            dataGridView1->Rows[i]->Cells[1]->Value = Convert::ToString(yt);

            //находит максимальное и минимальное значение и округляет до двух знаков после запятой
            if (y>ymax) ymax=ceil(y*100)/100;
            if (y<ymin) ymin=ceil(y*100)/100;
            x=x+xh;
            i++;}

    //выводит в компоненты textbox максимальное и минимальное значение функции
    textBox5->Text = Convert::ToString(ymax);
    textBox6->Text = Convert::ToString(ymin);
}

else {MessageBox::Show("Заполните пожалуйста данные", "Ошибка ввода данных",
    MessageBoxButtons::OK, MessageBoxIcon::Exclamation );}

}

```

24 Создадим событие *Click* для кнопки **Button2** с надписью «**Выход**». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **But-**

ton2 и вставим код для выхода из программы (уже использовался в лаб. раб. 2).

25 Запустим программу на выполнение нажав на функциональную кнопку **F5** и получим следующий вид окна рис. 3.4.

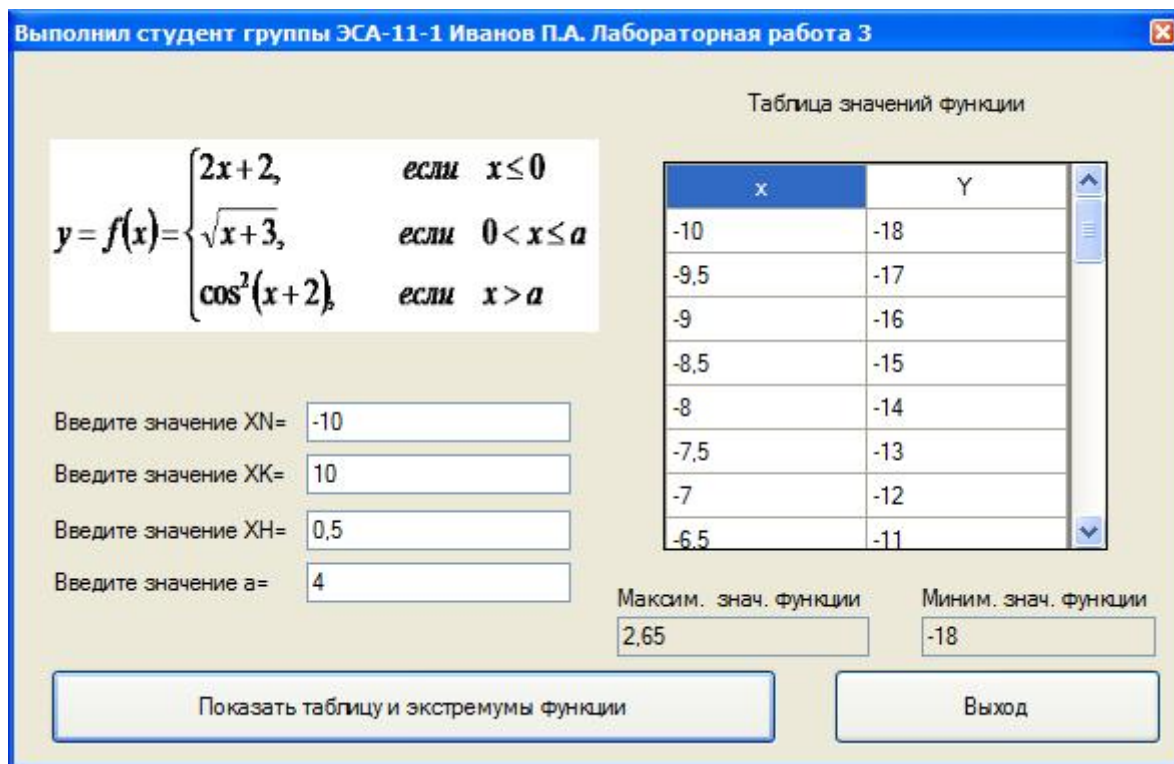


Рисунок 3.4 – Рабочий вид формы приложения

3.2 Рабочее задание

Создать Windows-приложение для вычисления значения функции и нахождения экстремумов функции

$$y = \begin{cases} f1(x), & \text{если } x \leq 0 \\ f2(x), & \text{если } 0 < x \leq a \\ f3(x), & \text{если } x > a \end{cases}$$

с использованием оператора *while* на отрезке $[xn; xk]$ с шагом xh . Данные взять с таблицы 3.1.

Таблица 3.1 – Индивидуальные задания

Вариант	Функции			Границы отрезка	Шаг табулирования
	$f1(x)$	$f2(x)$	$f3(x)$		
1	$\sqrt[5]{x^2 + x + 1}$	$\ln^2(\sqrt{x+5})$	$\sin(x^2) + x^{0,25}$	[-3,9; 3,8]	0,15
2	$3x^5 + \operatorname{ctg}(x^3 + 1)$	$e^{x+1} - \sin(\pi x)$	$\sqrt[5]{\sin^2 x + 2}$	[-1,3; 7,1]	0,6
3	$x^5 - \operatorname{ctg}(\pi x^3)$	$(7x +1)^{0,3} + \sin x$	$5x - x^2$	[-2,9; 6,2]	0,8
4	$ x ^{x+2} + \sin(x)$	$3^{x+3} + 2x$	$\sqrt[5]{x^2 + x + 1}$	[-3,7; 8,5]	0,11
5	$x^2 + \sin(7x) - 1$	$ x^3 + 10^x $	$\sqrt[7]{2x^4 + x^2 + 1}$	[-3,9; 1,2]	0,25
6	$\sqrt[3]{ x + 2} - 1$	$\sin(x^2) + x^{0,25}$	$\ln^2(x) + \sqrt{x}$	[-4,5; 6,1]	0,3
7	$\sqrt{ \sin^2 x + \cos^4 x }$	$\ln(x+1) + \sqrt{3x}$	$5^{x+1} + \operatorname{tg}(x+3)^2$	[-3,4; 3,4]	0,33
8	$x^3 - 3x^2\sqrt{ x +6}$	$\frac{2x+2}{\operatorname{tg}(2x-1)+1}$	$x^4 - x^x$	[-4,1; 5,0]	0,45
9	$x^4 + 2x^3 - x$	$e^{-x} + \sqrt[3]{x}$	$\ln(x^3 + x^2)$	[-1,7; 2,9]	0,75
10	$\frac{(3x-1)^2}{x^5 + 2x + 1}$	$\ln^2 \sqrt{x+5} $	$\sqrt[5]{1+x^2}$	[-1,6; 4,7]	0,65
11	$x^5 \operatorname{ctg}(2x^3)$	$\frac{5}{\operatorname{tg}(2x+3)+1}$	$x^2 e^{-x}$	[-1,6; 3,7]	0,3
12	$x^x \operatorname{tg}(x+5)$	$x^3 \cos x$	$\sin x^2 + x^{0,25}$	[-2,8; 8,2]	0,4
13	$\ln x+1 + \sqrt{3 x }$	$\sin(x^2 + 3x)$	$\ln^2 x + \sqrt{x+3}$	[-1,7; 2,6]	0,25
14	$\sin^2 x^3$	$\sqrt[5]{6x - x^2 + 1}$	$\sin(x - e^{-x})$	[-2,2; 7,4]	0,23
15	$\cos(x^3 + 1)e^{-x}$	$\left \sqrt[3]{\frac{2x+5}{x^3+2}} \right $	$2\sqrt{x^2 + 7\sin(x^3)}$	[-1,1; 7,9]	0,8
16	$2\sqrt{ x^3 } \sin(x^3)$	$(x+1)^2 \cos x^3$	$\sqrt{x^4 + 2} + \sin x^2$	[-1,2; 2,6]	0,1
17	$\sin(x^5 + 3)$	$\sqrt{x^3} \sin x$	$x^4 - \sin(x+1)$	[-1,7; 2,4]	0,3

Продолжение таблицы 3.1

18	$x^4 \operatorname{tg}(x+2)$	$\ln(4x^2+1)$	$\ln \sqrt[5]{5+x^2}$	$[-4,3; 8,0]$	0,5
19	$x^5 + \sqrt[3]{x+10}$	$1,3\sqrt{4+x^2}$	$ x+1 ^x$	$[-9,1; 5,8]$	0,14
20	$ x ^5 \operatorname{ctg} 2x $	$\ln(x^2+1)$	$e^{-2x} - \sqrt[3]{ x+1 }$	$[-3,4; 2,5]$	0,23
21	$x^5 \operatorname{ctg}(2x^3)$	$\sqrt[5]{x^4+3}$	$ \sin^2 x + 1 ^{2x}$	$[-2,2; 8,1]$	0,15
22	$\operatorname{ctg}(3x-1)^2$	$2+xe^{-x}$	$\sin(x^3+1)$	$[-2,8; 5,2]$	0,5
23	$x^3 + 4x^2 \sqrt{ x }$	$(x-1)^3 + \cos x^3$	$\sqrt{ x ^3} \sin x^3$	$[-3,2; 7,8]$	0,36
24	$(2x+1)(x +2)^3$	$e^x + \sin(x+2)$	$3\ln \sqrt[5]{\sin^2 x + 2}$	$[-6,1; 1,3]$	0,15
25	$\operatorname{ctg}(x^3+1)$	$\ln(\sin x + 1)^2$	$\sqrt[3]{2x^2 + x^4 + 1}$	$[-7,4; 0,6]$	0,16
26	$1,3\sqrt{4+x^2}$	3^{x+3}	$5^{x+1} + \operatorname{tg}(x+1)$	$[-1,2; 7,1]$	0,45
27	$e^{2x} + \sin(2x^3)$	$\sin^3 x^4$	$e^{-x} + \sqrt[3]{3x^2+1}$	$[-2,2; 3,9]$	0,55
28	$x^3 + (x +1)^{0,1x}$	$(x-1)^3 + \cos(x^3)$	$2x + \operatorname{tg}(x^2+2)$	$[-0,3; 4,5]$	0,62
29	$\left \sqrt[3]{\frac{2x+5}{x^3+2}} \right $	$\frac{5x+x^2}{(x^2+3)^3}$	$\cos^2(x^3+\sqrt{x})$	$[-2,4; 4,4]$	0,4
30	$\ln(x^2+5)$	$\sin(e^x+2)$	$\frac{\sin(x+3)}{e^{2x} + \cos(x+1)}$	$[-2,9; 8,2]$	0,2

ЛАБОРАТОРНАЯ РАБОТА 4

Построение графика функции на промежутке с определенным шагом

Цель: научиться создавать Windows-приложения построения графика функции в среде *Visual Studio 2010* на определенном промежутке с указанным шагом используя условия проверки правильности ввода данных.

4.1 Пример выполнения работы

Создать Windows-приложение, которое предлагает пользователю ввести данные начала промежутка (x_1), конца промежутка (x_k) и шага изменения переменной (h) для построения графика функции $y = \cos(x)$ (сделать проверку правильности ввода данных).

Ход выполнения

- 1 Войдем в среду *Visual Studio 2010*.
- 2 В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.
- 3 Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, **Lab4**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор** (например N:\СИ\2_trim\Lab4).
- 4 Для формы изменим значение свойства **Text** занеся следующие данные – «*Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 4*».
- 5 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.
- 6 В верхней части окна разместим **Label1** и в свойстве **Text** занесем текст «*Построение графика функции на интервале $[x_1;x_k]$* ».
- 7 Ниже под **Label1** разместим компонент **Chart1**. Выделим его и зайдем в окно свойств, выберем свойство **Series** и нажмем на троеточие. Появится диалоговое окно см. рис. 4.1.

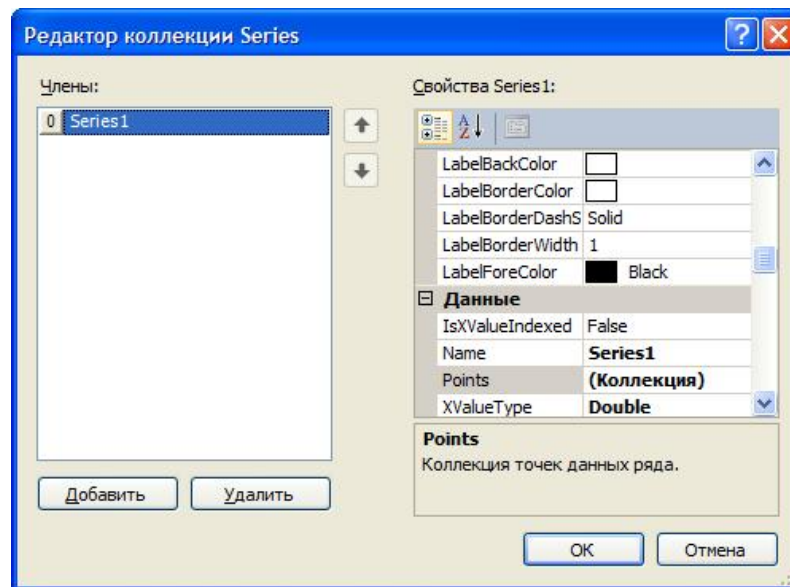


Рисунок 4.1 – Диалоговое окно свойства Series компонента Chart

8 Нажмем на кнопку добавить и у нас появится **Series1** в левой части окна (каждая серия способна строить на компоненте **Chart1** новый график, поэтому, если нужно построить несколько графиков на одной координатной плоскости, то нужно сделать соответствующее количество Series).

9 В правой части этого же окна выберем подпункт «Данные» свойства **Series1**, а в нем пункт **XValueType** и с помощью выпадающего списка выберем параметр «**Double**». Аналогично для пункта **YValueType** выберем параметр «**Double**».

10 Ниже в правой части этого же окна выберем подпункт «Диаграмма» свойства **Series1**, а в нем пункт **ChartType** и с помощью выпадающего списка выберем параметр «**Spline**» (это позволяет соединять точки кривой линией).

11 И последняя настройка компонента **Chart1** в правой части этого же диалогового окна выберем подпункт «Условные обозначения» свойства **Series1**, а в нем пункт **IsVisibleInLegend** и с помощью выпадающего списка выберем параметр «**False**» (это позволяет не отображать легенду в диаграмме).

12 Ниже компонента **Chart1** разместим четыре компонента **Label1**, **Label2**, **Label3**, **Label4**, **Label5**.

13 Выделим **Label2** и в свойстве **Text** занесем текст «Введите интервал построения графика». Аналогично выделим **Label3** и в свойстве **Text** занесем текст «Введите начальное значение $XN=$ », для **Label4** и в свойстве **Text** занесем текст «Введите конечное значение $XK=$ » и для **Label5** в свойстве **Text** занесем текст «Введите значение шага $XH=$ ».

14 Напротив компонента **Label3** разместим компонент **TextBox1**, напротив компонента **Label4** разместим компонент **TextBox2**, а напротив компонента **Label5** разместим компонент **TextBox3**.

15 В правом нижнем углу разместим две кнопки **Button1** и **Button2**.

16 Выделим компонент **Button1** и в свойство **Text** занесем текст «*Нарисовать график*». При нажатии на эту кнопку будет в компоненте **Chart1** рисоваться график функции $y = \cos(x)$.

17 Выделим компонент **Button2** и в свойство **Text** занесем текст «*Выход*».

18 В результате проведенных операций должна получиться форма примерно такого вида, как показано на рис. на рис. 4.2.

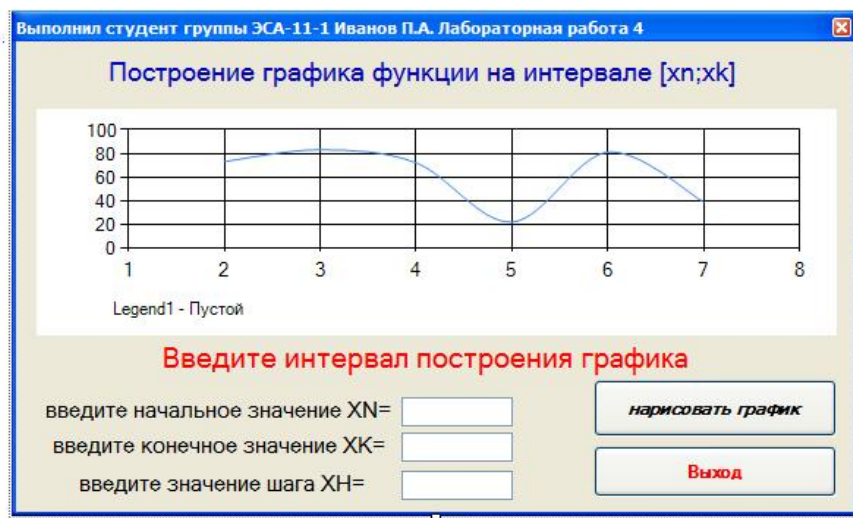


Рисунок 4.2 – Окно формы на этапе создания программы

19 Перейдем к коду программы и после строки

```
#pragma once
```

подключим библиотеки использования математических функций, для этого вставим следующую строку:

```
#include<cmath>
```

20 В разделе **using** добавить строку:

```
using namespace System::Windows::Forms::DataVisualization::Charting;
```

21 Создадим событие **Click** для кнопки **Button1** с надписью «*Нарисовать график*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button1**. Внесем изменения в код подпрограммы:

```

double x, xn, xk, xh;
    Series^ plot1 = chart1->Series[0];

//Очистка компонента Chart1
    plot1->Points->Clear();

//Проверка, что введены данные xn, xk, xh и их преобразование в переменные типа
Double
if ((textBox1->Text!="") && (textBox2->Text!="") && (textBox3->Text!=""))
{
    xn = Convert::ToDouble(textBox1->Text);
    xk = Convert::ToDouble(textBox2->Text);
    xh = Convert::ToDouble(textBox3->Text);

//Проверка правильности ввода данных
if ((xn>=xk) || (xh>(xk-xn)))
    {
    MessageBox::Show( "Данные заполнены неверно", "Ошибка          вво-
да данных", MessageBoxButtons::OK,          MessageBoxIcon );
    }
else
    {x=xn;
    while (x<=xk)
    {
//Нанесение точки с координатами X и Y в компоненте Chart1
        plot1->Points->AddXY(x, cos(x));
        x=x+xh;
    }
    }
else
    {
    MessageBox::Show( "Заполните пожалуйста данные", "Ошибка          вво-
да данных", MessageBoxButtons::OK,          MessageBoxIcon);
    }
}

```

22 Создадим событие *Click* для кнопки **Button2** с надписью «**Выход**». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button3** и в полученную заготовку подпрограммы вставим код выхода из программы (уже использовался в предыдущих лабораторных работах).

23 Запустим программу на выполнение нажав на функциональную кнопку **F5** и получим следующий вид окна рис. 4.3.

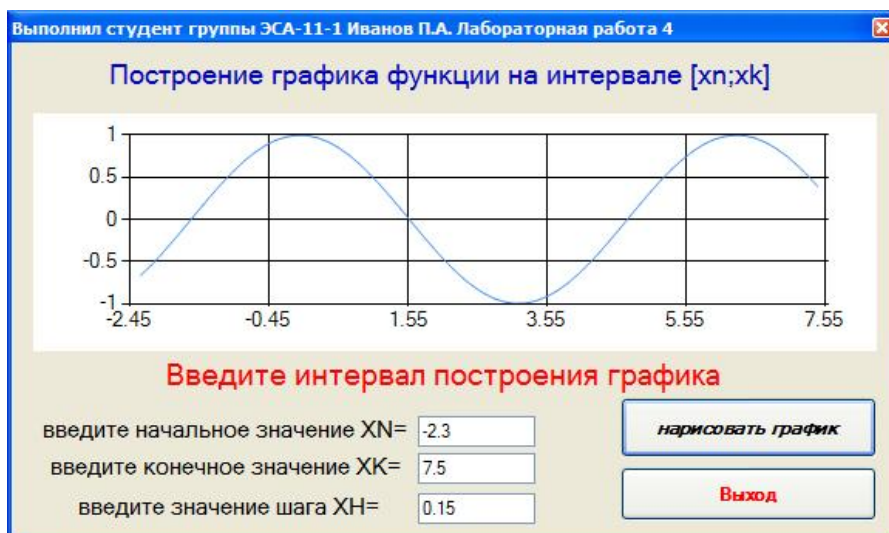


Рисунок 4.3 – Рабочий вид формы приложения

4.2 Рабочее задание

Создать Windows-приложение, которое предлагает пользователю ввести данные начала промежутка (x_n), конца промежутка (x_k) и шага изменения переменной (x_h) для построения графика функции (сделать проверку правильности ввода данных). Данные функции взять с таблицы 4.1.

Таблица 4.1 – Индивидуальные задания

Вариант задания	Функция $f(x)$	Начальное значение x (x_n)	Конечное значение x (x_k)	Шаг (x_h)
1	$\sqrt[3]{\frac{2x+5}{x^3+2}}$	2,1	16,5	0,2
2	$\frac{\sqrt{\sin(x^2+3)+4}}{x^2+2}$	-4,2	28,1	0,1
3	$\frac{\sin(x+2)^3}{\ln x^2+3x+1 }$	-1,7	45,3	0,3
4	$x^3 + (x +1)^{0,1x}$	-2,25	34,9	0,5
5	$\frac{\sin^3(x+2)}{\sqrt[4]{\sin^2 x + \cos^4 x}}$	2,45	25,2	0,14
6	$ 5 ^{\sin(x)+2} x + \sin(x)$	3,35	36,26	0,23

Продолжение таблицы 4.1

7	$\frac{\operatorname{tg}(x+3)^2}{ x ^{1,2} \sin 3x}$	1,7	4,9	0,15
8	$\frac{x^3 - 4x + 2}{x^2 + \sin(7x) - 1}$	-5,2	11,7	0,5
9	$\frac{\operatorname{tg}(0,1\pi x^2) + x}{\cos^2(2x+3)}$	2,9	17,48	0,36
10	$\frac{\cos(x^3 - 4x + 4)}{x^3 - \ln(x +1)}$	-1,9	29,7	0,15
11	$\frac{\sin(x+2)^2}{\sqrt[3]{2x^2 + x^4 + 1}}$	-2,74	28,29	0,16
12	$\frac{\sqrt{ x ^3} \sin x^3}{\cos^2(x+1)}$	-1,25	6,39	0,45
13	$\frac{\sin^2(x+5)^2}{e^{-x} + \sqrt[3]{3x^2 + 1}}$	-1,78	6,99	0,55
14	$\frac{x^4 + 2x^3 - x}{\cos(x+3)^2}$	-2,46	28,8	0,62
15	$\frac{\operatorname{tg}(x^2 + 4x - 1)}{2\sqrt{x^3} \sin(x^3)}$	3,75	17,7	0,4
16	$ x ^5 \operatorname{ctg}(x+2)$	-5,55	10,33	0,15
17	$\frac{5x + x^2}{(x^2 + 3)^3}$	-4,41	11,25	0,6
18	$\frac{\sin^2(x+3)}{x^5 - \operatorname{ctg}(\pi x^3)}$	2,17	19	0,8
19	$\frac{\sin(x+3)}{x^5 \operatorname{ctg}(2x^3)}$	8	12,1	0,11
20	$\frac{ x +2}{\cos^2(x^3 + 2x+1)^2}$	-2,90	26,45	0,25
21	$\frac{\sin^2(x+5)}{\sqrt[3]{ x +2} - 1}$	-1,25	67,32	0,3
22	$\frac{e^x \ln x }{\operatorname{ctg}(3x-1)^2}$	-3,78	18,10	0,33
23	$e^{-2x} + \sqrt[7]{2x^4 + x^2 + 1}$	-1,9	15,4	0,45

Продолжение таблицы 4.1

24	$\frac{5}{\operatorname{tg}(2x+3)+1}$	-2,7	10,5	0,75
25	$\frac{(3x-1)^2}{x^5 + \sin(x+2)^3}$	-3,8	17,1	0,65
26	$\frac{x^3 + 2x^2 - 4x}{x^5 \operatorname{ctg}(2x^3)}$	-4,6	29,9	0,3
27	$\frac{\operatorname{tg}^2(x+1)}{x^4 + 2x^3 - x}$	-5,74	19,2	0,4
28	$\frac{\cos^2(x+2)^3}{2\sqrt{x^3} \sin(x^3)}$	-2,25	16,3	0,25
29	$\frac{2x+2}{\operatorname{tg}(2x-1)+1}$	-3,9	18,7	0,23
30	$\frac{\cos(x+2)^2}{e^{-2x} + \sqrt[4]{3x^2+1}}$	-1,8	27,9	0,8

ЛАБОРАТОРНАЯ РАБОТА 5

Понятие одномерного массива. Селективная обработка элементов массива

Цель: использование циклического оператора с параметром для ввода, вывода и обработки элементов массива; изучение методов селективной обработки элементов массива и нахождение минимального и максимального элемента в массиве; вывод элементов массива с помощью визуальных компонентов *Visual Studio 2010*.

5.1 Пример выполнения работы

Создать Windows-приложение, которое предлагает пользователю задать размер линейного массива, заполняет автоматически этот массив случайными целыми числами в диапазоне от -10 до 10, выводит элементы массива, затем по выбору пользователя определяет: сумму четных элементов массива и количество положительных элементов массива.

Ход выполнения

- 1 Войдем в среду *Visual Studio 2010*.
- 2 В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.
- 3 Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, **Lab5**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор** (например N:\CI\2_trim\Lab5).
- 4 Для формы изменим значение свойства **Text** занеся следующие данные – «*Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 5*».
- 5 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.
- 6 В верхней части окна разместим контейнер **GroupBox1** и в свойстве **Text** занесем текст «*Ввод элементов массива*».
- 7 На этом же контейнере **GroupBox1** разместим пять компонентов: **Label1**, **Label2**, **TextBox1**, **TextBox2** и **Button1**. Для первого компонента **Label1** в свойство **Text** занесем текст «*Введите число элементов массива*». Для второго компонента **Label2** в свойство **Text** занесем текст «*Исходный массив*». Для компонента **TextBox2** в свойстве **ReadOnly**

(только чтение) присвоим значение *true*, запрещающее пользователю заносить в компонент какие-либо данные. Для компонента **TextBox1** данные будут вводиться с клавиатуры с помощью компонента текстовый редактор класса **TextBox**. В свойство **Text** этого компонента введем какое-либо значение – значение по умолчанию. Это значение будет показываться при запуске приложения на выполнение. При выполнении приложения его можно будет заменить другим.

8 Выделим компонент **Button1** и в свойство **Text** занесем текст «*Создать массив*». При нажатии на эту кнопку будет автоматически создаваться массив с помощью генератора случайных чисел, элементы которого будут в указанном диапазоне [-10;10] выведены в компоненте **TextBox2**. Получим следующую форму на рис. 5.1.

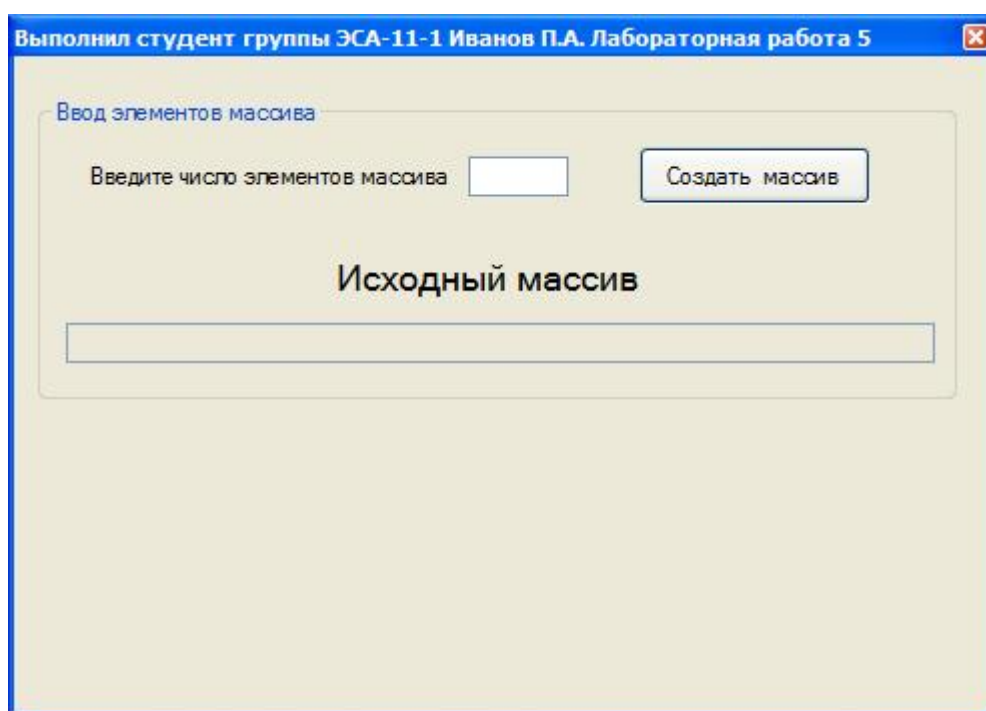


Рисунок 5.1 – Окно формы на этапе создания программы

9 В нижней части окна разместим контейнер **GroupBox2** и в свойстве **Text** занесем текст «*Найти*».

10 На этом же контейнере **GroupBox2** разместим шесть компонентов: **TextBox3**, **TextBox4**, **checkBox1**, **checkBox2**, **Button2** и **Button3**.

11 Выделим компонент **checkBox1** и в свойство **Text** занесем текст «*сумму четных элементов массива*». Аналогично для компонента **checkBox2** в свойство **Text** занесем текст «*количество положительных элементов массива*».

12 Выделим компонент **Button2** и в свойство **Text** занесем текст «**Вычислить**». Аналогично для компонента **Button3** в свойство **Text** занесем текст «**Выход**».

13 Компоненты **TextBox3** и **TextBox4** будут использованы для вывода результатов нахождения задач, поэтому необходимо запретить ввод в них данных пользователем. Для этого свойству **ReadOnly** (только чтение) присвоить значение **true**, запрещающее пользователю заносить в компонент какие-либо данные.

14 В результате проведенных операций должна получиться форма примерно такого вида, как показано на рис. 5.2.

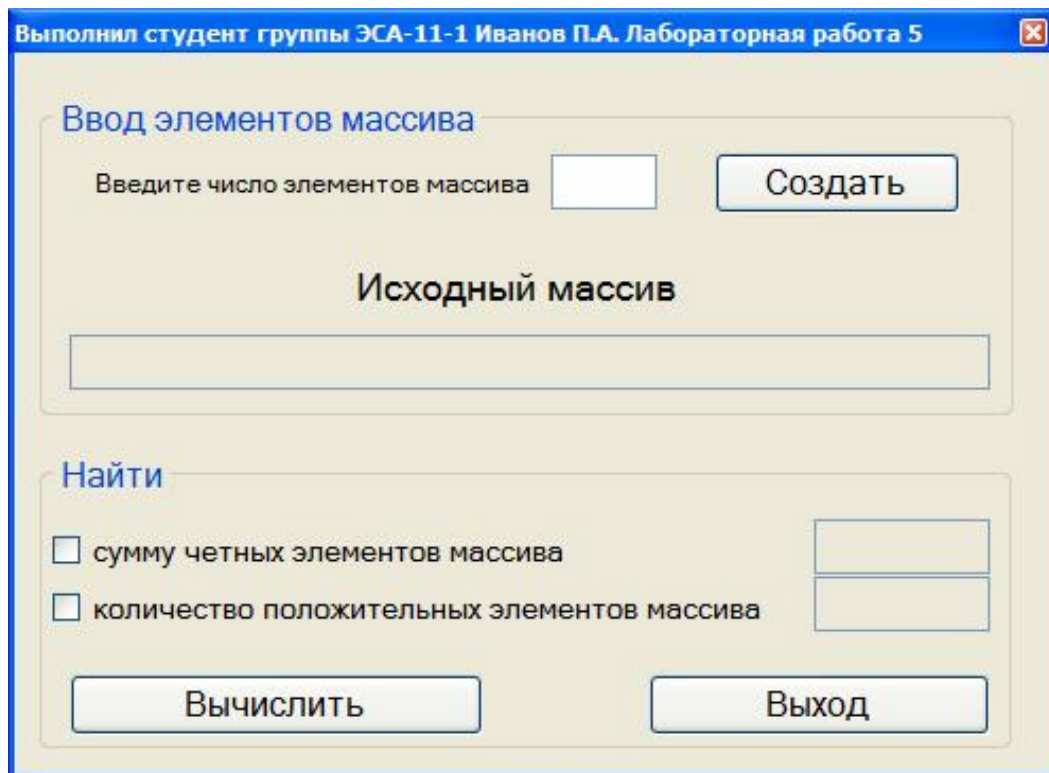


Рисунок 5.2 – Окно формы программы

15 Перейдем к коду программы и после строки

```
#pragma once
```

подключим библиотеки использования математических функций, описание переменных и размера массива, для этого вставим следующие строки:

```
#include <cmath>
#include <stdlib.h>
#define m 15
int i, n, p;
int A[m];
```

16 Создадим событие *Click* для кнопки **Button1** с надписью «*Создать*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button1**. Внесем изменения в код подпрограммы:

```
//Проверка, что не пустой компонент textBox1
    if(textBox1->Text!="")
        {n=Convert::ToInt32(textBox1->Text); }else
{MessageBox::Show( "Заполните пожалуйста данные", "Ошибка ввода
данных", MessageBoxButtons::OK, MessageBoxIcon::Exclamation );}

//очистка компонента textBox2
{textBox2->Text = "";}
//Процесс создания массива и заполнение компонента textBox2 случайными числами из
диапазона [-10;10]
for (i = 0; i < n; ++i)
{
A[i] = rand ( ) % 21-10;
this->textBox2->AppendText (A[i]+" ");
}
}
```

17 Создадим событие *Click* для кнопки **Button2** с надписью «*Вычислить*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button2** и в полученную заготовку подпрограммы вставим код для расчетов:

```
//Обнуляем переменные kol и sum
int kol=0; int sum=0;

    for (i = 0; i < n; ++i)
    {
//Проверяем условие и находим сумму
if (A[i]%2==0){sum=sum+A[i];}

//Проверяем условие и находим количество
if (A[i]>0){kol=kol+1;}
}
//Если компоненты checkBox включены, то выводим результаты заданий в
компоненты textBox
if (checkBox1->Checked==true){this->textBox3->Text=Convert::ToString (sum);}
if (checkBox2->Checked==true){this->textBox4->Text=Convert::ToString (kol);}
```

18 Создадим событие *Click* для кнопки **Button3** с надписью «*Выход*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button3** и в полученную заготовку подпрограммы вставим код выхода из программы (уже использовался в предыдущих лабораторных работах).

19 Запустим программу на выполнение нажав на функциональную кнопку **F5** и получим следующий вид окна рис. 5.3.

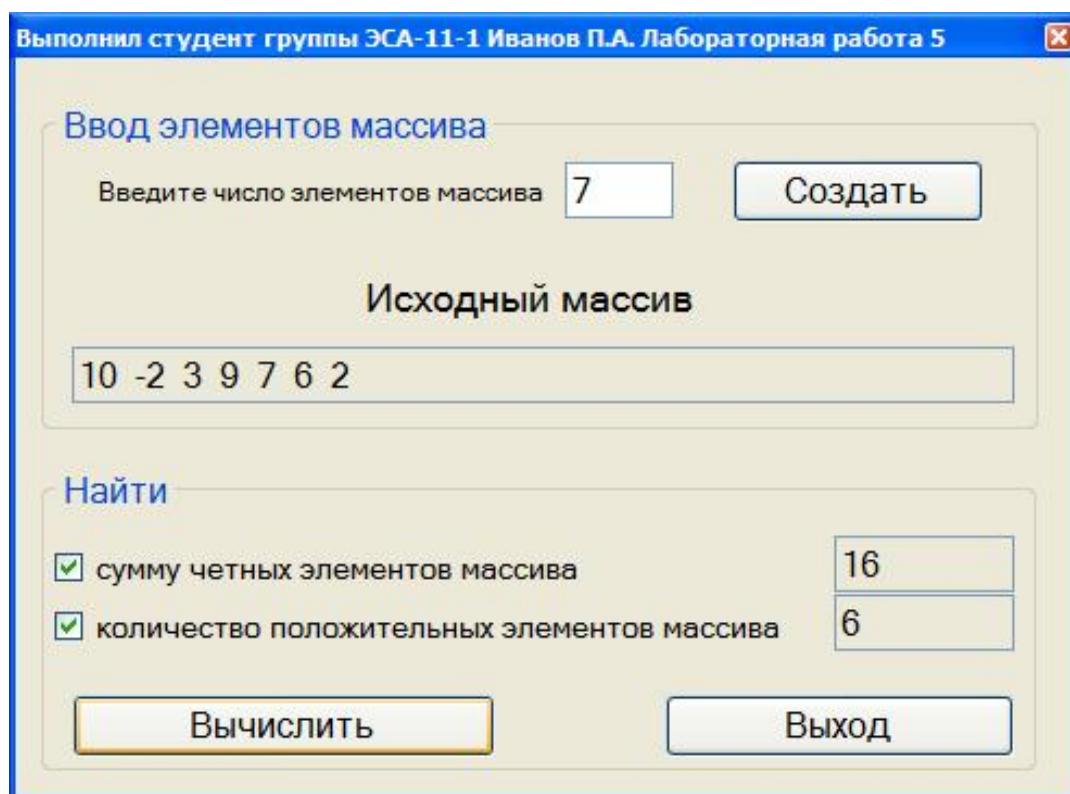


Рисунок 5.3 – Рабочий вид формы приложения

5.2 Рабочее задание

Создать Windows-приложение, которое предлагает пользователю задать размер линейного массива, заполняет автоматически этот массив случайными целыми числами в диапазоне от -50 до 70 , выводит элементы этого массива, затем по выбору пользователя определяет: соответственно для каждого варианта задание **а), б), в)**.

Данные взять с таблицы 5.1.

Таблица 5.1 – Индивидуальные задания

Вариант	задание	Условие задания
1	<i>а</i>	Найти количество отрицательных элементов
	<i>б</i>	Найти сумму отрицательных элементов
	<i>в</i>	Найти минимальный элемент кратный пяти
2	<i>а</i>	Найти количество четных элементов
	<i>б</i>	Найти сумму элементов кратных 3
	<i>в</i>	Найти разность максимального и минимального элементов массива

Продолжение таблицы 5.1

3	<i>a</i>	Найти среднее арифметическое элементов массива
	<i>б</i>	Найти сумму наибольшего и наименьшего элементов массива
	<i>в</i>	Найти максимальный по модулю элемент массива
4	<i>a</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>в</i>	Найти произведение модулей наибольшего отрицательного и наименьшего четного элементов массива
5	<i>a</i>	Найти количество элементов кратных 5
	<i>б</i>	Найти сумму четных элементов массива стоящих на нечетных местах
	<i>в</i>	Найти сумму второго и наибольшего положительного элементов массива
6	<i>a</i>	Найти среднее геометрическое четных элементов массива
	<i>б</i>	Найти номер наибольшего по модулю элемента массива
	<i>в</i>	Найти максимальный четный элемент массива
7	<i>a</i>	Вычислить среднее арифметическое максимального и минимального элементов массива
	<i>б</i>	Найти минимальный по модулю элемент массива
	<i>в</i>	Найти сумму элементов из интервала [0;10]
8	<i>a</i>	Вычислить среднее геометрическое номеров максимального и минимального элементов массива
	<i>б</i>	Найти разность суммы положительных и произведения отрицательных чисел массива
	<i>в</i>	Найти количество положительных элементов
9	<i>a</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму элементов массива, у которых индекс кратен 3
	<i>в</i>	Найти произведение модулей наибольшего и наименьшего элементов массива
10	<i>a</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму второго и наибольшего отрицательного элементов массива
	<i>в</i>	Найти разность максимального и минимального элементов массива
11	<i>a</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму элементов в диапазоне [-10;20]
	<i>в</i>	Найти максимальный по модулю элемент массива
12	<i>a</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>б</i>	Найти сумму четных элементов массива из диапазона [-20;30]
	<i>в</i>	Найти минимальный по модулю элемент массива
13	<i>a</i>	Максимальный по модулю элемент
	<i>б</i>	Найти среднее арифметическое элементов массива
	<i>в</i>	Найти сумму отрицательных элементов

Продолжение таблицы 5.1

14	<i>a</i>	Найти количество элементов кратных 4
	<i>б</i>	Найти сумму отрицательных элементов
	<i>в</i>	Найти сумму наибольшего и наименьшего элементов массива
15	<i>a</i>	Найти разность максимального и минимального положительных элементов
	<i>б</i>	Найти сумму нечетных элементов
	<i>в</i>	Найти минимальный элемент из диапазона [-20;30]
16	<i>a</i>	Найти количество положительных элементов
	<i>б</i>	Найти сумму элементов больших 3
	<i>в</i>	Найти максимальный элемент массива
17	<i>a</i>	Найти количество отрицательных элементов
	<i>б</i>	Найти сумму отрицательных элементов
	<i>в</i>	Найти минимальный элемент кратный пяти
18	<i>a</i>	Найти количество четных элементов
	<i>б</i>	Найти сумму элементов кратных 3
	<i>в</i>	Найти разность максимального и минимального элементов массива
19	<i>a</i>	Найти среднее арифметическое элементов массива
	<i>б</i>	Найти сумму наибольшего и наименьшего элементов массива
	<i>в</i>	Найти максимальный по модулю элемент массива
20	<i>a</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>в</i>	Найти произведение модулей наибольшего отрицательного и наименьшего четного элементов массива
21	<i>a</i>	Найти количество элементов кратных 5
	<i>б</i>	Найти сумму четных элементов массива стоящих на нечетных местах
	<i>в</i>	Найти сумму второго и наибольшего положительного элементов массива
22	<i>a</i>	Найти среднее геометрическое четных элементов массива
	<i>б</i>	Найти номер наибольшего по модулю элемента массива
	<i>в</i>	Найти максимальный четный элемент массива
23	<i>a</i>	Вычислить среднее арифметическое максимального и минимального элементов массива
	<i>б</i>	Найти минимальный по модулю элемент массива
	<i>в</i>	Найти сумму элементов из интервала [0;10]
24	<i>a</i>	Вычислить среднее геометрическое номеров максимального и минимального элементов массива
	<i>б</i>	Найти разность суммы положительных и произведения отрицательных чисел массива
	<i>в</i>	Найти количество положительных элементов
25	<i>a</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму элементов массива, у которых индекс кратен 3
	<i>в</i>	Найти произведение модулей наибольшего и наименьшего элементов массива

Продолжение таблицы 5.1

26	<i>a</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму второго и наибольшего положительного элементов массива
	<i>в</i>	Найти разность максимального и минимального элементов массива
27	<i>a</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму элементов в диапазоне [-10;20]
	<i>в</i>	Найти максимальный по модулю элемент массива
28	<i>a</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>б</i>	Найти сумму четных элементов массива из диапазона [-20;30]
	<i>в</i>	Найти минимальный по модулю элемент массива
29	<i>a</i>	Найти количество элементов кратных 5
	<i>б</i>	Найти сумму четных элементов массива стоящих на нечетных местах
	<i>в</i>	Найти максимальный четный элемент массива
30	<i>a</i>	Найти количество положительных элементов
	<i>б</i>	Найти сумму элементов больших 3
	<i>в</i>	Найти максимальный элемент массива

ЛАБОРАТОРНАЯ РАБОТА 6

Понятие матрицы. Селективная обработка элементов строк, столбцов и диагоналей матрицы

Цель: изучение использования вложенных циклических операторов для ввода, вывода и обработки элементов матрицы; обработка элементов столбцов, строк и диагоналей матрицы. Использование визуального компонента `DataGridView` для работы с матрицами у *Visual Studio 2010*.

6.1 Пример выполнения работы

Создать Windows-приложение, которое предлагает пользователю ввести количество строк и столбцов для создания таблицы, в которую будут заноситься элементы матрицы, затем по выбору пользователя определяет: произведение отрицательных элементов матрицы и количество четных элементов матрицы.

Ход выполнения

1 Войдем в среду *Visual Studio 2010*.

2 В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту CLR и на центральной панели выбрать *Приложение Windows Form*.

3 Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, **Lab6**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) **Обзор** (например `N:\CI\2_trim\Lab6`).

4 Для формы изменим значение свойства **Text** занеся следующие данные – «*Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 6*».

5 Свойству **FormBorderStyle** формы (стиль рамки окна) присвоить значение *FixedToolWindow*. Это значение определяет окно как диалоговое, его размеры на этапе прогона приложения (в процессе работы приложения) не могут быть изменены.

6 В верхней части окна разместим контейнер **GroupBox1** и в свойстве **Text** занеся текст «*Ввод данных матрицы*».

7 На этом же контейнере **GroupBox1** разместим семь компонентов: **Label1**, **Label2**, **Label3**, **Label4**, **TextBox1**, **TextBox2** и **DataGridView1**. Для первого компонента **Label1** в свойство **Text** занеся текст «*Задайте размер матрицы*». Для второго компонента **Label2** в свойство **Text** занеся текст «*N-строк*». Для третьего компонента **Label3** в свойство **Text** занеся текст «*M-столбцов*». Для четвертого компонента **Label4** в свойство **Text** занеся текст «*Матрица $A[n,m]$* ».

8 Выделим компонент **DataGridView1** и для свойств **RowHeadersVisible** (отображение заголовка строк) и **ColumnHeadersVisible** (отображение заголовка столбцов) выберем параметр *False*.

9 Ниже контейнера **GroupBox1** разместим контейнер **GroupBox2** и в свойстве **Text** занесем текст «*Найти*».

10 Выделим компонент **DataGridView2** и на него разместим два компонента **checkBox1** и **checkBox2**.

11 Выделим первый компонент **checkBox1** и в свойство **Text** занесем текст «*произведение отрицательных элементов матрицы*». Аналогично для компонента **checkBox2** в свойство **Text** занесем текст «*количество четных элементов матрицы*». Напротив этих компонентов **checkBox** разместим два компонента **TextBox3** и **TextBox4**.

12 Ниже компонента **GroupBox2** разместим три компонента **Button1**, **Button2** и **Button3**. На первом компоненте **Button1** в свойство **Text** занесем текст «*Создать таблицу*». Для второго компонента **Button2** в свойство **Text** занесем текст «*Перенести данные из таблицы в массив и решить задание*». Для третьего компонента **Button3** в свойство **Text** занесем текст «*Выход*».

13 В результате проведенных операций должна получиться форма примерно такого вида, как показано на рис. 6.2.

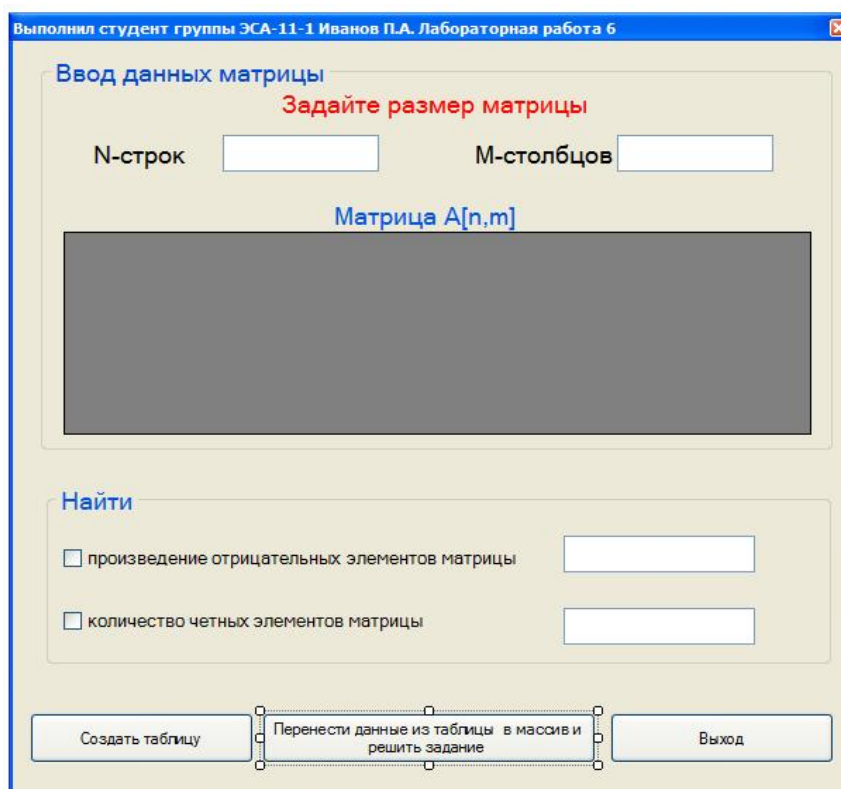


Рисунок 6.2 – Окно формы программы

14 Перейдем к коду программы и после строки

```
#pragma once
```

подключим библиотеки использования математических функций, описание переменных и размера массива, для этого вставим следующие строки:

```
#include<math.h>
int m, n, kol, p;
int A[50][50];
```

15 Создадим событие *Click* для кнопки **Button1** с надписью «*Создать таблицу*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button1**. Внесем изменения в код подпрограммы:

```
//Проверка, что не пустые компоненты textBox1 и textBox2
    if ((textBox1->Text!="") && (textBox2->Text!=""))
    { m = Convert::ToInt32(textBox1->Text);
      n = Convert::ToInt32(textBox2->Text);

//Чистка столбцов компонента DataGridView, если они не пусты
      int k = 0;
      k = dataGridView1->ColumnCount;
      if (k != 0)
          for (int i = 0; i < k; i++)
              dataGridView1->Columns->Clear();

//Заполнение компонента DataGridView столбцами
      dataGridView1->ColumnCount = n;

//Заполнение компонента DataGridView строками
      dataGridView1->Rows->Add(m);
      else
    {MessageBox::Show( "Заполните пожалуйста данные", "Ошибка ввода данных",
      MessageBoxButtons::OK, MessageBoxIcon::Exclamation );}
```

16 Создадим событие *Click* для кнопки **Button2** с надписью «*Перенести данные из таблицы в массив и решить задание*». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button2** и в полученную заготовку подпрограммы вставим код для расчетов:

```
//переменную kol обнуляем, а переменную p присваиваем единице
kol=0;p=1;

//Производим считывание из ячеек таблицы и вносим данные в массив
for (int i = 0; i < m; i++)
    for (int j = 0; j < n; j++)
    {
A[i][j] = Convert::ToSingle(this->dataGridView1->Rows[i]->Cells[j]->Value);
```

```

if (A[i][j]<0) {p=p*A[i][j];}
if (A[i][j]%2==0) {kol++;}
}
//Вывод данных нахождения произведения отрицательных элементов матрицы
if ((checkBox1->Checked==true) && (p!=1)) {this->textBox3->Text=Convert::ToString (p);}
else
if (checkBox1->Checked==true) {this->textBox3->Text=Convert::ToString ("нет элементов");}

//Вывод данных нахождения количество четных элементов матрицы
if ((checkBox2->Checked==true) && (kol!=0)) {this->textBox4->Text=Convert::ToString (kol);}
else
if (checkBox2->Checked==true) {this->textBox4->Text=Convert::ToString ("нет элементов");}
return;

```

17 Создадим событие **Click** для кнопки **Button3** с надписью «**Выход**». Для этого сделаем двойной щелчок левой кнопкой мыши по компоненту **Button3** и в полученную заготовку подпрограммы вставим код выхода из программы (уже использовался в предыдущих лабораторных работах).

18 Запустим программу на выполнение нажав на функциональную кнопку **F5** и получим следующий вид окна рис. 6.3.

Выполнил студент группы ЭСА-11-1 Иванов П.А. Лабораторная работа 6

Ввод данных матрицы

Задайте размер матрицы

N-строк M-столбцов

Матрица A[n,m]

3	4	7
6	2	8
-5	-1	-3

Найти

произведение отрицательных элементов матрицы

количество четных элементов матрицы

Создать таблицу Перенести данные из таблицы в массив и решить задание Выход

Рисунок 6.3 – Рабочий вид формы приложения

6.3 Рабочее задание

Создать Windows-приложение, которое предлагает пользователю задать количество строк и столбцов матрицы и при нажатии на кнопку автоматически создается таблица и в ее ячейки автоматически вводятся элементы матрицы с помощью генератора случайных чисел. Затем по выбору пользователя определяет: соответственно для каждого варианта задание а), б) и в).

Данные взять с таблицы 6.5.

Таблица 6.5 – Индивидуальные задания

Вариант	Задание	Условие задания
1	<i>а</i>	Найти количество элементов, больших заданного числа C (ввод числа C сделать с клавиатуры).
	<i>б</i>	Найти сумму элементов расположенных по периметру.
	<i>в</i>	В матрице $A(4;4)$ найти сумму произведения четных чисел 1-ой строки и произведения положительных чисел 3-го столбца.
2	<i>а</i>	Найти минимальный по модулю элемент и номер строки и столбца, где он находится.
	<i>б</i>	Найти сумму наибольшего положительного и наименьшего четного.
	<i>в</i>	Подсчитать количество кратных 3 чисел 2-ой строки и количество четных чисел 1-го столбца матрицы $A(6;6)$.
3	<i>а</i>	Найти произведение элементов, меньших заданного числа T (ввод числа T сделать с клавиатуры).
	<i>б</i>	Найти произведение элементов, находящихся на главной диагонали.
	<i>в</i>	Найти разность произведения нечетных чисел 3-ей строки и произведения отрицательных чисел 1-го столбца матрицы $A(4;4)$.
4	<i>а</i>	Найти сумму положительных кратных 5 элементов.
	<i>б</i>	Найти произведение элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(8;8)$ найти разность произведения нечетных чисел 3-ей строки и суммы положительных чисел 6-го столбца.
5	<i>а</i>	Найти произведение отрицательных четных элементов.
	<i>б</i>	Найти сумму элементов, находящихся на главной диагонали.
	<i>в</i>	В матрице $A(5;5)$ найти сумму количества четных чисел 2-ой строки и количества отрицательных чисел 4-го столбца.
6	<i>а</i>	Найти количество положительных нечетных элементов.
	<i>б</i>	Найти сумму элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(3;3)$ найти произведение количества нечетных чисел 1-ой строки и количества положительных чисел 3-го столбца.

Продолжение таблицы 6.5

7	<i>a</i>	Найти количество элементов, меньших числа 5.
	<i>б</i>	Найти количество отрицательных элементов, находящихся на главной диагонали.
	<i>в</i>	Найти максимальный элемент 3-го столбца и сумму нечетных элементов 1-ой строки матрицы $A(5;5)$.
8	<i>a</i>	Найти произведение положительных кратных 3 элементов.
	<i>б</i>	Найти количество отрицательных элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(6;6)$ найти произведение суммы четных чисел в 3-ей строке и суммы отрицательных чисел 1-го столбца.
9	<i>a</i>	Найти сумму отрицательных не кратных 5 элементов.
	<i>б</i>	Найти количество четных элементов, находящихся на главной диагонали.
	<i>в</i>	Найти произведение суммы положительных чисел в 4-ом столбце на количество четных чисел 2-ой строки матрицы $A(6;6)$.
10	<i>a</i>	Найти квадрат максимального элемента и номер строки и столбца, где он находится.
	<i>б</i>	Найти количество четных элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(7;7)$ найти разность количества положительных чисел 1-ой строки и количества четных чисел 3-го столбца.
11	<i>a</i>	Найти сумму четных элементов из интервала $[-10;10]$ матрицы $A(4;4)$.
	<i>б</i>	Найти количество нечетных элементов, находящихся на главной диагонали.
	<i>в</i>	Найти произведение количества четных элементов 3 строки на сумму нечетных элементов 2 столбца матрицы $A(4;4)$.
12	<i>a</i>	Найти количество кратных 3 элементов из интервала $[-6;8]$ матрицы $A(5;5)$.
	<i>б</i>	Найти количество нечетных элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(5;5)$ найти произведение количества четных чисел 2-ой строки и количества отрицательных чисел 4-го столбца.
13	<i>a</i>	Найти произведение отрицательных нечетных элементов матрицы $A(5;5)$.
	<i>б</i>	Найти сумму четных элементов, находящихся на главной диагонали.
	<i>в</i>	В матрице $A(5;5)$ найти произведение количества нечетных чисел 3-го столбца и количества отрицательных чисел 3 строки.
14	<i>a</i>	Найти количество положительных элементов из интервала $[-5;6]$ матрицы $A(6;6)$.
	<i>б</i>	Найти сумму четных элементов, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(6;6)$ найти произведение суммы кратных 3 чисел 2-ей строки и суммы отрицательных чисел 2-го столбца.

Продолжение таблицы 6.5

15	<i>a</i>	Найти максимальный по модулю элемент и номер строки и столбца, где он находится.
	<i>б</i>	Найти сумму элементов кратных 3, находящихся на главной диагонали.
	<i>в</i>	Найти произведение количества четных чисел в 2-ом столбце на количество нечетных чисел 2-ой строки матрицы $A(4;4)$.
16	<i>a</i>	Найти сумму положительных кратных 3 элементов.
	<i>б</i>	Найти сумму элементов кратных 3, находящихся на побочной диагонали.
	<i>в</i>	Найти среднее геометрическое нечетных элементов 2-го столбца и количество кратных 5 элементов 3-ей строки матрицы $A(5;5)$.
17	<i>a</i>	Найти количество отрицательных четных элементов.
	<i>б</i>	Найти сумму отрицательных четных элементов, находящихся на главной диагонали.
	<i>в</i>	Найти сумму нечетных элементов 3-го столбца и произведение отрицательных кратных 3 элементов 2-ой строки матрицы $A(6;6)$.
18	<i>a</i>	Найти произведение положительных четных элементов.
	<i>б</i>	Найти сумму отрицательных четных элементов, находящихся на побочной диагонали.
	<i>в</i>	Найти произведение суммы кратных 3 чисел в 4-ом столбце на количество нечетных чисел 2-ой строки матрицы $A(4;4)$.
19	<i>a</i>	Найти сумму отрицательных четных элементов.
	<i>б</i>	Найти произведение элементов не кратных 3, которые находятся на главной диагонали.
	<i>в</i>	В матрице $A(7;7)$ найти разность количества нечетных чисел 1-ой строки и количества четных чисел 4-го столбца.
20	<i>a</i>	Найти максимальный элемент, номер строки и столбца, в котором он находится.
	<i>б</i>	Найти разницу суммы четных и количества отрицательных элементов матрицы, находящихся на главной диагонали.
	<i>в</i>	Найти сумму нечетных элементов 2-го столбца и произведение отрицательных кратных 3 элементов 4-ой строки матрицы $A(4;4)$.
21	<i>a</i>	Найти минимальный элемент, номер строки и столбца, в котором он находится.
	<i>б</i>	Найти произведение суммы четных на количество положительных элементов матрицы, находящихся на главной диагонали.
	<i>в</i>	Найти произведение отрицательных четных элементов 2-ой строки и количество не кратных 5 элементов 2-го столбца матрицы $A(5;5)$.
22	<i>a</i>	Найти количество положительных кратных 5 элементов.
	<i>б</i>	Найти произведение суммы четных на количество положительных элементов матрицы, находящихся на побочной диагонали.
	<i>в</i>	Подсчитать количество положительных кратных 3 элементов 1-ой строки и количество нечетных элементов 2-го столбца матрицы $A(6;6)$.

Продолжение таблицы 6.5

23	<i>a</i>	Найти произведение отрицательных нечетных элементов.
	<i>б</i>	Найти модуль суммы элементов кратных 5, находящихся на побочной диагонали.
	<i>в</i>	Найти разность произведения нечетных чисел 3-ей строки и произведения отрицательных чисел 1-го столбца матрицы $A(4;4)$.
24	<i>a</i>	Найти квадрат минимального элемента и номер строки и столбца, где он находится.
	<i>б</i>	Найти разницу суммы четных и количества отрицательных элементов матрицы, находящихся на побочной диагонали.
	<i>в</i>	В матрице $A(7;7)$ найти разность произведения нечетных чисел 2-ой строки и суммы положительных чисел 6-го столбца.
25	<i>a</i>	Найти произведение положительных не кратных 5 элементов.
	<i>б</i>	Найти модуль суммы элементов кратных 5, находящихся на главной диагонали.
	<i>в</i>	В матрице $A(5;5)$ найти сумму количества четных чисел 3-ей строки и количества отрицательных чисел 4-го столбца.
26	<i>a</i>	Найти максимальный по модулю элемент и номер строки и столбца, где он находится.
	<i>б</i>	Найти разность максимального и минимального элементов, находящихся на главной диагонали.
	<i>в</i>	В матрице $A(6;6)$ найти произведение количества нечетных чисел 2-го столбца и количества положительных чисел 3 строки.
27	<i>a</i>	Найти количество отрицательных не кратных 3 элементов.
	<i>б</i>	Найти разность максимального и минимального элементов, находящихся на побочной диагонали.
	<i>в</i>	Найти максимальный элемент 2-ой строки и количество четных элементов 5-го столбца матрицы $A(5;5)$.
28	<i>a</i>	Найти произведение положительных нечетных элементов.
	<i>б</i>	Найти модуль разности суммы четных и произведения нечетных элементов матрицы, находящихся на главной диагонали.
	<i>в</i>	В матрице $A(6;6)$ найти произведение суммы четных чисел 3-ей строки и суммы отрицательных чисел 1-го столбца.
29	<i>a</i>	Найти сумму отрицательных нечетных элементов.
	<i>б</i>	Найти модуль разности суммы четных и произведения нечетных элементов матрицы, находящихся на побочной диагонали.
	<i>в</i>	Найти произведение суммы положительных чисел 1-й строки на сумму четных чисел 2-го столбца матрицы $A(5;5)$.
30	<i>a</i>	Найти произведение отрицательных четных элементов.
	<i>б</i>	Найти произведение максимального элемента главной диагонали на минимальный элемент побочной диагонали.
	<i>в</i>	В матрице $A(7;7)$ найти разность количества отрицательных чисел 2-ой строки и количества нечетных чисел 3-го столбца.