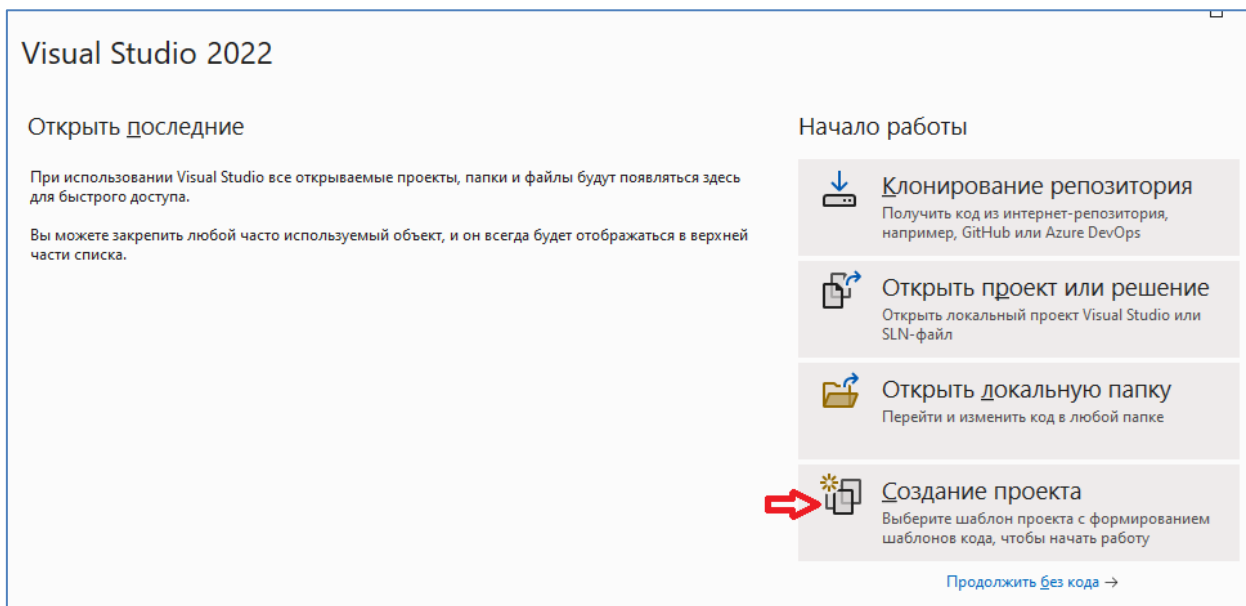


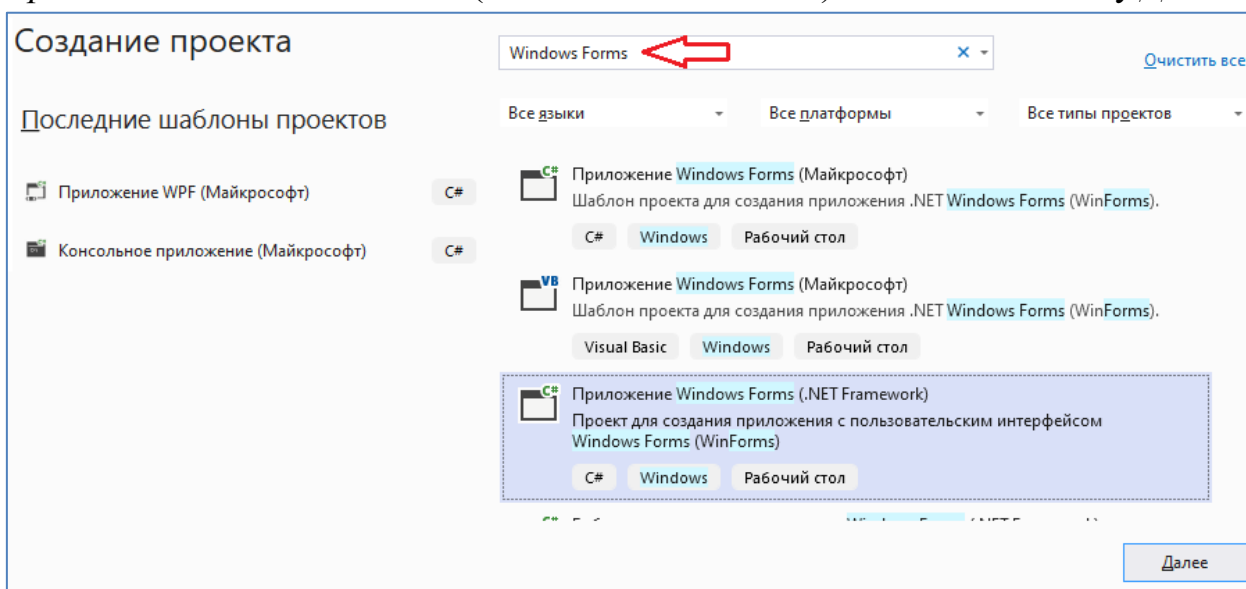
1. Создание проекта

Для создания проекта оконного приложения выполните следующие действия.

1. Откройте Visual Studio
2. В начальном окне выберите *Создание проекта*



3. В следующем окне выбора шаблона проекта в окне поиска введите *Windows Forms*, прокрутите список шаблонов, найдите шаблон *Приложение Windows Forms* (с использованием C#) и нажмите кнопку *Далее*.



4. В окне *Настроить новый проект* задайте *Имя проекта*, выберите местоположение проекта и нажмите кнопку *Создать*

Настроить новый проект

Приложение Windows Forms (.NET Framework) C# Windows Рабочий стол

Имя проекта

Расположение


Решение

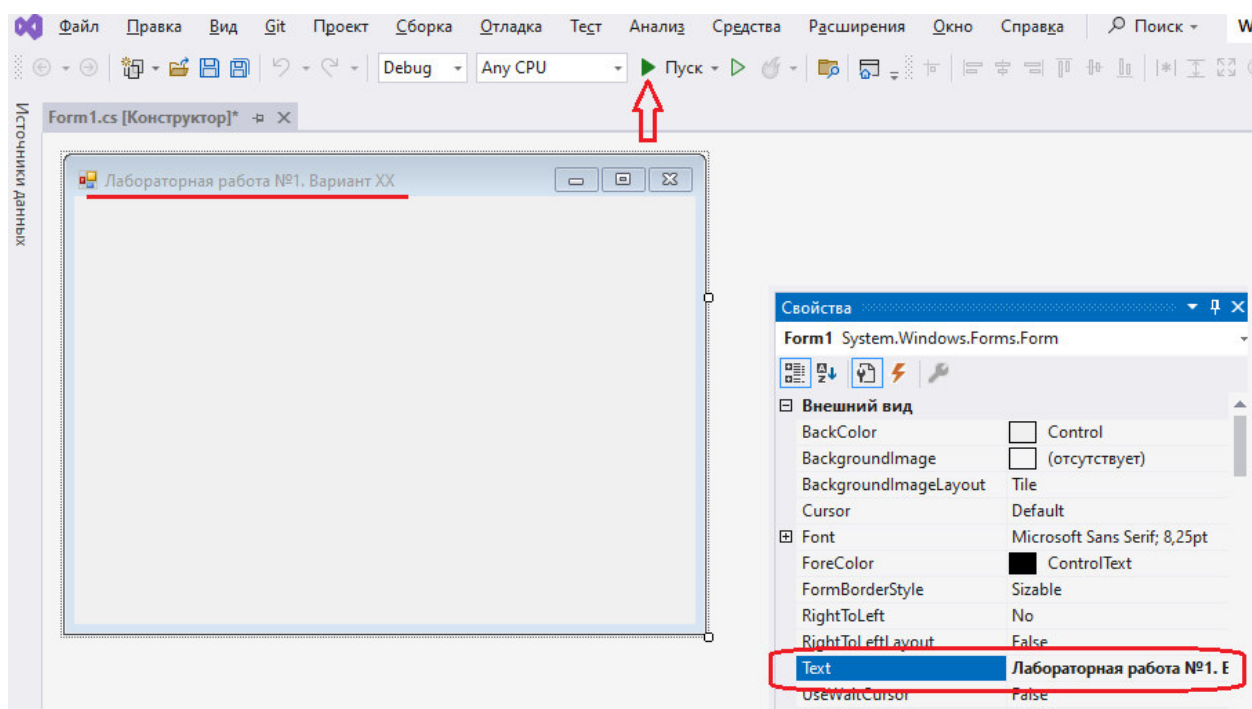
Имя решения ⓘ

☒ Поместить решение и проект в одном каталоге

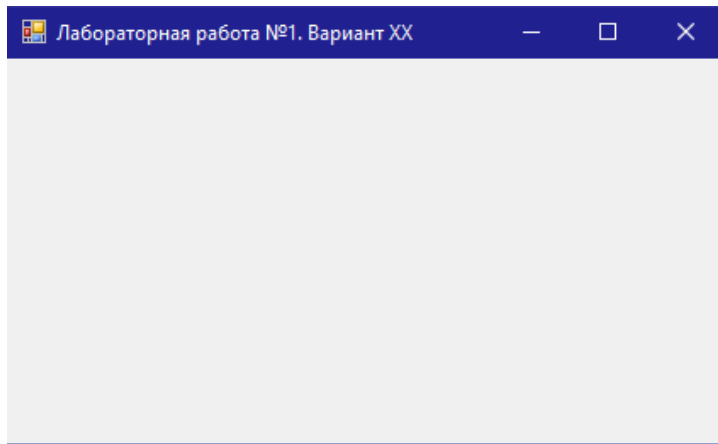
Платформа

Проект будет создан в "D:\MyProject\C#\LR1_WinForm\WindowsFormsApp\"

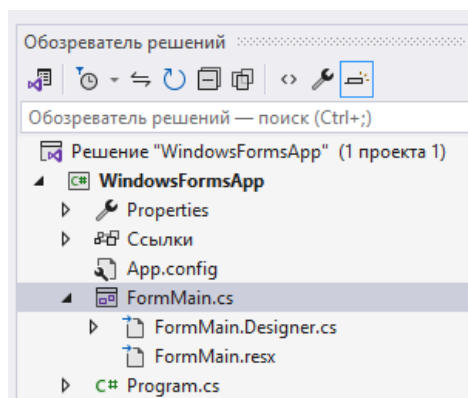
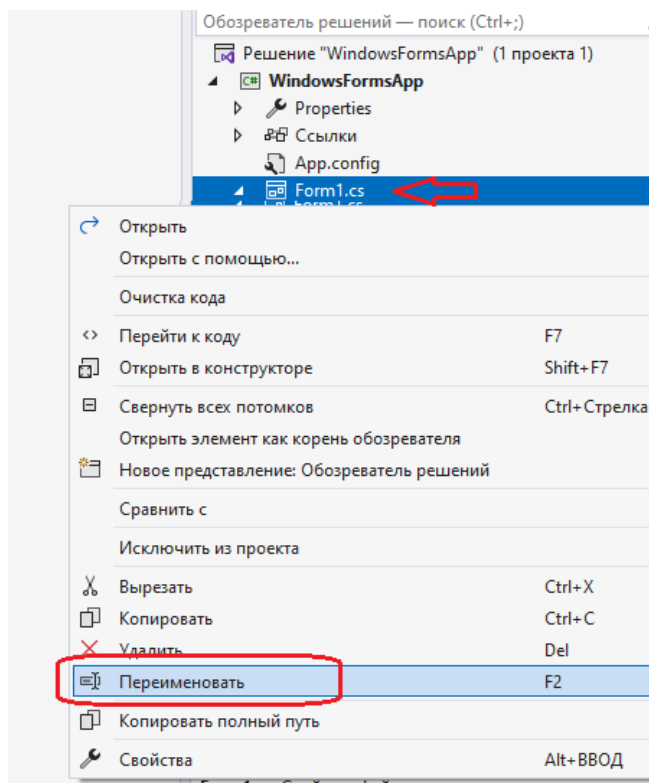
Откроется окно проекта, в котором будет пустое окно. Изменим заголовок окна в панели Свойства (свойство Text) и протестируем приложение, нажав кнопку Выполнить  Пуск



Будет выведено пустое окно



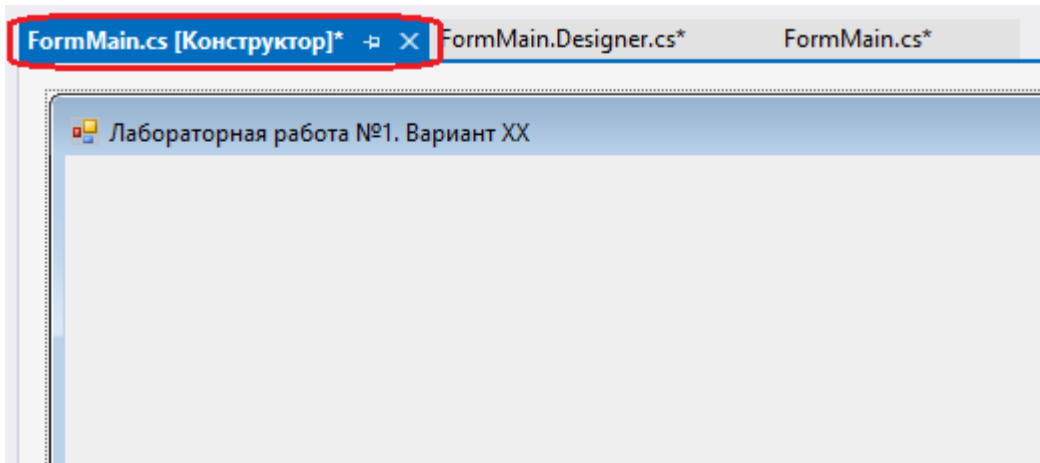
В обозревателе решений переименуйте форму проекта Form1 на FormMain



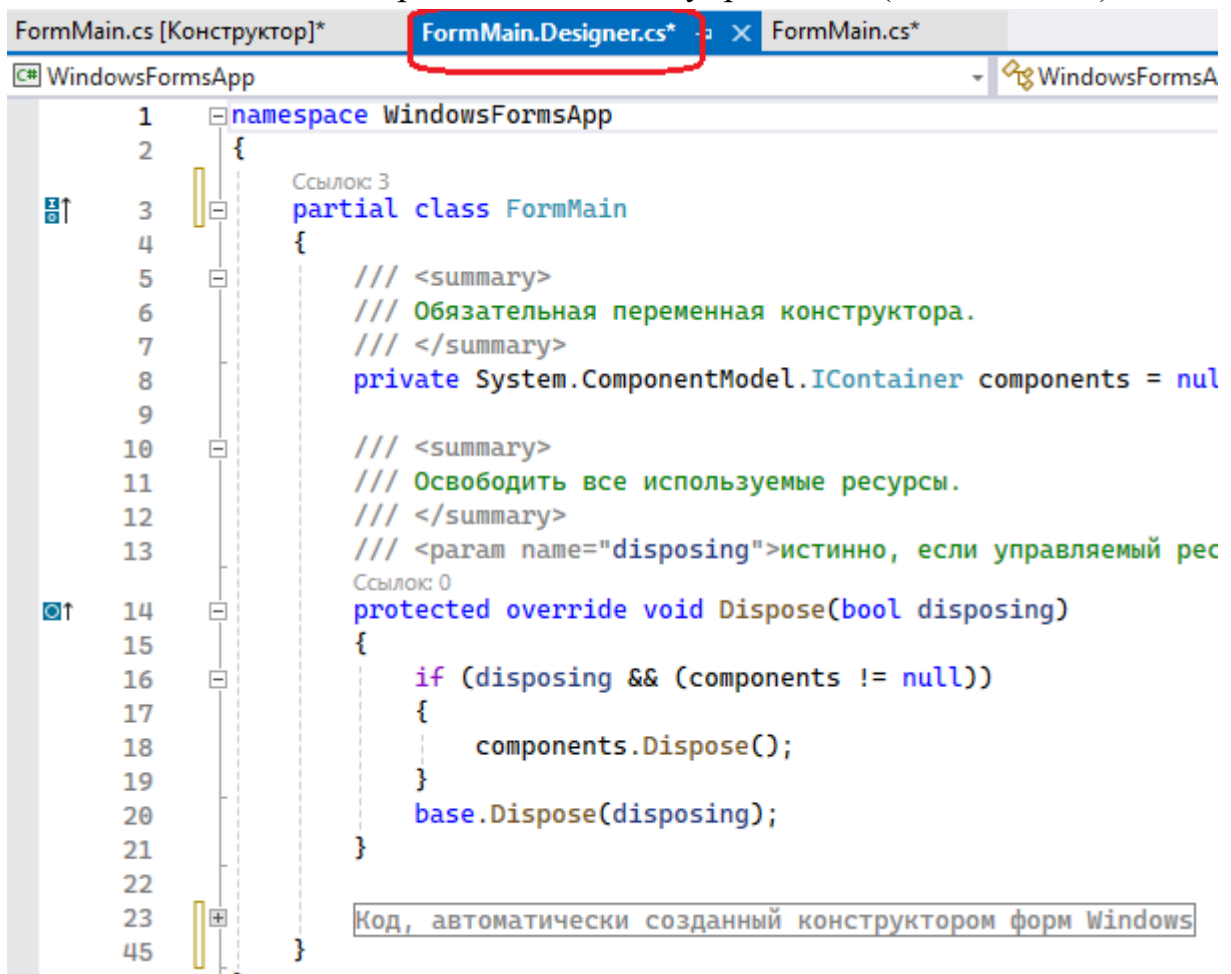
2. Основные файлы проекта (для каждого окна или формы приложения).

Проект включает несколько файлов:

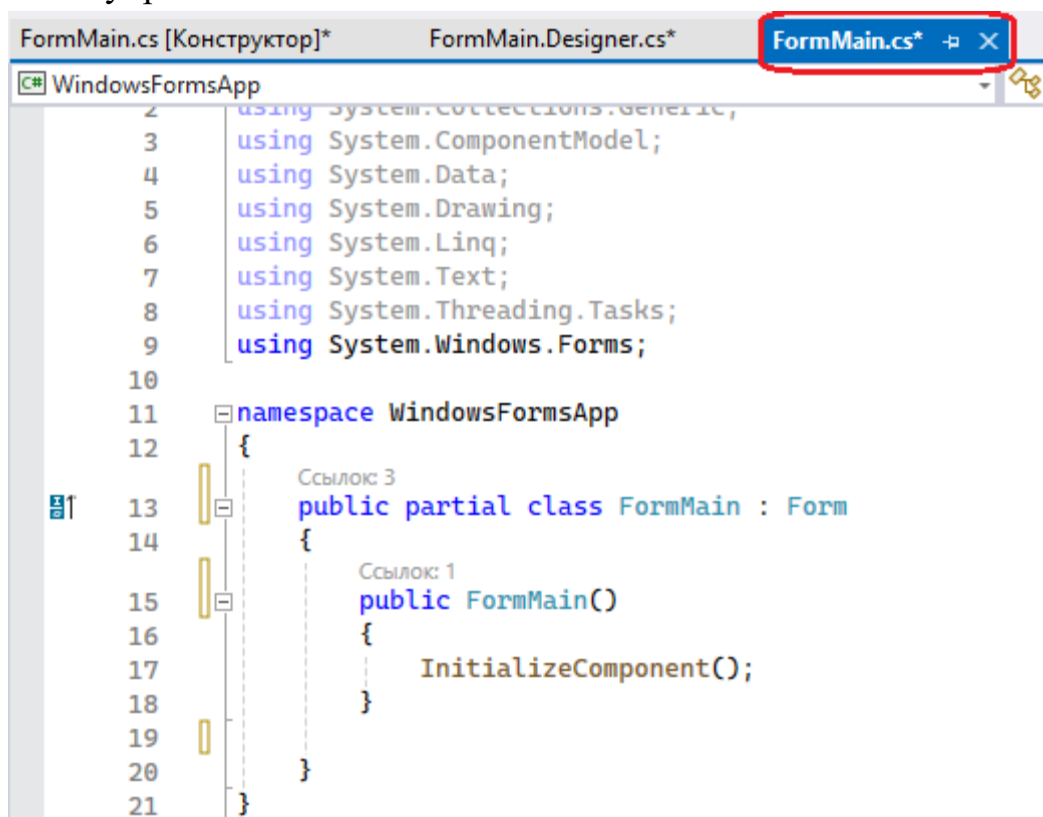
- 1) Окно конструктора, в котором будем добавлять основные элементы (компоненты) формы



- 2) Окно программного кода, формируемого конструктором формы при добавлении и настройке элементов управления (компонентов)



- 3) Файл программного кода формы, в который будем добавлять операторы программного кода (включая обработчики элементов управления)



3. Конструирование пользовательского интерфейса

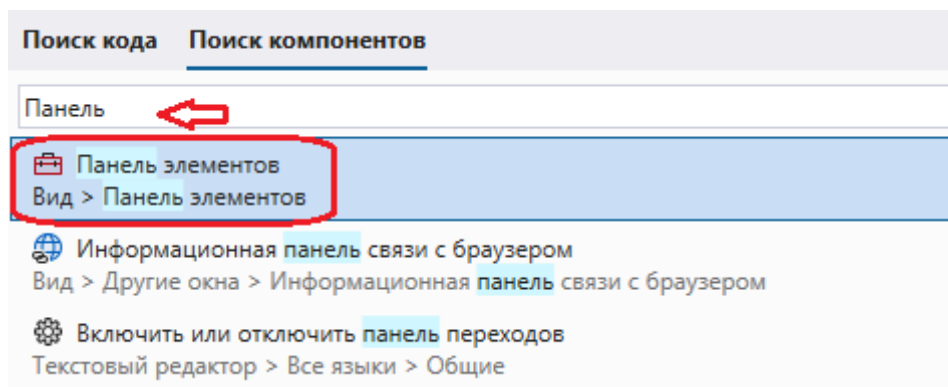
Все элементы управления (компоненты) будем добавлять в окне Конструктора. Настройки свойств добавляемых элементов (положение, размеры, наполнение, оформление) будем выполнять на панели Свойства.

Наполним окно компонентами для ввода данных и вывода результатов.

Если конструктор не открыт, выберите *MainWindow.xaml* и нажмите клавишу **SHIFT+F7**, чтобы открыть конструктор.

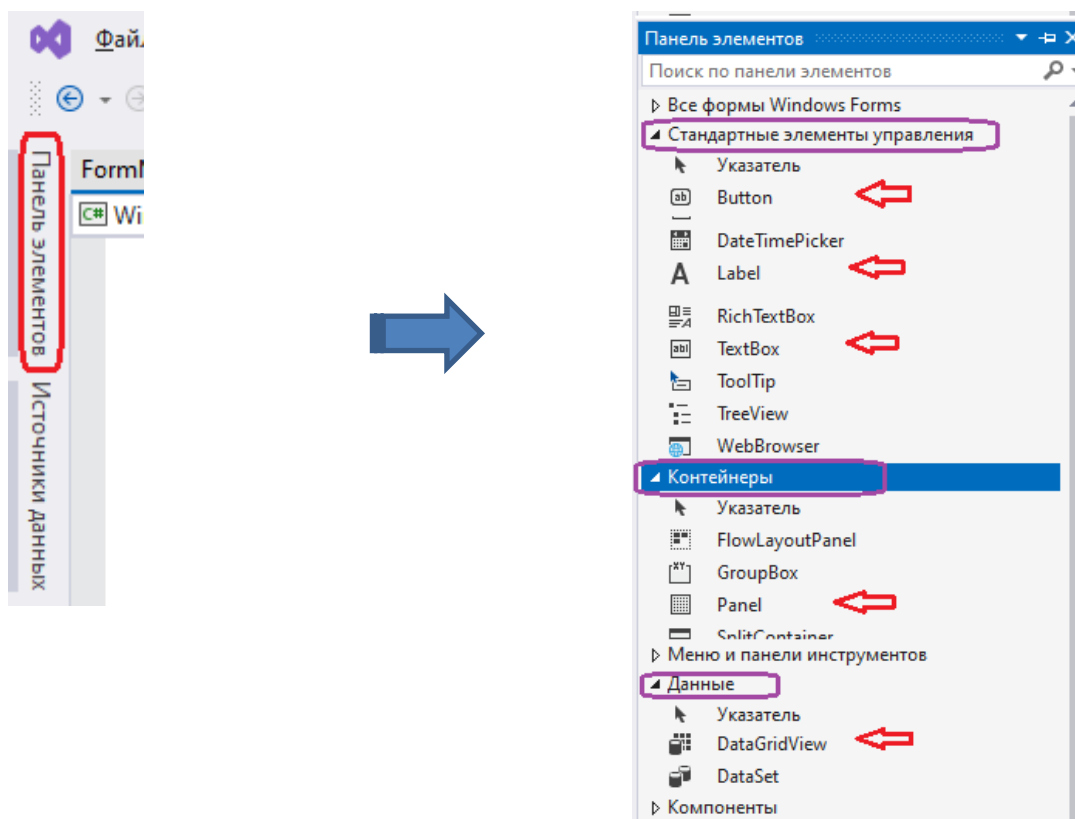
Добавление элементов управления

Нажмите клавиши **CTRL+Q** для активации поля поиска и введите **Панель элементов**. Выберите в списке результатов **Вид > Панель элементов**.



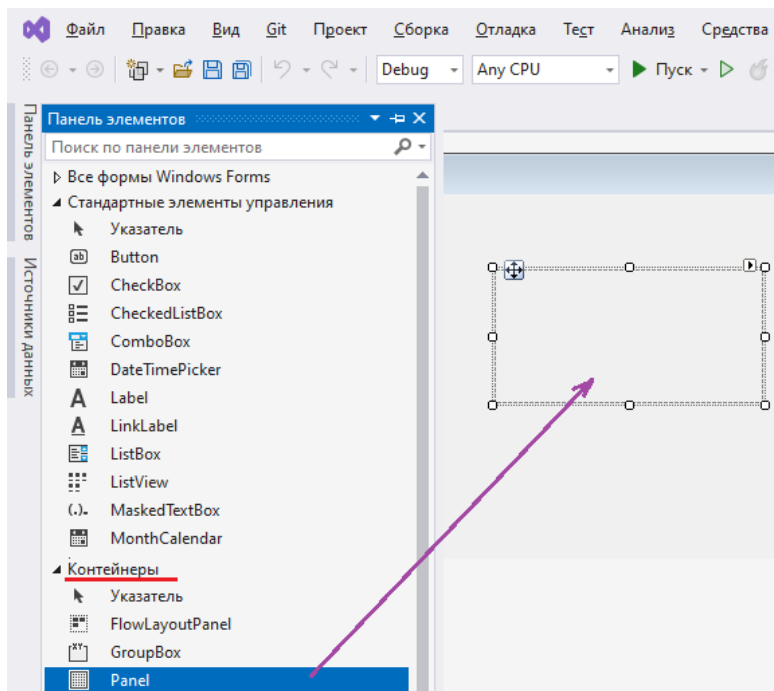
В левой части окна проявится ярлык Панели элементов. После нажатия на ярлык Панель раскроется, на которой находятся основные элементы, сгруппированные в узлах.

Раскройте узлы Стандартные элементы управления, Контейнеры, Данные, на которых находятся компоненты для пользовательского интерфейса приложения.



Добавление элементов выполняется перетаскиванием его из панели на форму в окне *Конструктора*.

Например, выберем в разделе *Контейнеры* элемент *Panel* и перетащим его на форму

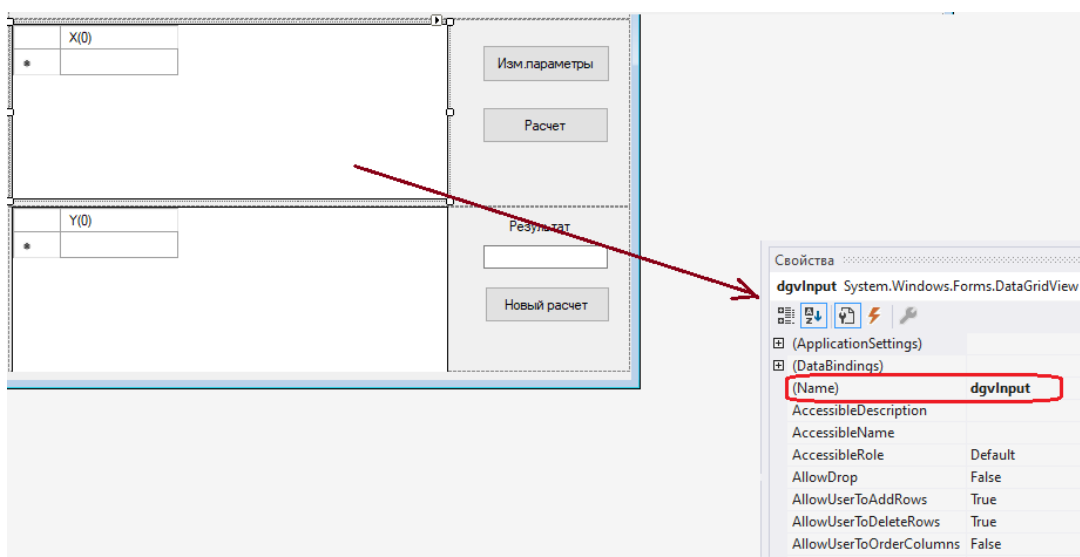


Растянем панель на всю ширину окна



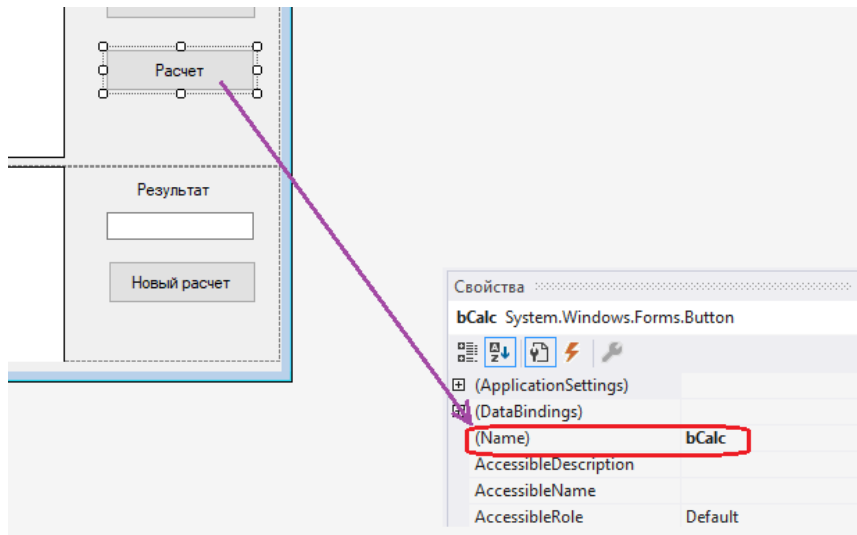
Далее добавляем остальные элементы формы.

Каждый элемент можно настроить (положение, размеры и т.п.) на панели свойств.

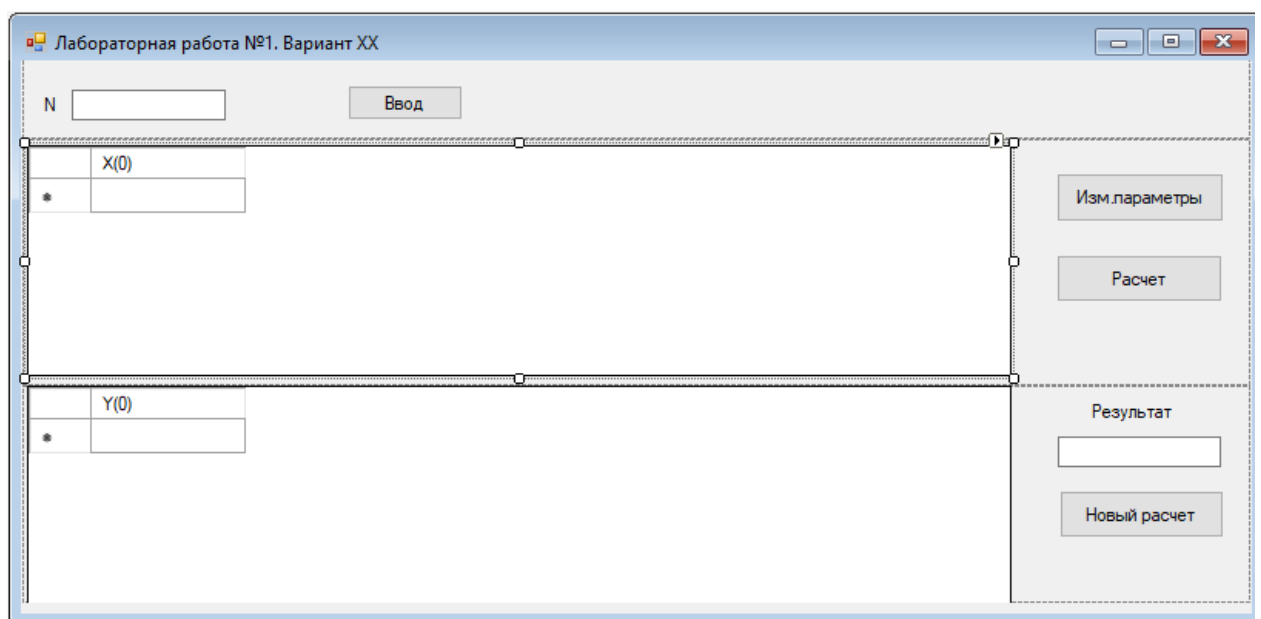


Для тех элементов, которые используются в программном коде (все, кроме элементов статического текста) необходимо заменить имя (свойство Name) с обезличенного на содержательное.

Например, для кнопки Расчет вместо обезличенного имени *Button5* зададим содержательное *bCalc*



Окончательный вид формы после добавления всех элементов



Необходимые переменные и массивы должны быть объявлены в начале класса:

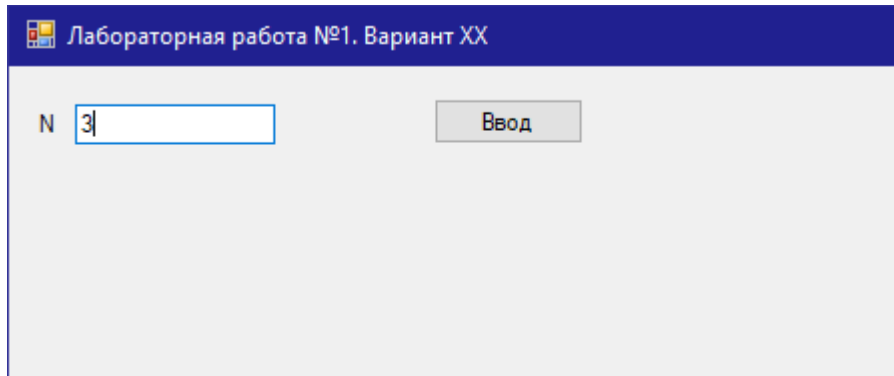
```
Ссылка 3
public partial class FormMain : Form
{
    int N, N1;
    int[] x, y;
```



4. Работа программного приложения

В ходе работы приложения форма в разных режимах должна иметь следующий вид.

1) Вид формы при запуске



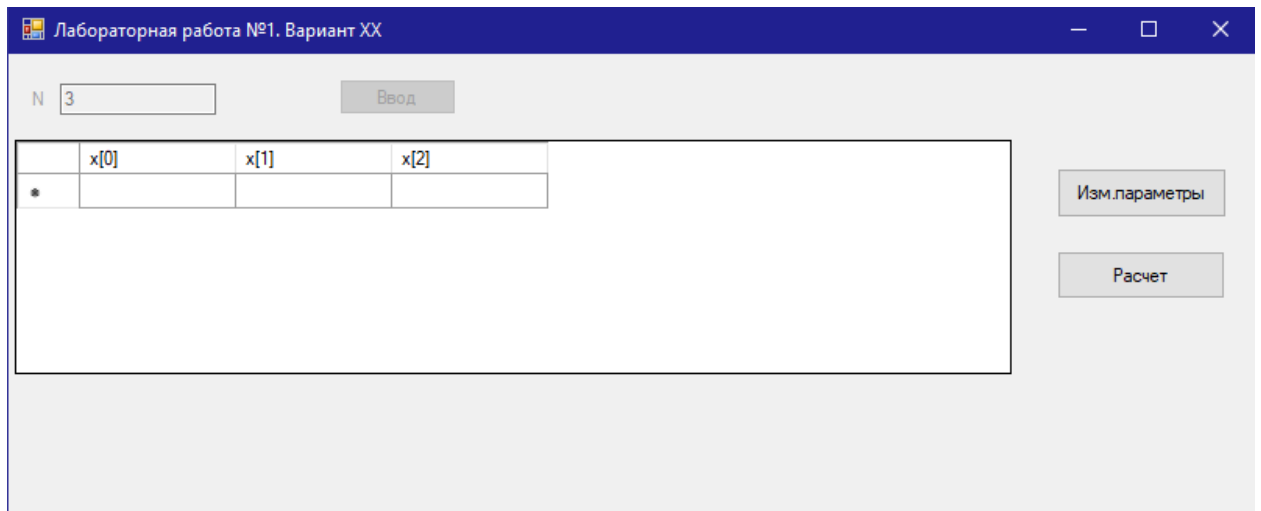
Лабораторная работа №1. Вариант XX

N 3 Ввод

Панели ввода исходного массива и вывода результата должны быть невидимыми (свойство `Visible = false`).

Необходимо контролировать правильность задания количества элементов массива (положительное число, не текст)

2) Вид формы при нажатии кнопки Ввод



Лабораторная работа №1. Вариант XX

N 3 Ввод

	x[0]	x[1]	x[2]
*			

Изм.параметры

Расчет

Панель задания параметров должна быть недоступна (свойство `Enabled=false`).

Панель вывода результатов должна быть невидимой (свойство `Visible = false`).

Количество строк таблицы должно быть равно 1 (свойство `RowCount=1`).

Предварительно надо очистить все строки (метод `Rows.Clear()`).

Количество колонок должно быть равно введенному значению N (свойство *ColCount=N*). Здесь N – это переменная, значение которой определяется числом, введенным в поле ввода «N» (компонент txtN). Причем введенное в поле значение должно быть преобразовано из символьного значения в целое:

```
Convert.ToInt32(txtN.Text) .
```

Заголовочная строка таблицы сетки должна содержать номера элементов массива: X[0], X[1] и т.д.. Для этого в цикле необходимо выполнить оператор

```
dgvInput.Columns[i].HeaderText="x["+i.ToString()+"]";
```

3) При нажатии кнопки *Изм. параметры* вид формы должен вернуться к начальному виду.

4) При нажатии кнопки *Расчет* форма должна иметь вид

Лабораторная работа №1. Вариант XX

N 3 Ввод

	x[0]	x[1]	x[2]
▶	2	5	1
*			

Изм. параметры

Расчет

	y[0]	y[1]
*	0	1

Результат

6

Новый расчет

Панель задания параметров и ввода массивы должны быть недоступными (свойство *Enabled=false*).

Для выполнения расчетов необходимо создать массив и заполнить его значениями входного массива, взятыми из таблицы-сетки. Так как значения имеют символьное значение их надо преобразовать в вещественное значение. Для этого в цикле необходимо выполнить оператор

```
x[i]=Convert.ToInt32(dataGridView1.Rows[0].Cells[i].Value);
```

После получения результирующего массива необходимо его значения необходимо записать в результирующую таблицу-сетку. Для этого в цикле необходимо выполнить оператор

```
dgvRezult.Rows[0].Cells[i].Value = y[i];
```

Заголовочная строка таблицы сетки должна содержать номера элементов массива: Y[0], Y[1] и т.д.

Если результирующий массив не сформирован (нет подходящих элементов), то вывести «массив не сформирован»

Скалярный результат записывается в элемент-поле. Числовое значение должно быть преобразовано в символьную строку методом ToString().

Если скалярное результирующее значение не может быть сформировано, то в текстовом поле написать «нет результата»

- 5) При нажатии кнопки *Новый расчет* вид формы должен вернуться к начальному виду.**

5. Создание обработчиков для кнопок

Все операции выполняются в процедурах- обработчиках, связанных с кнопками. Для создания обработчика надо сделать двойной клик на кнопке. В файле программного кода будет создан обработчик, в теле которого надо ввести необходимые операторы

