

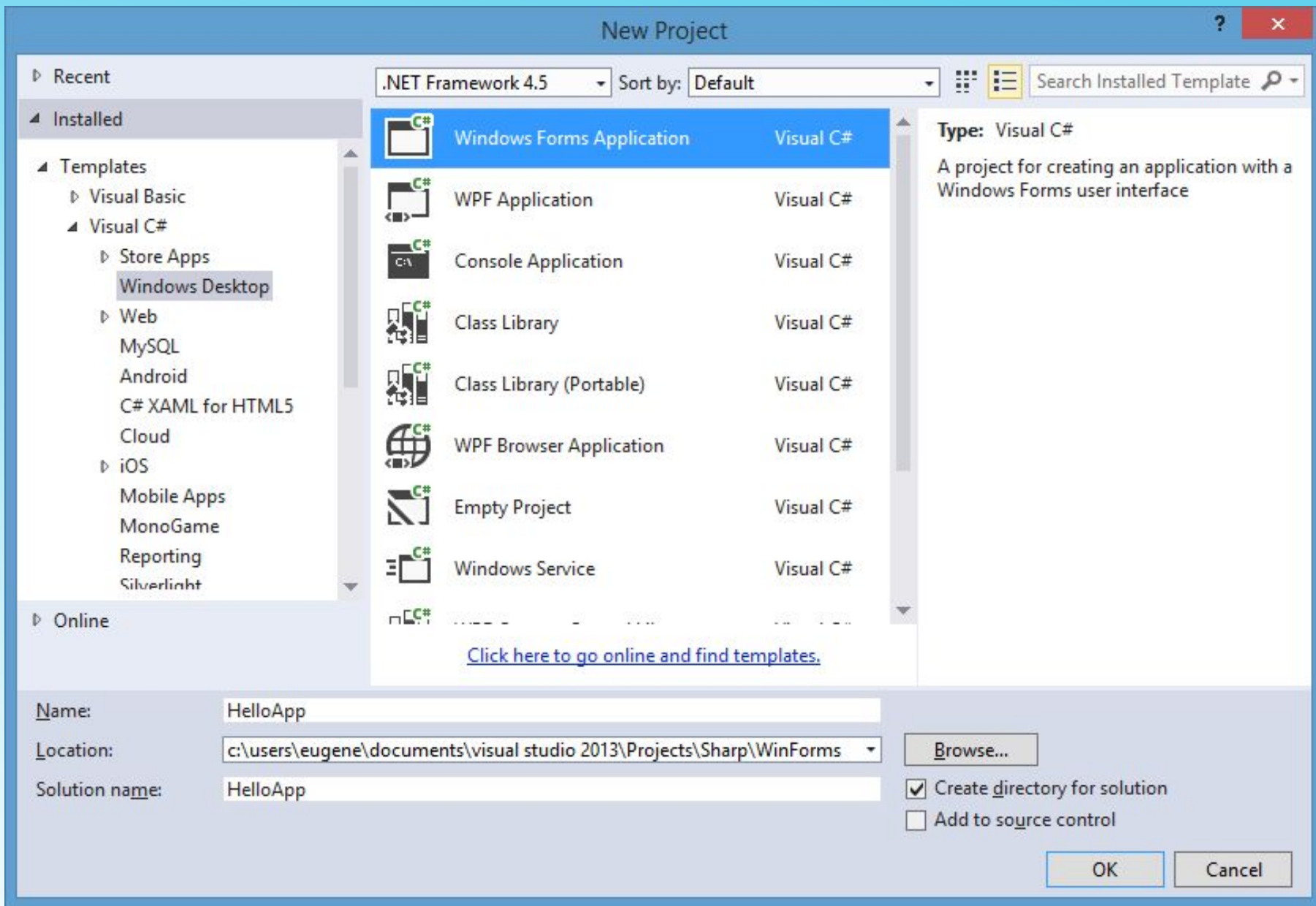
БАҒДАРЛАМАЛАРДЫҢ
ГРАФИКАЛЫҚ ИНТЕРФЕЙСІН
ЖАСАУ ТЕХНОЛОГИЯЛАРЫ.

.NET платформасын қолдана отырып графикалық интерфейстерді құру үшін әртүрлі технологиялар қолданылады - Windows Forms, WPF, Windows Store дүкеніне арналған қосымшалар (Windows 8/8.1/10 үшін). Дегенмен, ең қарапайым және ыңғайлы платформа әлі күнге дейін Windows Forms немесе нысандары болып табылады. Бұл нұсқаулық WinForms технологиясын және негізгі басқару элементтерін қолдана отырып, графикалық интерфейстерді құру принциптерін түсінуді мақсат етеді.

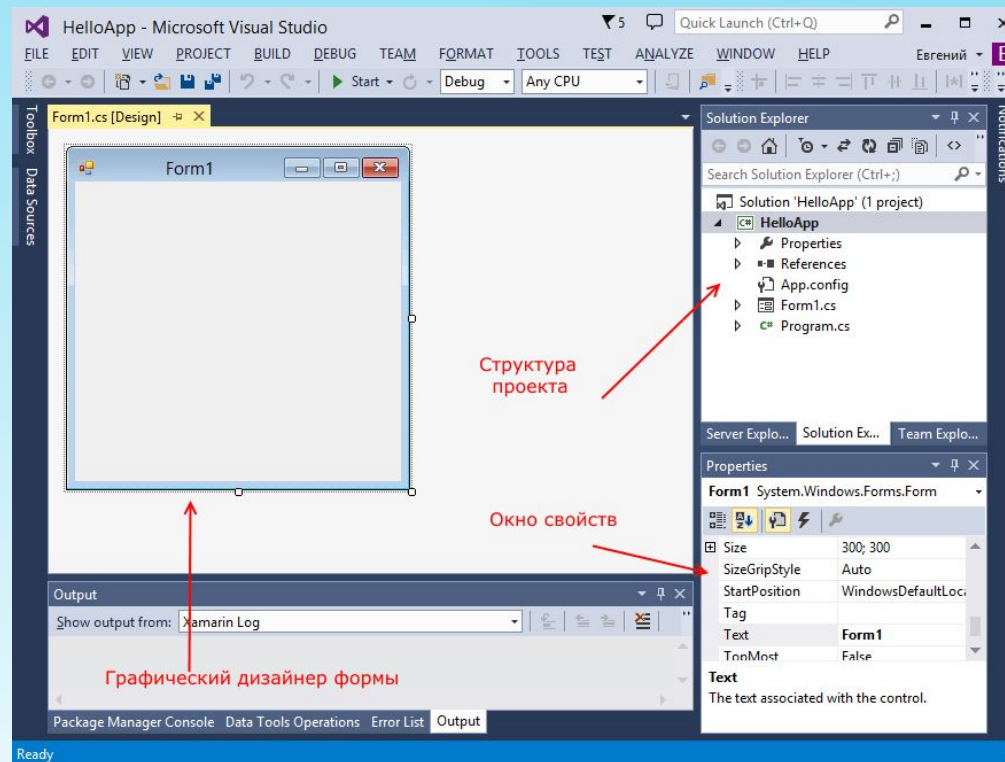
Windows Forms - пайдаланушының графикалық интерфейсіне жауап беретін және Microsoft.NET Framework құрамына кіретін қосымшаларды бағдарламалау интерфейсі (API). Бұл интерфейс басқарылатын кодта бар Win32 API үшін орауыш жасау арқылы Microsoft Windows интерфейсінің элементтеріне кіруді жеңілдетеді. Сонымен қатар, Басқарылатын код — Windows Forms үшін API енгізетін класстар даму тіліне тәуелді емес. Яғни, бағдарламашы Windows Forms-ті C#, C++ және C ++ тілдерінде жазу кезінде бірдей қолдана алады.

Графикалық қосымшаны құру

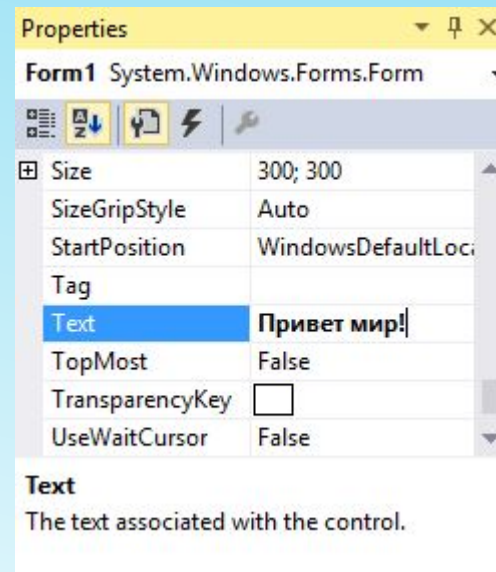
- ▶ Графикалық жобаны жасау үшін бізге Visual Studio дамыту ортасы қажет.
- ▶ Ортаны және оның барлық компоненттерін орнатқаннан кейін Visual Studio бағдарламасын іске қосыңыз және графикалық қосымшаның жобасын жасаңыз. Ол үшін мәзірден файл (Файл) тармағын таңдап, ішкі мәзірден жаңа - > жобаны таңдаңыз (жасау - > Жоба). Осыдан кейін біз жаңа жобаны құру үшін диалогтық терезені ашамыз:



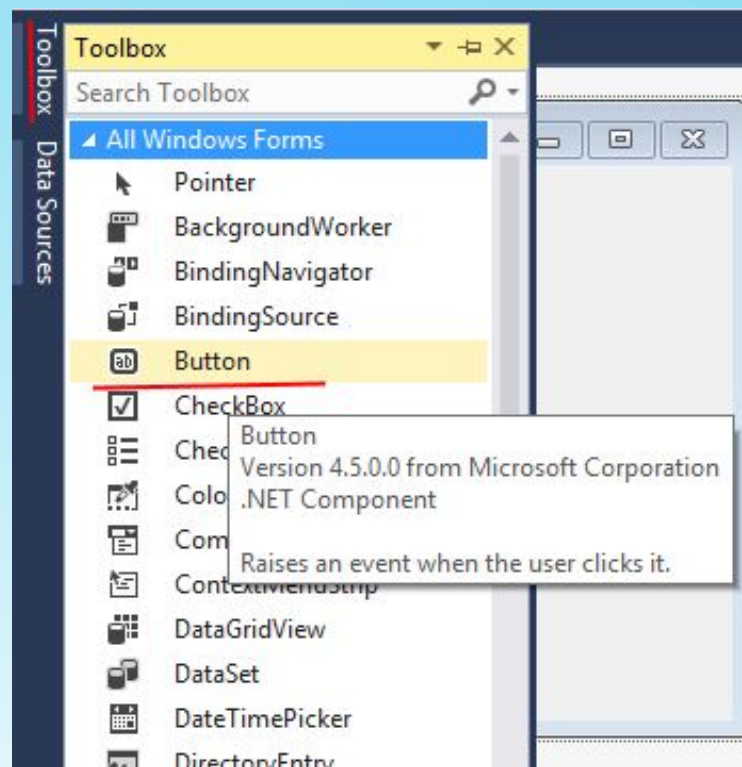
- ▶ Сол жақ бағанда біз Windows Desktop, ал орталық бөлігінде жоба түрлерінің арасында Windows Forms Application түрін таңдаймыз және оған төмендегі өрісте қандай да бір атау береміз. Мысалы, оны HelloApp деп атайық. Осыдан кейін ОК түймесін басыңыз. Осыдан кейін Visual Studio біздің жобамызды әдепкі файлдармен ашады:



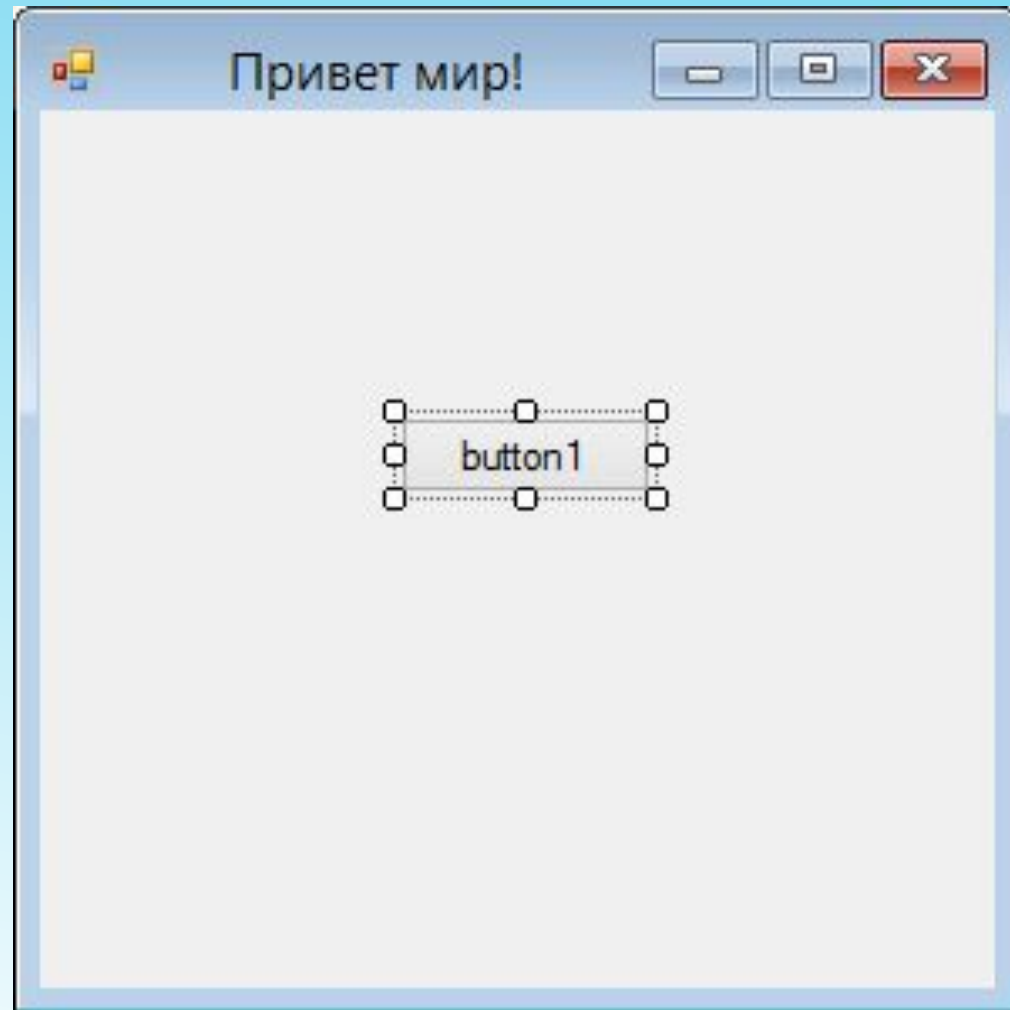
- ▶ Visual Studio кеңістігінің көп бөлігін болашақ қосымшаның пішінін қамтитын графикалық дизайнер алады. Әзірге ол бос және тек Form1 тақырыбы бар. Оң жақта шешім/жоба файлдарының терезесі - Solution Explorer (шешім шолушысы). Біздің қосымшамен байланысты барлық файлдар, соның ішінде Form1 пішінді файлдар бар .cs. Төменгі оң жақта қасиеттер терезесі-Properties. Қазіргі уақытта басқару элементі ретінде форма таңдалғандықтан, пішінмен байланысты қасиеттер осы өрісте көрсетіледі. Енді осы терезеде Text пішінінің сипатын тауып, оның мәнін басқасына өзгертіңіз:



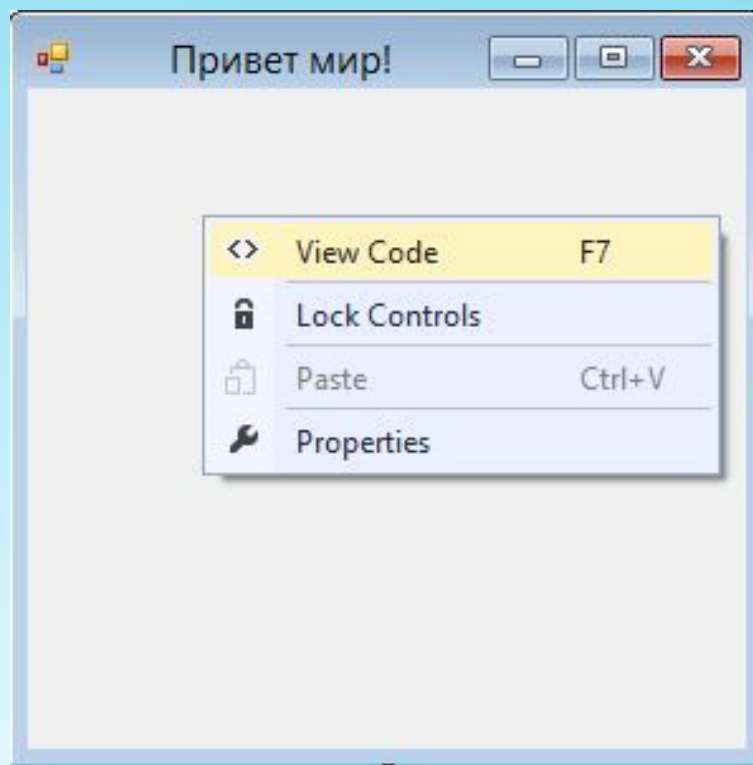
- ▶ Осылайша біз пішіннің тақырыбын өзгерттік. Енді біз кейбір басқару элементтерін, мысалы, батырманы өріске жібереміз. Ол үшін Visual Studio-ның сол жағында TOOLBOX (құралдар тақтасы) қойындысын табыңыз. Осы қойындыны нұқыңыз, сонда біз тінтуірдің көмегімен кез-келген элементті пішінге бере алатын элементтері бар панельді ашамыз:



- ▶ Элементтердің ішінен түймені тауып, оны тінтуір меңзерімен басып, пішінге өткізіңіз:



- ▶ Бұл көрнекі бөлік. Енді біз бағдарламалауды бастаймыз. Түймені басу арқылы хабарламаны көрсететін c# тіліндегі қарапайым кодты қосыңыз. Ол үшін біз осы формамен байланысты Код файлына өтуіміз керек. Егер бізде код файлы ашық болмаса, форманы тінтуірдің оң жақ түймесімен нұқып, пайда болған мәзірден қарау кодын таңдаңыз (код файлын қараңыз):



- ▶ Алайда, біз қосымша кодты көп жазбау үшін басқа әдісті қолданамыз. Керек батырманы тінтуір батырмасын екі рет шертіп, біз автоматты түрде Form1 код файлына кіреміз .cs, ол келесідей көрінеді:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
1 using System.Drawing;
2 using System.Linq;
3
4 using System.Text;
5 using System.Threading.Tasks;
6
7 using System.Windows.Forms;
8
9
10 namespace HelloApp
11 {
12     public partial class Form1 : Form
13     {
14         public Form1()
15         {
16             InitializeComponent();
17         }
18
19         private void button1_Click(object sender, EventArgs e)
20         {
21
22         }
23     }
24 }
25 }
```

Кодты келесідей өзгерту арқылы түймені басу арқылы хабарлама шығысын қосыңыз:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HelloApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

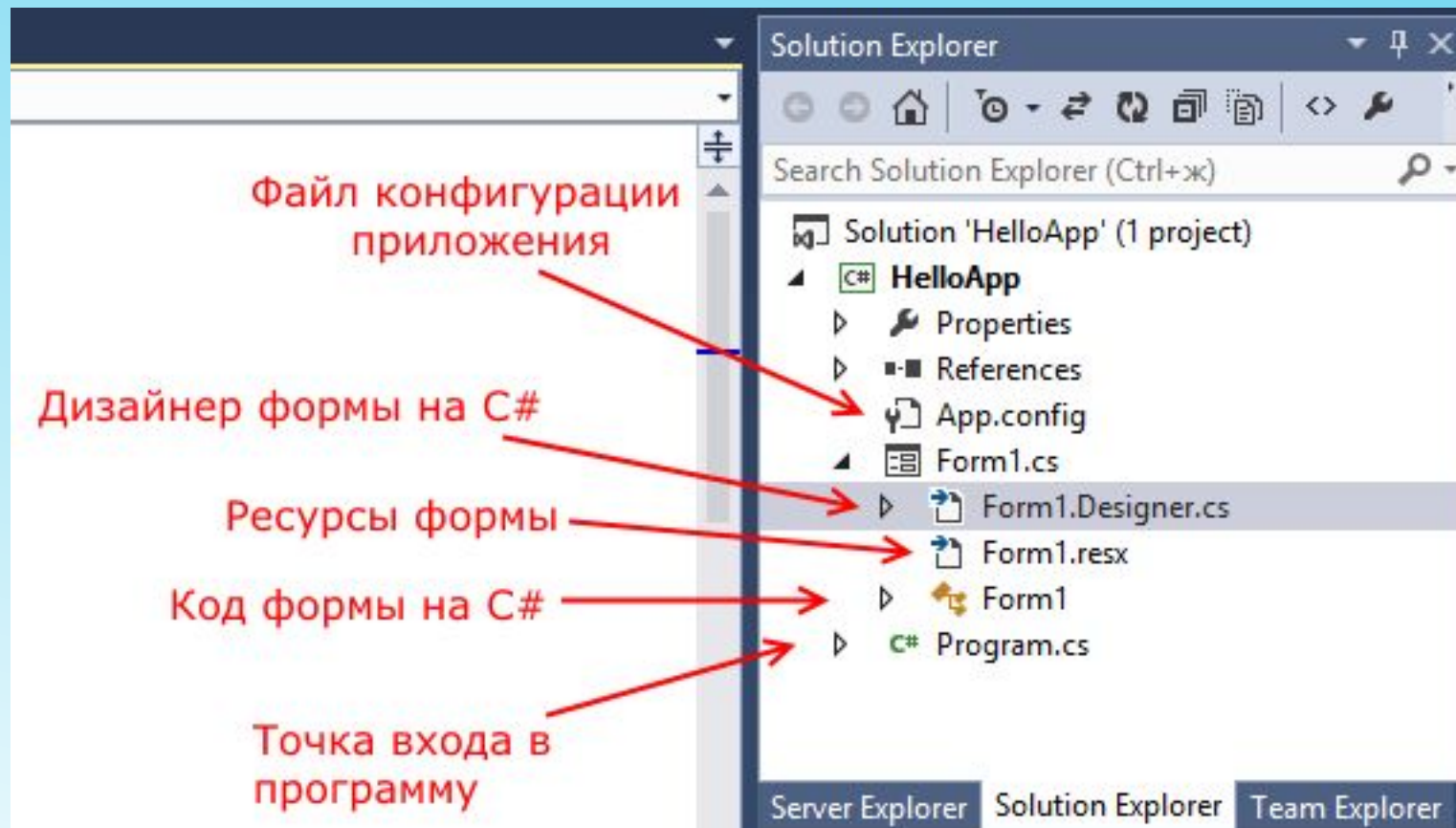
        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Привет");
        }
    }
}
```

- ▶ Қолданбаны іске қосу
- ▶ Бағдарламаны күйін келтіру режимінде іске қосу үшін F5 пернесін немесе Visual Studio тақтасындағы жасыл көрсеткіні нұқыңыз. Осыдан кейін біздің формамыз жалғыз батырмамен басталады. Егер біз формадағы батырманы бассaq, онда бізге сәлемдесу хабары көрсетіледі. Бағдарламаны іске қосқаннан кейін студия оны .exe кеңейтімі бар файлға құрастырады. Сіз бұл файлды жоба қалтасына кіріп, содан кейін bin/Debug немесе bin/Release каталогынан таба аласыз.

▶ Формалармен жұмыс

- ▶ Қосымшаның пайда болуы бізге негізінен формалар арқылы келеді. Пішіндер негізгі құрылыс блоктары болып табылады. Олар әртүрлі басқару элементтері үшін контейнер ұсынады. Оқиға механизмі пішін элементтеріне пайдаланушының кірісіне жауап беруге және осылайша пайдаланушымен өзара әрекеттесуге мүмкіндік береді. Жобаны графикалық редакторда Visual Studio-да ашқан кезде біз форманың көрнекі бөлігін көре аламыз. Бірақ іс жүзінде форма әдістерден, қасиеттерден, оқиғалардан және басқалардан тұратын қуатты функционалдылықты жасырады.

- ▶ Пішіндердің негізгі қасиеттерін қарастырайық. Егер біз қосымшаны іске қосатын болсақ, онда бізге бір бос форма көрсетіледі. Дегенмен, тіпті бос нысаны бар қарапайым жобаның бірнеше компоненттері бар:

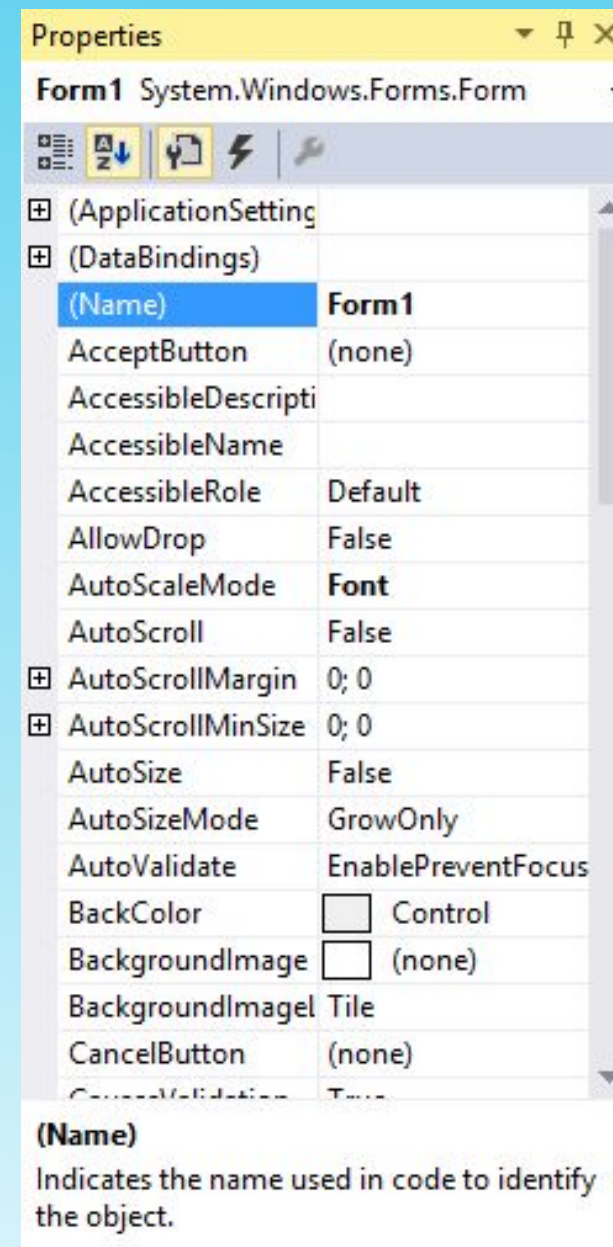


Біз тек пішінді көреміз, бірақ графикалық қосымшаның кіру нүктесі - бұл **Program.cs** файлында орналасқан **Program** класы:

Алдымен бағдарлама осы класты бастайды, содан кейін `Application.Run(new Form1())` өрнегін қолданып `Form1` пішінін іске қосады. Егер кенеттен біз қосымшадағы бастапқы форманы басқа формаға өзгерткіміз келсе, осы `Form1` өрнегінде тиісті форма класына өзгерту керек.

```
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 HelloApp
8 {
9     static class Program
10    {
11        [STAThread]
12        static void Main()
13        {
14            Application.EnableVisualStyles();
15            Application.SetCompatibleTextRenderingDefault(false);
16
17            Application.Run(new Form1());
18        }
19    }
```


- ▶ Пішіндердің негізгі қасиеттері
- ▶ Оң жақтағы арнайы Properties (сипаттар) терезесінің көмегімен Visual Studio бізге элементтің қасиеттерін басқаруға ыңғайлы интерфейсін ұсынады:



Бұл қасиеттердің көпшілігі пішінді визуалды түрде көрсетуге әсер етеді. Негізгі қасиеттері:

- ▶ **Name:** форма атауын орнатады - дәлірек, форма класынан мұраға қалған класс
- ▶ **BackColor:** пішіннің фондық түсін көрсетеді. Осы сипатты нұқу арқылы біз ұсынылған түстер тізімінен немесе түстер палитрасынан сәйкес келетін түсті таңдай аламыз
- ▶ **BackgroundImage:** пішіннің фондық кескінін көрсетеді
- ▶ **BackgroundImageLayout:** BackgroundImage сипатында көрсетілген кескін пішінде қалай орналасатынын анықтайды.
- ▶ **ControlBox:** пішін мәзірі көрсетілгенін көрсетеді. Бұл жағдайда ішкі мәзір қолданба белгішесі, пішін тақырыбы, сондай-ақ пішінді азайту түймелері мен крест орналасқан ең жоғарғы деңгейдің мәзірін білдіреді. Егер бұл қасиет жалған болса, онда біз әдетте пішін жабылатын белгішені де, крестті де көрмейміз

- ▶ **Cursor:** пішінде қолданылатын курсор түрін анықтайды
- ▶ **Enabled:** егер бұл сипат жалған болса, онда ол Пайдаланушыдан кіріс ала алмайды, яғни біз батырмаларды басып, мәтінді мәтін өрістеріне енгізе алмаймыз және т. б.
- ▶ **Font:** барлық пішінге және оған орналастырылған барлық басқару элементтеріне қаріп орнатады. Алайда, пішін элементтерінде қаріпті орнату арқылы біз оны қайта анықтай аламыз
- ▶ **ForeColor:** форумдағы қаріп түсі
- ▶ **FormBorderStyle:** пішін шекарасы мен тақырып жолы қалай көрсетілетінін көрсетеді. Бұл сипатты None-ға орнату арқылы сіз еркін пішіндегі қосымшаның көрінісін жасай аласыз
- ▶ **HelpButton:** анықтама формасы түймесі көрсетілгенін көрсетеді
- ▶ **Icon:** пішін белгішесін орнатады

- ▶ **Location:** егер бастау сипаты үшін Нұсқаулық орнатылған болса, экранның жоғарғы сол жақ бұрышына қатысты позицияны анықтайды
- ▶ **MaximizeBox:** пішін тақырыбындағы терезені барынша көбейту батырмасы бар-жоғын көрсетеді
- ▶ **MinimizeBox:** терезені азайту батырмасы бар-жоғын көрсетеді
- ▶ **Maximum Size:** пішіннің максималды мөлшерін анықтайды
- ▶ **MinimumSize:** ең аз пішін өлшемін орнатады
Ашықтық: форманың мөлдірлігін анықтайды
- ▶ **Size:** пішіннің бастапқы өлшемін анықтайды
- ▶ **StartPosition:** пішін экранда пайда болатын бастапқы позицияны көрсетеді
Мәтін: пішіннің тақырыбын анықтайды
- ▶ **TopMost:** егер бұл қасиет шын болса, онда пішін әрқашан басқа терезелердің үстінде болады
- ▶ **Visible:** пішін көрінеді ме, егер біз пішінді қолданушыдан жасырғымыз келсе, онда біз осы қасиетті жалған деп белгілей аламыз
- ▶ **WindowState:** іске қосу кезінде пішіннің қандай күйде болатындығын көрсетеді: қалыпты, максималды немесе минималды