



.NET

Windows

Forms

Благодаря незаурядной мощности языка, на него пал выбор разработчиков движка Unity. Сегодня — является одним из топовых движков для игр на Windows. Выпуск и активное использование движка пошли на руку C#, который стал ещё популярнее.

```
25 InitializeComponent();
26
27 //this.Title = "Hello To All";
28 this.WindowStartupLocation = WindowStartupLocation.CenterScreen;
29 }
30
31 private void BtnMessageBox_Click(object sender, RoutedEventArgs e)
32 {
33     MessageBox.Show("Hi everyone, how are you doing?");
34 }
35
36 private void CloseLabel_MouseDoubleClick(object sender, MouseButtonEventArgs e)
37 {
38     this.Close();
39 }
40
41 private void GetInput_Click(object sender, RoutedEventArgs e)
42 {
43     string userInput = userInputField.Text;
44     if (userInput != "")
45     {
46     }
47 }
48 }
49
```

В код можно вставить комментарии. Комментарии можно вставлять в любую область кода, чтобы делиться с другими разработчиками.

Какова роль .NET?

«Платформа .NET – лучшее творение Microsoft»

Ключевые черты

платформы:

- **Работает параллельно с разными языками.**

Популярностью C# во многом обязан общеязыковой среде CLR. Сейчас платформа способна работать с C#, VB.NET, C++, F#, но и на этом список не заканчивается, ведь она работает с диалектами, что привязаны к .NET (наподобие Delphi.NET). После компиляции кода с любого из перечисленных языков, вся интерпретируется в общий язык CIL – это своеобразный ассемблер для .NET. Такой подход позволяет использовать несколько языков для создания подключаемых модулей программы;

Ключевые черты платформы:

- **Кроссплатформенность.**

Данную платформу реально переносить, хоть и есть отдельные ограничения. Сегодня актуальная версия фреймворка работает на всех поддерживаемых Виндовс. За счёт проекта Mono появилась возможность разрабатывать программы под Linux (различные дистрибутивы), Android и iOS;

Ключевые черты
платформы
• унифицированная библиотека классов.

NET Framework обладает единой,
унифицированной библиотекой классов, с
которой работают все поддерживаемые
языки. Библиотека классов пригодится при
создании любых программ: от блокнота до
огромного веб-сайта;

Ключевые черты
платформы
• унифицированная библиотека классов.

NET Framework обладает единой,
унифицированной библиотекой классов, с
которой работают все поддерживаемые
языки. Библиотека классов пригодится при
создании любых программ: от блокнота до
огромного веб-сайта;

Ключевые черты

• Платформа фронт-энд технологий.

Среда CLR в сочетании с библиотекой классов – это основа для большого пакета вспомогательных технологий. Их могут использовать все программисты во время разработки приложений. В качестве примера, при взаимодействии с базами данных можно использовать технологию ADO.NET. Во время создания графических редакторов с многочисленными функциями удобно использовать WPF. Во время веб-разработки наверняка используют ASP.NET.

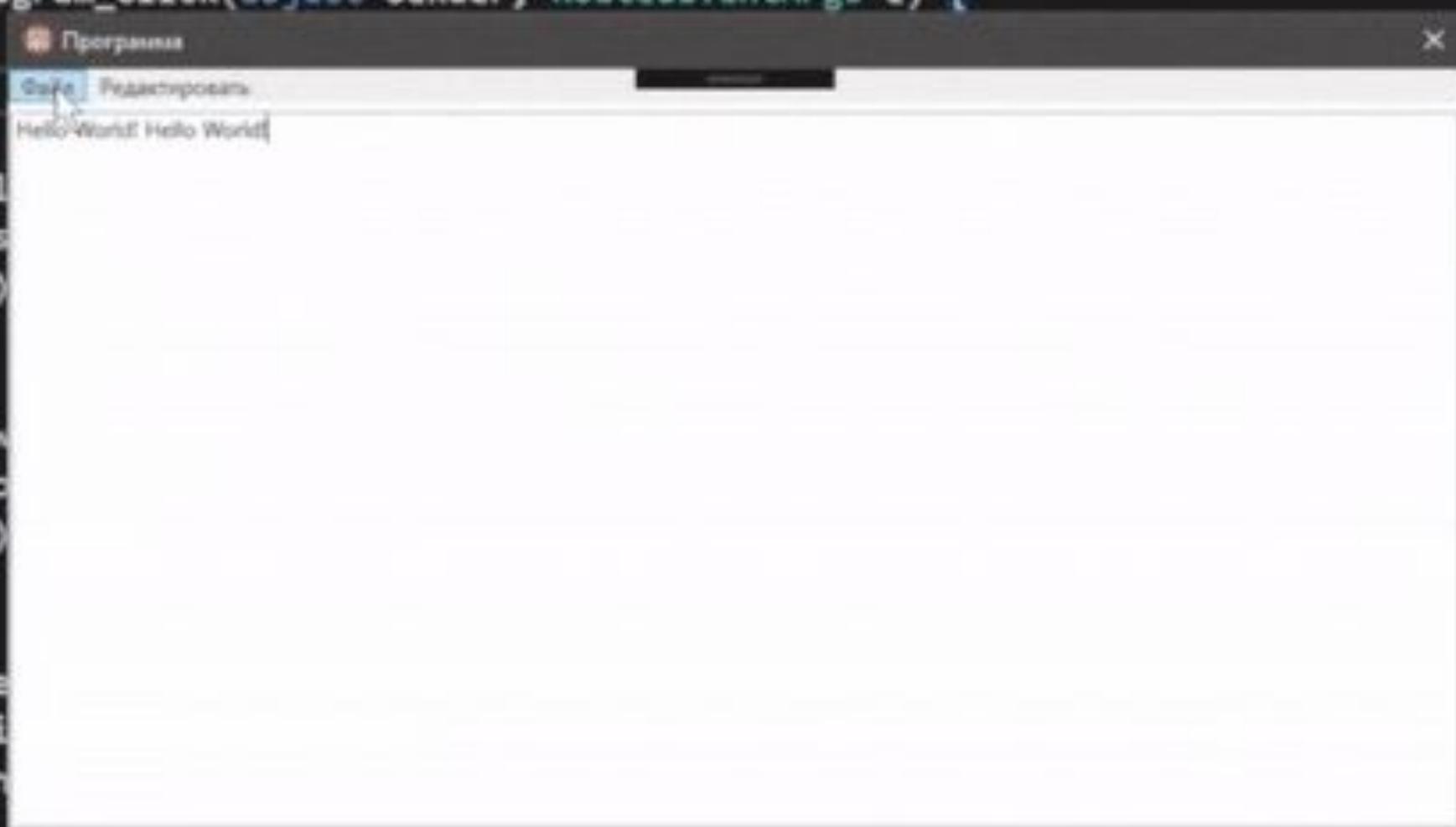
```
te void ExitProgram_Click(object sender, RoutedEventArgs e) {  
this.Close();
```

```
te void SaveFile  
aveFileDialog s  
Fd.ShowDialog()
```

```
te void OpenNew  
penFileDialog o  
Fd.ShowDialog()
```

```
te void TimesNe  
textBox.FontFami  
ardanaFont.IsCh
```

```
te void VerdanaFont_Click(object sender, RoutedEventArgs e) {
```



Особенность
фреймворка и языка
– автоматическая
очистка хлама в
памяти.

Windo

WS

Forms

Это платформа, на основе которой можно строить программы с графическим интересом, но при этом лишь под Windows.

1. Создайте проект



Приложение Windows Forms (.NET Framework)

Проект для создания приложения с пользовательским интерфейсом Windows Forms (WinForms)

C#

Windows

Рабочий стол



Библиотека элементов управления Windows Forms (.NET Framework)

Проект для создания элементов управления, которые будут использоваться в приложениях Windows Forms (WinForms)

C#

Windows

Рабочий стол

Библиотека



Приложение Windows Forms (.NET Framework)

Проект для создания приложения с пользовательским интерфейсом Windows Forms (WinForms)

Visual Basic

Windows

Рабочий стол



Библиотека элементов управления Windows Forms (.NET Framework)

Проект для создания элементов управления, которые будут использоваться в приложениях Windows Forms (WinForms)

Visual Basic

Windows

Рабочий стол

Библиотека

1. Настройте новый проект

1. Имя задайте как на картинке. (Название проекта для пользователей)
2. Расположение можете выбрать любой
3. Имя решения – это общее название проекта внутри системы

Приложение Windows Forms (.NET Framework)

C#

Windows

Рабочий стол

Название проекта

ExampleSQLApp

Расположение

C:\Users\admin\source\repos

Имя решения 

ExampleSQLApp

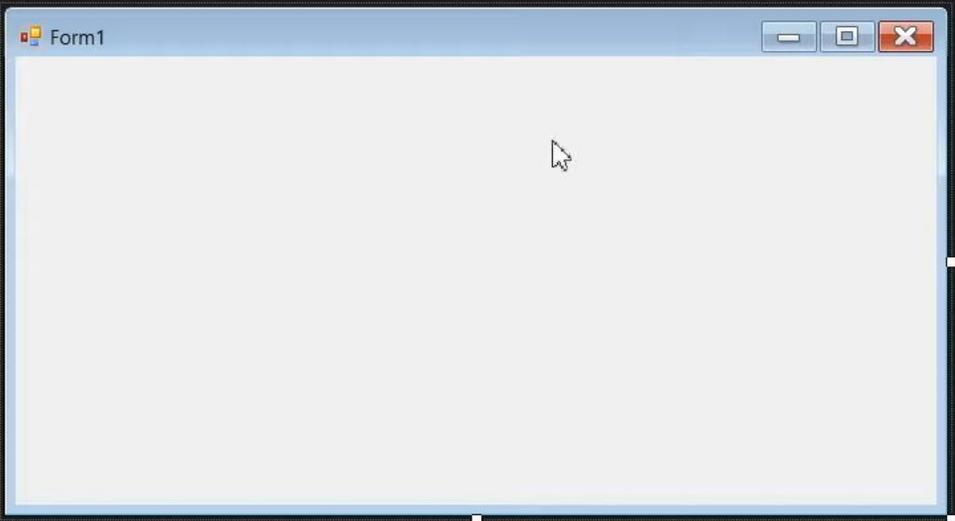
Поместить решение и проект в одной папке

Платформа

.NET Framework 4.7.2

Обозреватель серверов
Панель элементов

Form1.cs [Конструктор]*



Обозреватель решений

Обозреватель решений — поиск (Ctrl+)

Решение "ExampleSQLApp" (проекты: 1 из 1)

- ExampleSQLApp
 - Properties
 - Ссылки
 - App.config
 - Form1.cs
 - Program.cs

Обозреватель решений Team Explorer

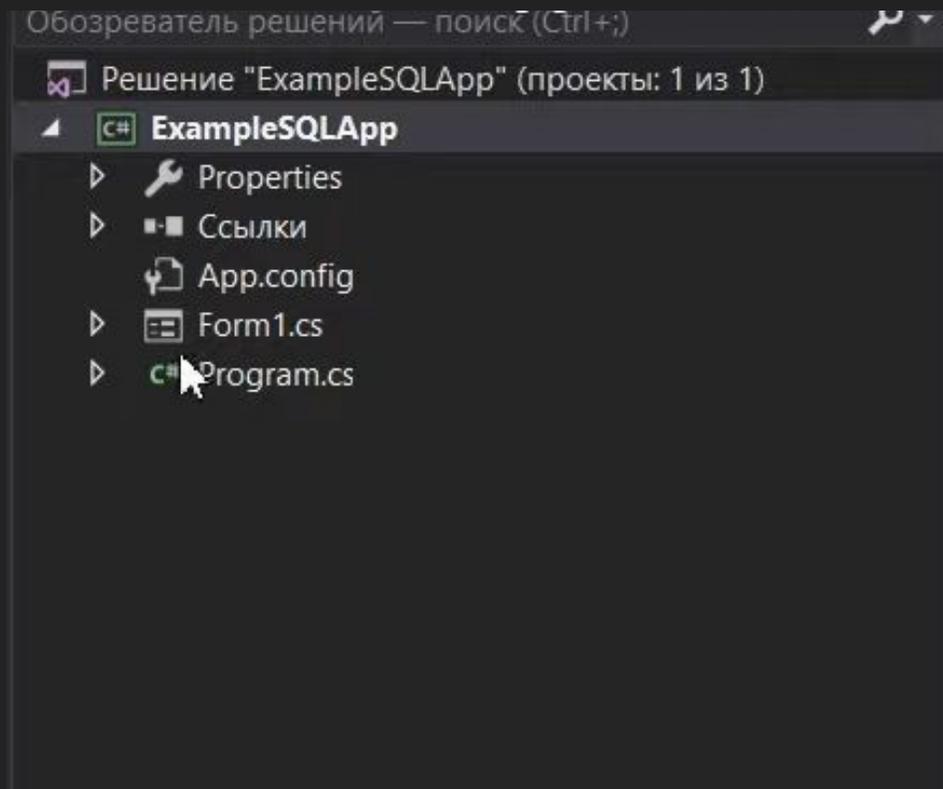
Свойства

Form1 System.Windows.Forms.Form

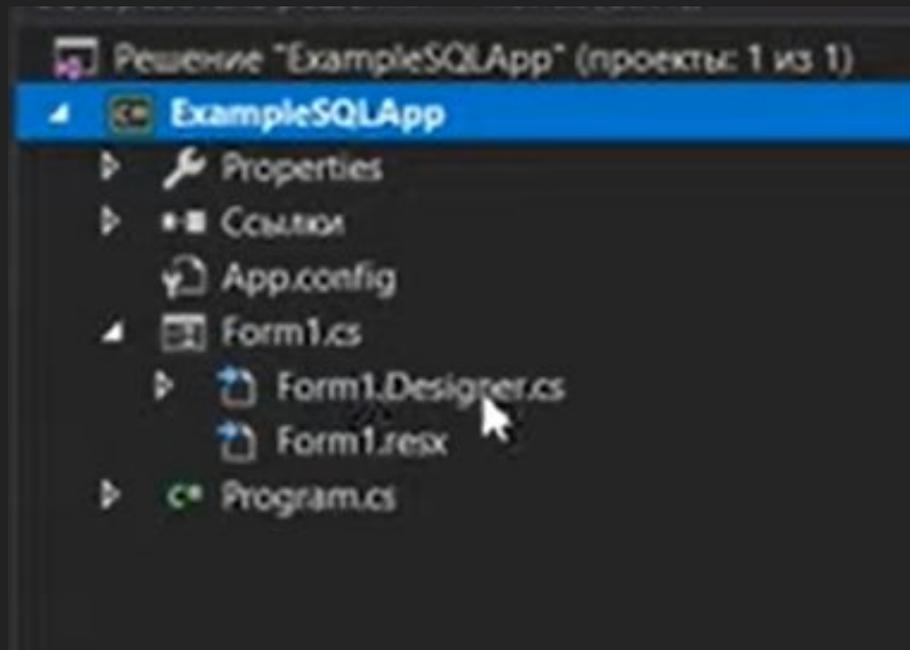
BackgroundImage	(отсутствует)
BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 7,8pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1
UseWaitCursor	False

Text
Текст, связанный с элементом управления.

Обозреватель решений



**Иерархия из всех тех
объектов и также
файлов, которые
находятся внутри
программы**



В Form1.cs находится наша форма.

**При открытии мы видим
Form1.Designer.cs – графический
конструктор**

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace ExampleSQLApp
8 {
9     static class Program
10     {
11         /// <summary>
12         /// Главная точка входа для приложения.
13         /// </summary>
14         [STAThread]
15
16         static void Main()
17         {
18             Application.EnableVisualStyles();
19             Application.SetCompatibleTextRenderingDefault(false);
20             Application.Run(new Form1());
21         }
22     }
23 }
```

Program.cs - главный класс, определяет точку входа в приложение

Данный файл содержит класс Program. Выполнение программы на языке C# начинается с метода Main. И в классе Program как раз определен подобный метод.

Обозреватель решений

Обозреватель решений — поиск (Ctrl+)

- Решение "ExampleSQLApp" (проекты: 1 из 1)
 - ExampleSQLApp
 - Properties
 - Ссылки
 - App.config
 - Form1.cs
 - Program.cs
 - Program

Обозреватель решений Team Explorer

Свойства

Обозреватель серверов
Панель элементов

Метод **Main** снабжен атрибутом **[STAThread]**. Этот атрибут необходим для корректной работы компонентов Windows.

В самом методе сначала вызывается метод `ApplicationConfiguration.Initialize()` который устанавливает некоторую базовую конфигурацию приложения. Затем вызывается метод

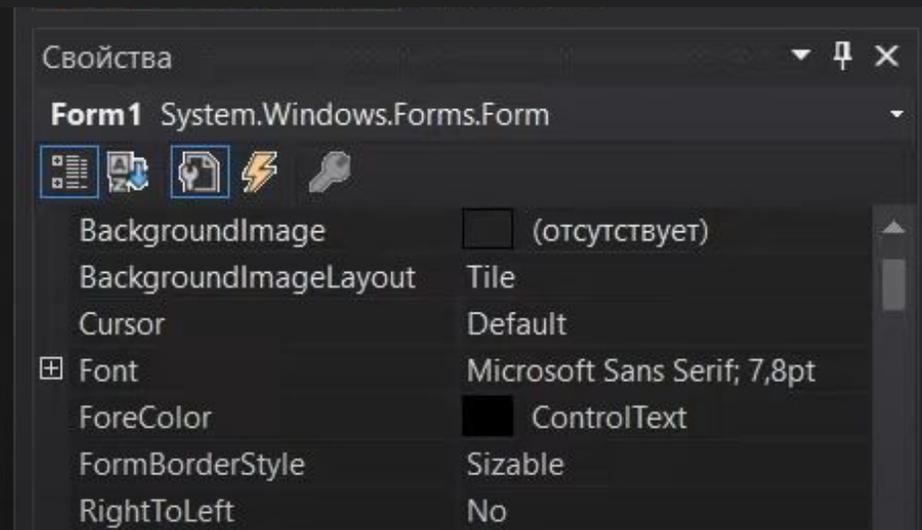
```
Application.Run(new Form1());
```

в который передается объект отображаемой по умолчанию на экране формы.

То есть, когда мы запустим приложение, сработает метод **Main**, в котором будет вызван метод `Application.Run(new Form1())`, благодаря чему мы увидим форму `Form1` на экране

`Application.Run(new Form1());` -
указываем какую форму вызываем.
В нашем случае это Form 1.

Свойства

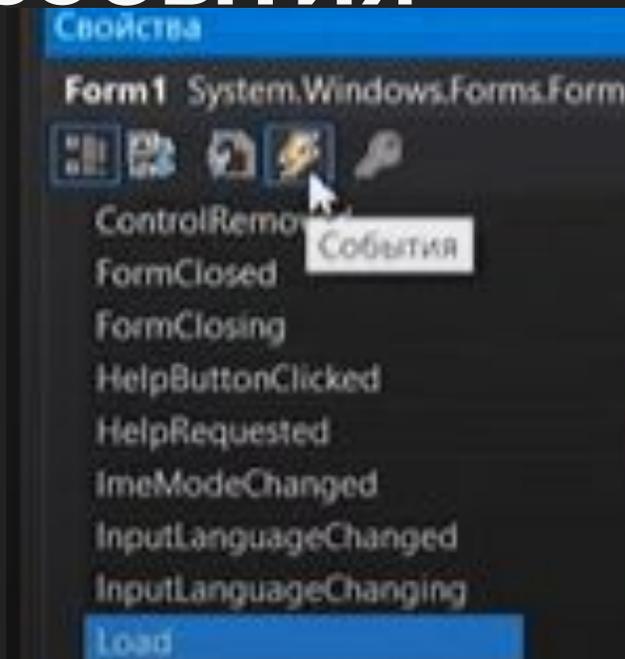


Отображаются свойства для различных объектов

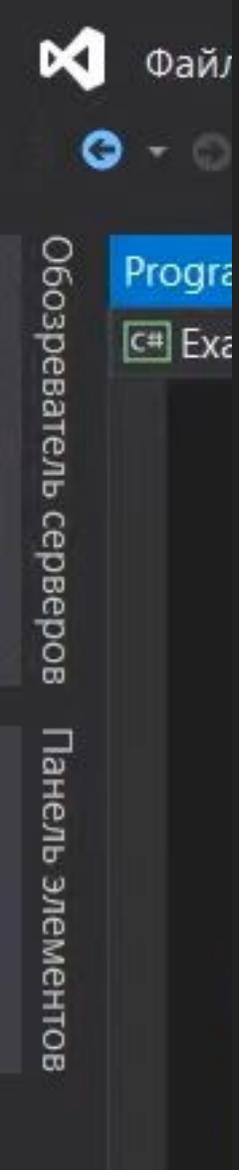
3. Посмотрите свойства Form1.

Попробуйте изменить задний фон (BackColor) на любой цвет.

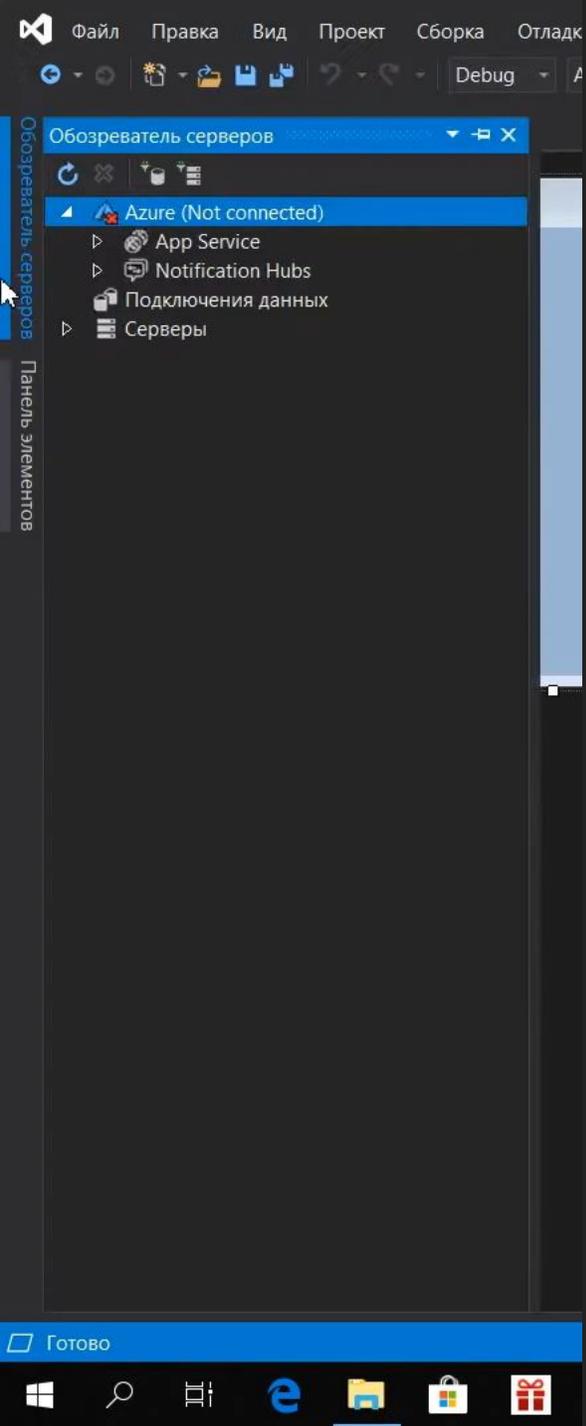
Также можно устанавливать различные события



Например, при нажатии мышки (MouseDown) будет происходить какое-то действие



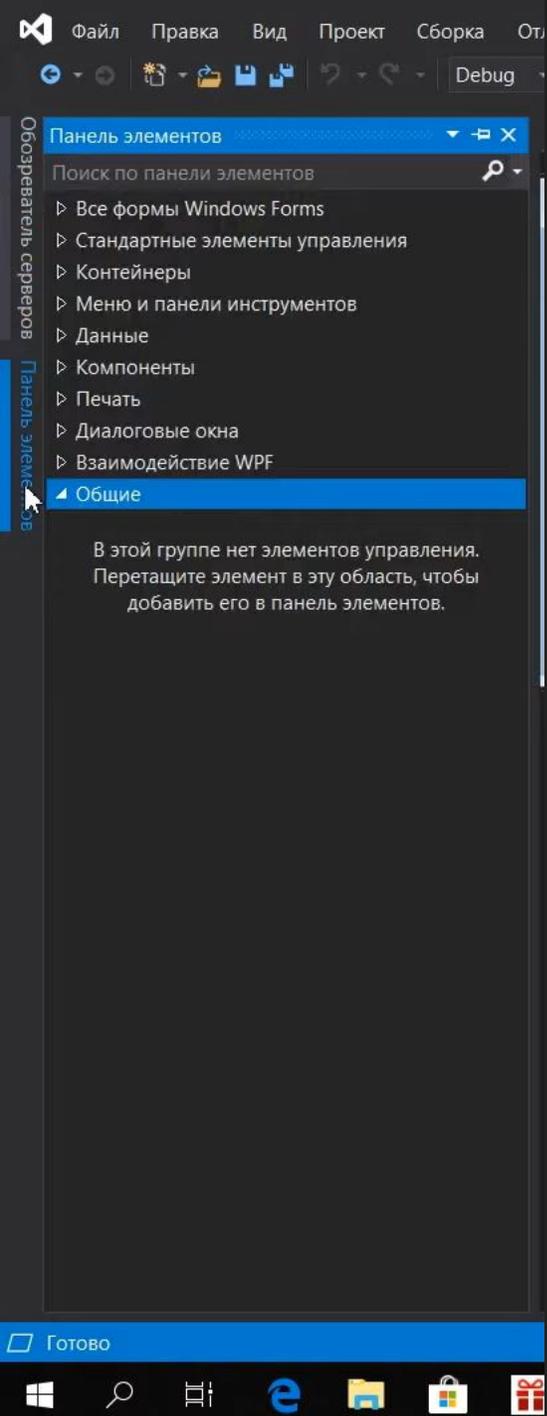
**Если некоторые вкладки не отображаются, то нажимаем меню ВИД -
выбираем любое окно, которое нам необходимо**



Обозреватель серверов

Тут находятся различные характеристики, настройки.

Для того что мы могли работать с сервером, БД.



Панель элементов

Здесь находятся различные элементы, которые мы можем перетащить на саму форму.

Примечание: при работе с окном, лучше его закрепить, чтобы было удобнее перетаскивать элементы

4. Добавьте кнопку `button`. Измените ее высоту и ширину



Также можем изменить ее свойства и добавить какие либо события

5. Попробуйте изменить цвет кнопки и начертание текста

6. Нажмите Вид-код

```
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ExampleSQLApp
12 {
13     ссылка: 3
14     public partial class Form1 : Form
15     {
16         ссылка: 1
17         public Form1()
18         {
19             InitializeComponent();
20         }
21     }
22 }
```

Класс Form1 соответствует названию нашей формы.

Все это наследуется от базового класса Form

Внутри класса мы видим конструктор, который не принимает никаких параметров. Он создает нам объект.

Form1.cs и Form1.cs [Конструктор] – одно и то же.
Представление кода или графического формата

7. Запуск приложения

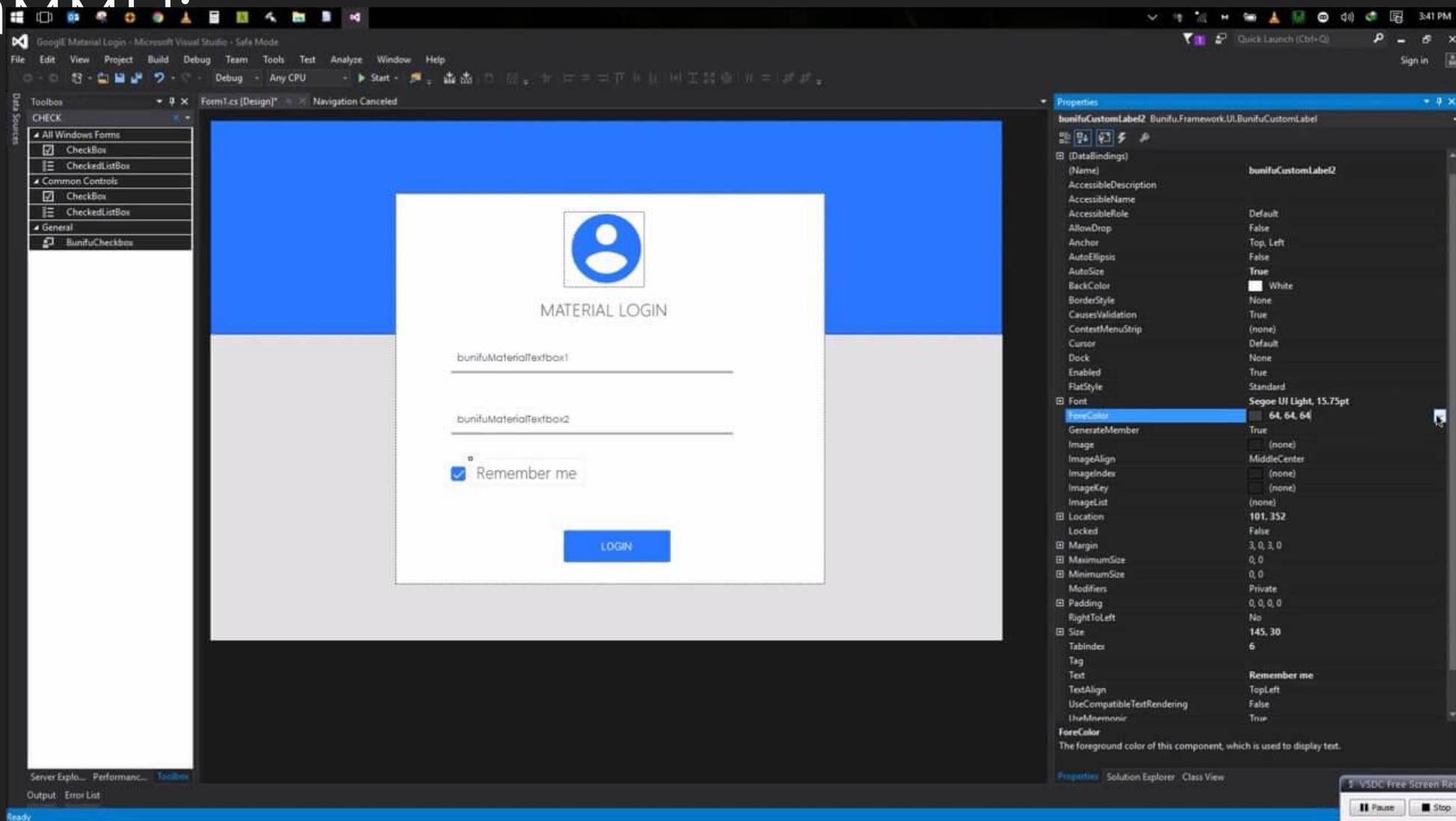
Запустите приложение

Полезные ссылки:

подборка цвета - [color picker](#);
иконки для приложений
- [iconfinder](#);

При разработке дизайна всегда стоит подготавливать макет готовой программы. Такой макет можно создать в PhotoShop, Figma, Sketch или в любых других программах, которые отвечают за разработку дизайна. Имея готовый макет вам будет проще расставлять объекты, добавлять к ним цвета, устанавливать форму и производить другие манипуляции.

На основе **WinForms** можно создавать абсолютно любой дизайн программы. Пример программы:



Библиотеки

Помимо использования стандартных стилей, вы всегда можете воспользоваться сторонними библиотеками, которые позволят быстрее создавать еще более красивые дизайны для приложений.

Несколько таких библиотек приведено ниже:

- Специализированная библиотека [Bunifu](#);
- Фреймворк [WPF](#);
- [Xamarin Forms](#).

Создание дизайна

В основе своей, создание дизайна разбивается на несколько этапов:

- Добавление объектов на главное окно;
- Добавление стилей для объектов. Можно добавить стили не только стандартные, но и стили из различных библиотек;
- Добавление обработчиков событий.

Звучит просто, хотя на деле все сложнее. Вам стоит самостоятельно попрактиковаться и создать несколько вариантов дизайна программы.