



МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

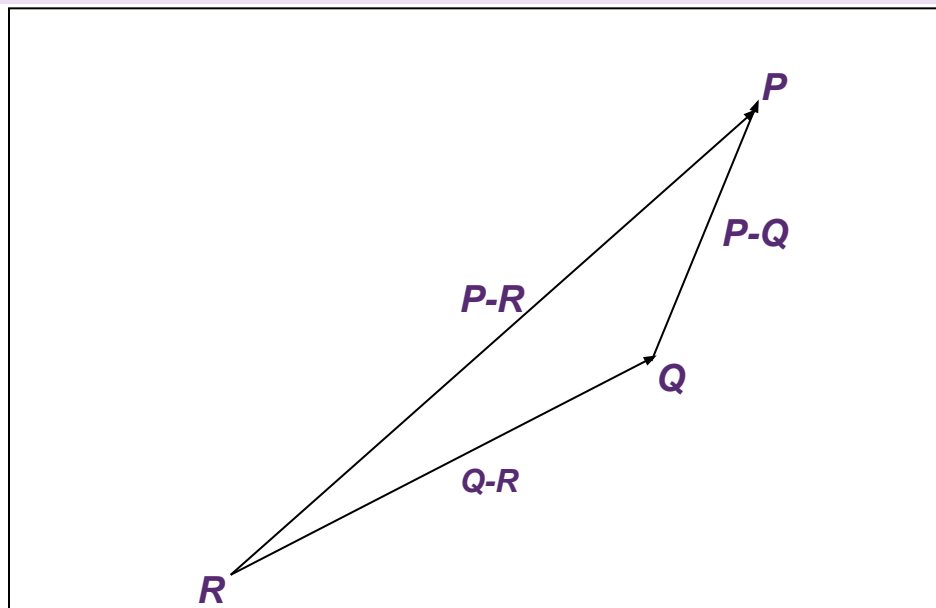
МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

- ▶ **Скаляр** – величина, каждое значение которой может быть выражено одним (действительным) числом.
- ▶ **Вектор** – это направленный отрезок прямой линии, характеризуемый только его длиной и направлением.

ВЕКТОРНОЕ И АФФИННОЕ ПРОСТРАНСТВО

- ▶ **Векторное** (линейное) пространство содержит объекты только двух типов: скаляры (действительные числа) и векторы.
- ▶ **Аффинное пространство** – это расширение векторного пространства, в которое включен дополнительный тип объектов – точка.

АФФИННОЕ ПРОСТРАНСТВО



Результатом операции сложения точки и вектора является точка.

Обратной ей является операция вычитания двух точек, результатом которой будет вектор.

Координатный метод

- ▶ каждая точка на экране (бумаге при печати на принтере) задается координатами;
- ▶ координаты используются для описания объектов, которые будут отображаться как пространственные;
- ▶ при выполнении многих промежуточных действий отображения используют разные системы координат и преобразования из одной системы в другую.

Системы координат

- ▶ Наиболее востребованными в компьютерной графике являются **декартовы системы координат**, как способ удобного создания абстракций реальных предметов окружающего мира.

Системы координат

- ▶ **Цилиндрические, сферические, проективные и различные другие координаты в пространстве обычно используются для моделирования специальных эффектов при деформации объектов.**

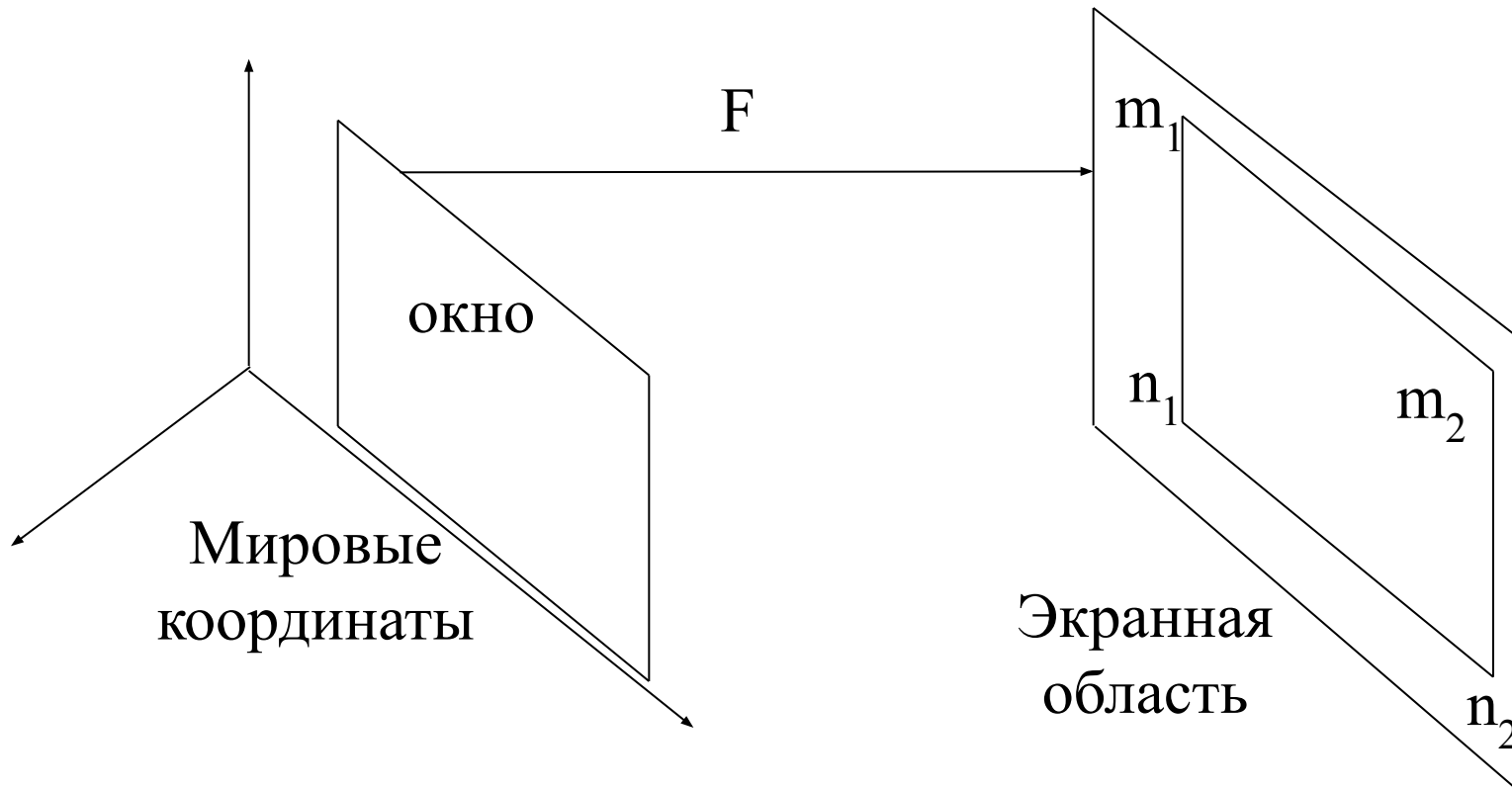
Системы координат

- ▶ При отображении пространственных объектов используют понятие **мировых координат** – трёхмерных декартовых координат пространства, в котором размещаются объекты.
- ▶ Каждый из объектов имеет собственную **объектную** систему координат.

Системы координат

- ▶ **Экранная** система координат связана с тем графическим устройством, где в заданной проекции на картинной плоскости отображается создаваемая трёхмерная сцена.

Системы координат



АФФИННАЯ СИСТЕМА КООРДИНАТ

- ▶ **Аффинная система координат** – система в n -мерном аффинном пространстве, определяемая совокупностью n линейно независимых векторов, исходящих из начала координат.

АФФИННАЯ СИСТЕМА КООРДИНАТ

- ▶ **Координаты точки в аффинной системе координат – это компоненты разложения радиус-вектора точки по координатным векторам.**
- ▶ **Число независимых координат, которые определяют ее положение, задает размерность пространства.**

АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

**Все изменения изображений можно
выполнить с помощью базовых операций:**

- ▶ поворот;**
- ▶ масштабирование;**
- ▶ отражение;**
- ▶ перенос (смещение).**

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

Предположим, что на плоскости введена декартова система координат (OXY) .

Тогда каждой точке M ставится в соответствие упорядоченная пара чисел (x, y) ее координат.

Вводя на плоскость еще одну декартову систему координат $(O'X'Y')$, ставим в соответствие той же точке M другую пару чисел (x', y') .

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

Переход от одной декартовой системы координат на плоскости к другой описывается следующими соотношениями:

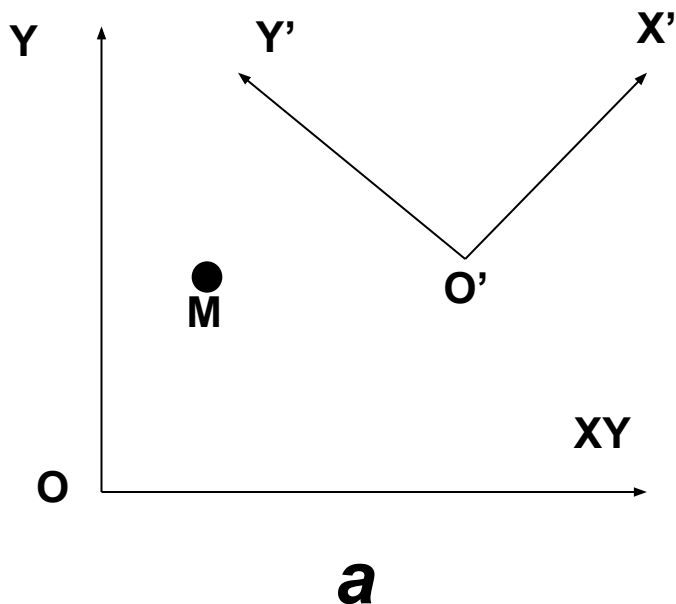
$$\begin{cases} x' = \alpha x + \beta y + \lambda, \\ y' = \gamma x + \delta y + \mu, \end{cases} \quad (*)$$

где $\alpha, \beta, \gamma, \lambda$ - произвольные числа, но $\begin{vmatrix} \alpha & \beta \\ \gamma & \delta \end{vmatrix} \neq 0$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

$$\begin{cases} x' = \alpha x + \beta y + \lambda, \\ y' = \gamma x + \delta y + \mu, \end{cases}$$

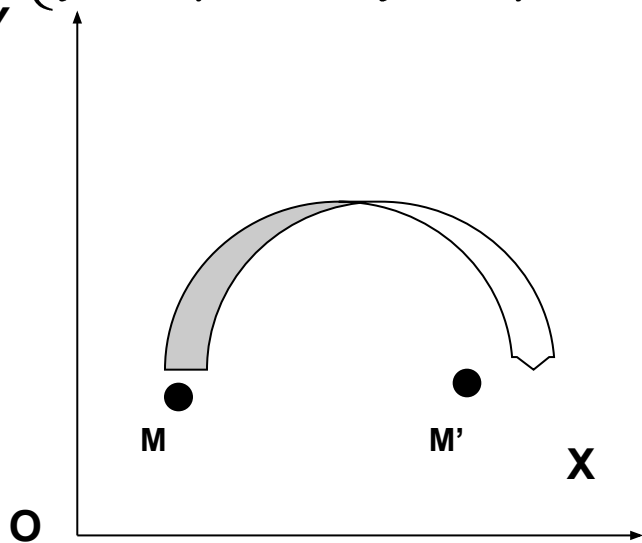
**Сохраняется точка и
изменяется
координатная
система –
произвольная точка
М остается той же,
изменяются лишь ее
координаты.**



2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

$$\begin{cases} x' = \alpha x + \beta y + \lambda, \\ y' = \gamma x + \delta y + \mu, \end{cases}$$

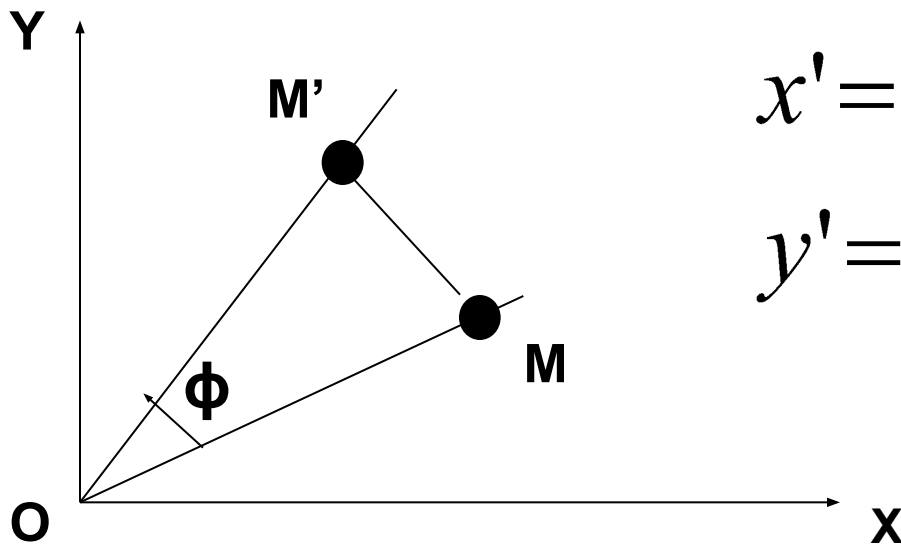
Изменяется точка, и сохраняется координатная система – формула задает отображение, переводящее произвольную точку $M(x, y)$ в точку $M(x', y')$, координаты которой определены в той же координатной системе.



В дальнейшем будем рассматривать второй случай преобразования, соответствующий рис. б.

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

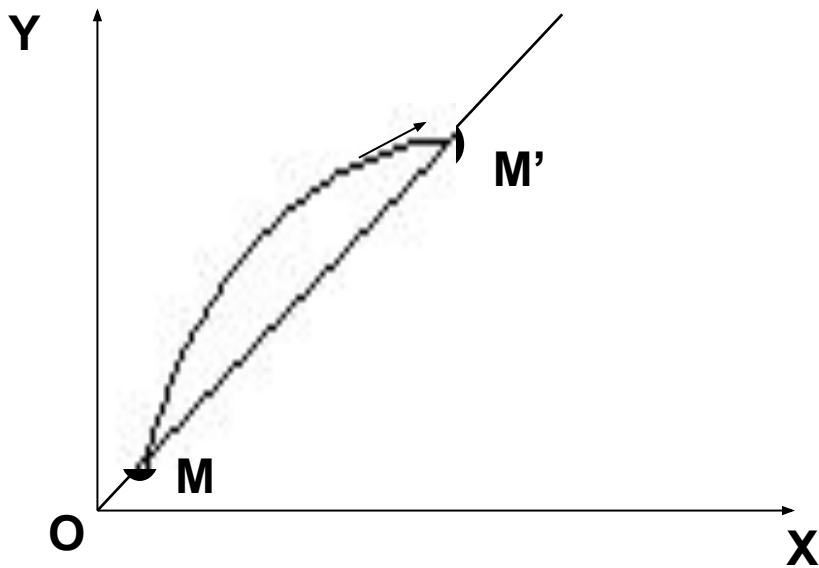
- ▶ I. Поворот (вокруг начальной точки на угол ϕ) описывается формулами:



$$\begin{aligned}x' &= x \cos(\varphi) - y \sin(\varphi), \\y' &= x \sin(\varphi) + y \cos(\varphi).\end{aligned}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

II. Растяжение (сжатие) вдоль координатных осей:



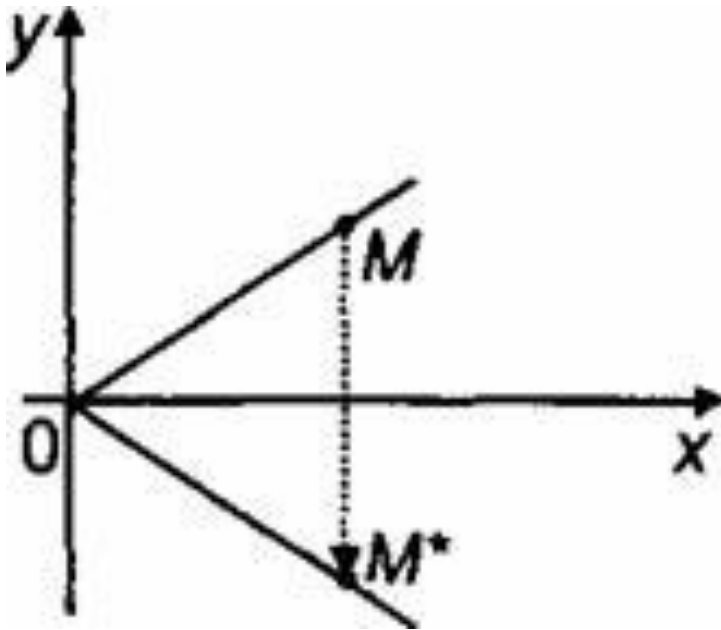
$$x' = \alpha x,$$

$$y' = \delta y.$$

$$\alpha > 0, \delta > 0.$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

- ▶ III. Отражение относительно оси абсцисс задается при помощи формул:

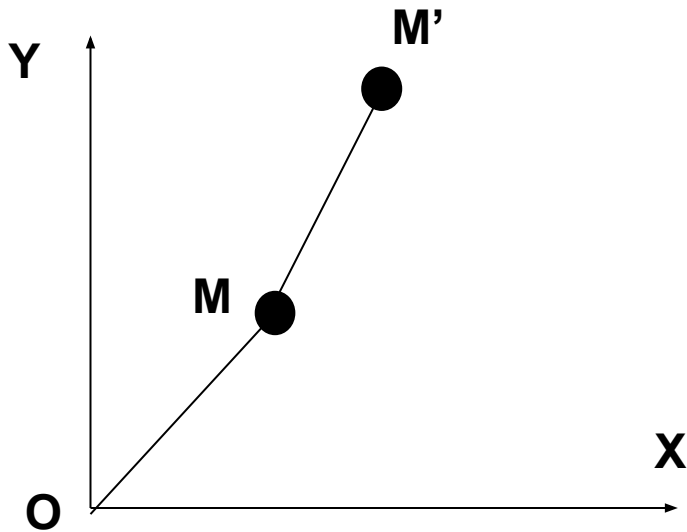


$$x' = x,$$

$$y' = -y.$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

- ▶ IV. Перенос обеспечивают соотношения:



$$x' = x + \lambda,$$
$$y' = y + \mu.$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

Выбор этих частных случаев определяется двумя обстоятельствами:

- ▶ **каждое из приведенных выше преобразований имеет простой и наглядный геометрический смысл;**
- ▶ **как доказывается в курсе аналитической геометрии, любое преобразование всегда можно представить как последовательное исполнение (суперпозицию) простейших преобразований I-IV.**

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

Для применения перечисленных выше преобразований в задачах КГ применяется их матричная запись:

$$\begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}, \begin{bmatrix} \alpha & 0 \\ 0 & \delta \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

- ▶ **Преобразование переноса (IV) в матричном виде записать невозможно.**
- ▶ **Для простоты алгоритмизации любого преобразования необходимо иметь матричный вид записи всех четырех преобразований .**

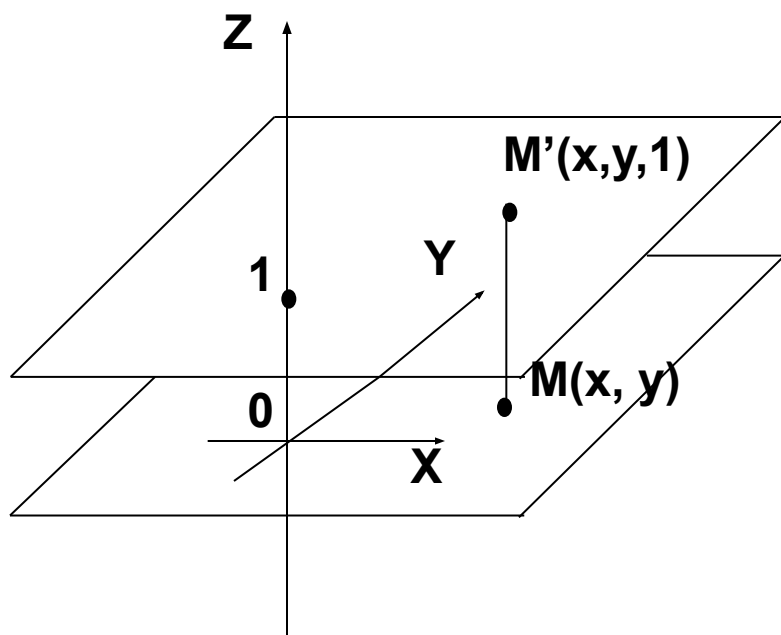
ОДНОРОДНЫЕ КООРДИНАТЫ

Однородными координатами точки $M(x, y)$ называется тройка одновременно не равных нулю

x_1, x_2, x_3 **чисел**, если:

$$x = \frac{x_1}{x_3}, \quad y = \frac{x_2}{x_3}.$$

ОДНОРОДНЫЕ КООРДИНАТЫ



Точке $M(x, y)$ ставится в соответствие точка $M'(x, y, 1)$ в пространстве.

Произвольная точка на прямой, соединяющей начало координат, точку $O(0,0,0)$, с точкой $M'(x, y, 1)$, может быть задана тройкой (hx, hy, h) .

Исключая точку O из рассмотрения, будем считать, что $h \neq 0$.

ОДНОРОДНЫЕ КООРДИНАТЫ

Считая $h=1$ выражение (*) можно переписать в виде:

$$(x', y', 1) = (x, y, 1) \cdot \begin{bmatrix} \alpha & \gamma & 0 \\ \beta & \delta & 0 \\ \lambda & \mu & 1 \end{bmatrix}$$

$$\text{или} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & 0 \\ \beta & \delta & 0 \\ \lambda & \mu & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

I. Матрица вращения (rotation):

$$[R] = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

II. Матрица масштабирования:

$$[D] = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

III. Матрица отражения:

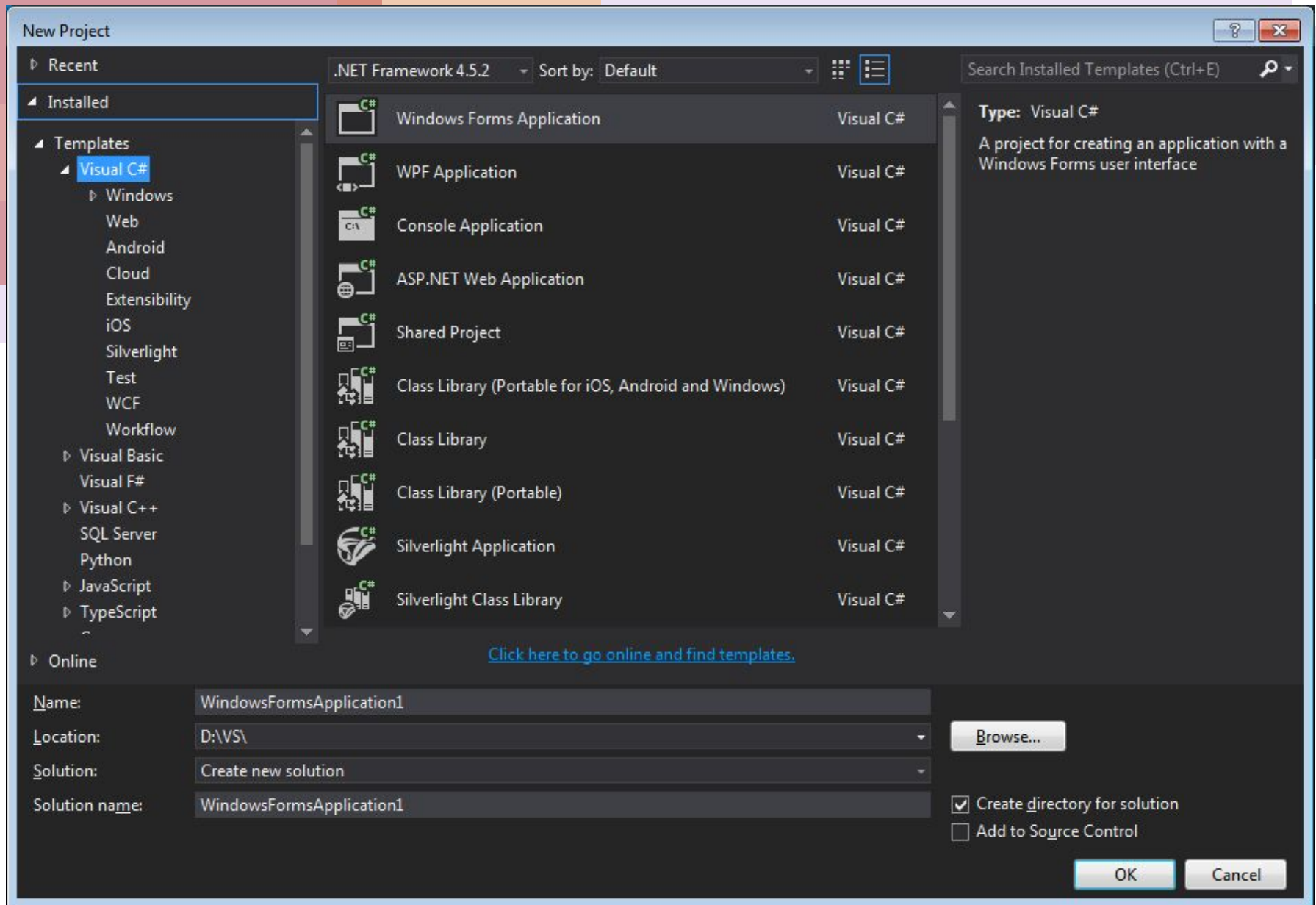
$$[M] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2D АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

IV. Матрица переноса:

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \lambda & \mu & 1 \end{bmatrix}$$

Visual Studio, C#

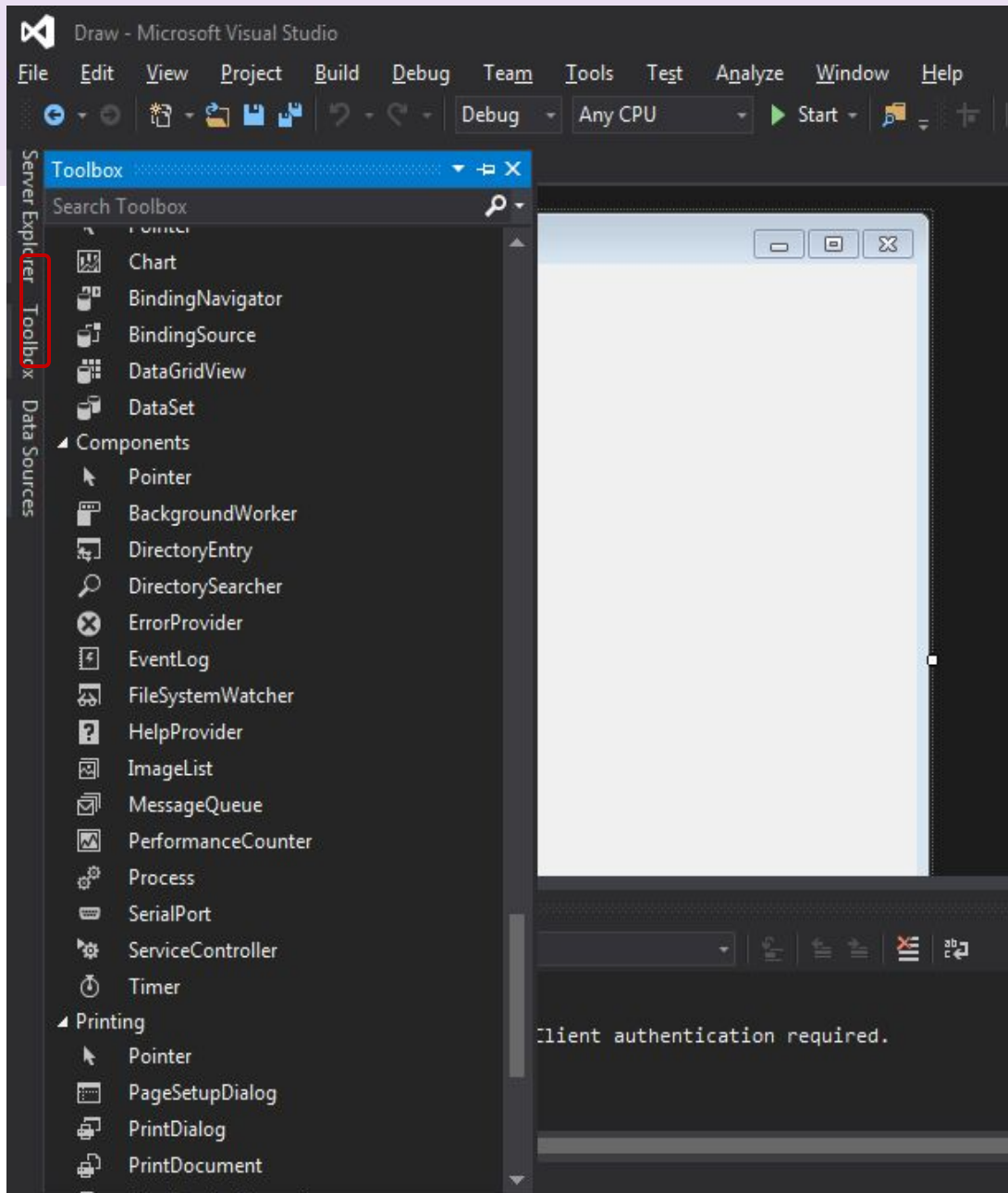
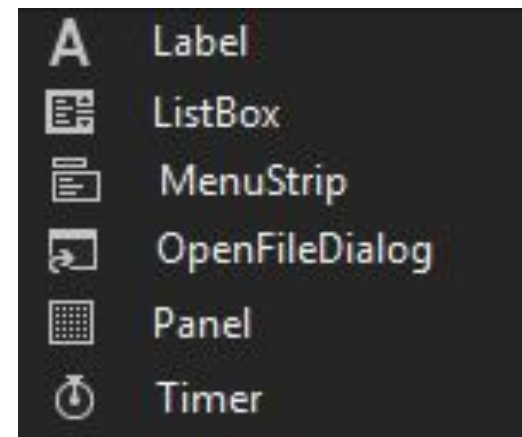
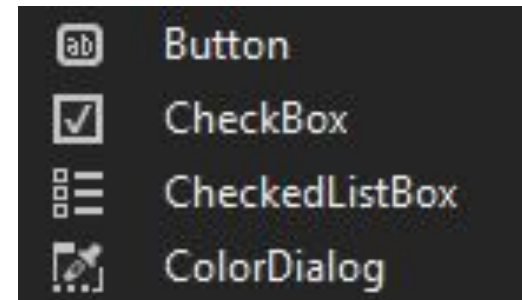


Visual Studio окно элементов

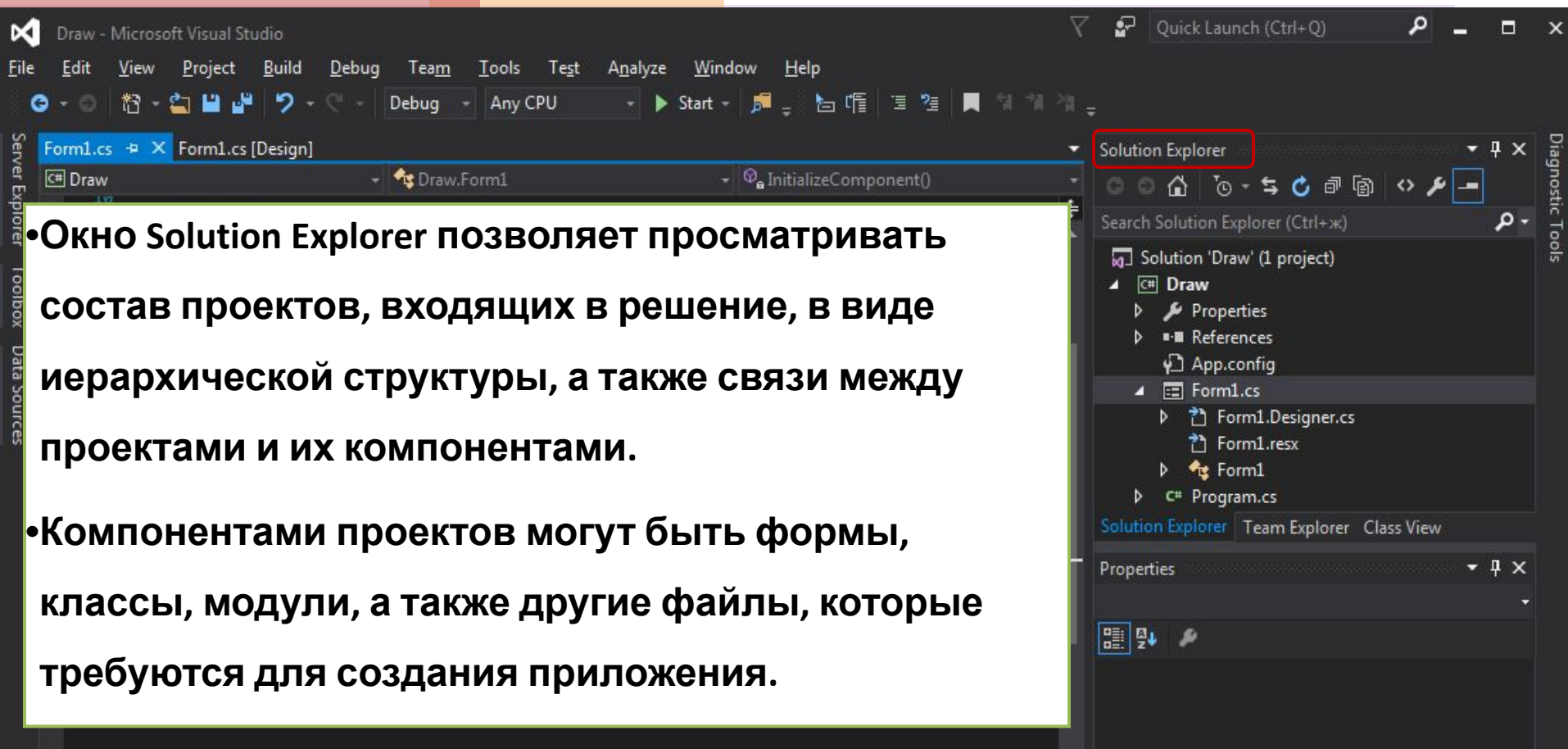
Toolbox (Ctrl+Alt+X)

Содержит список элементов управления, которые можно использовать на формах приложения.

Мы будем использовать:



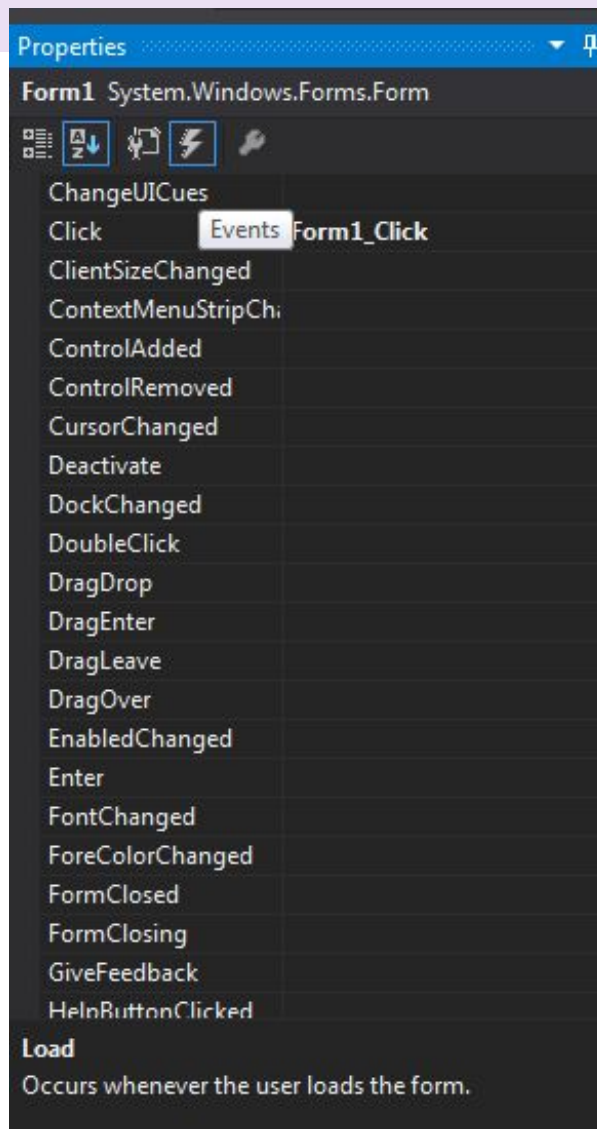
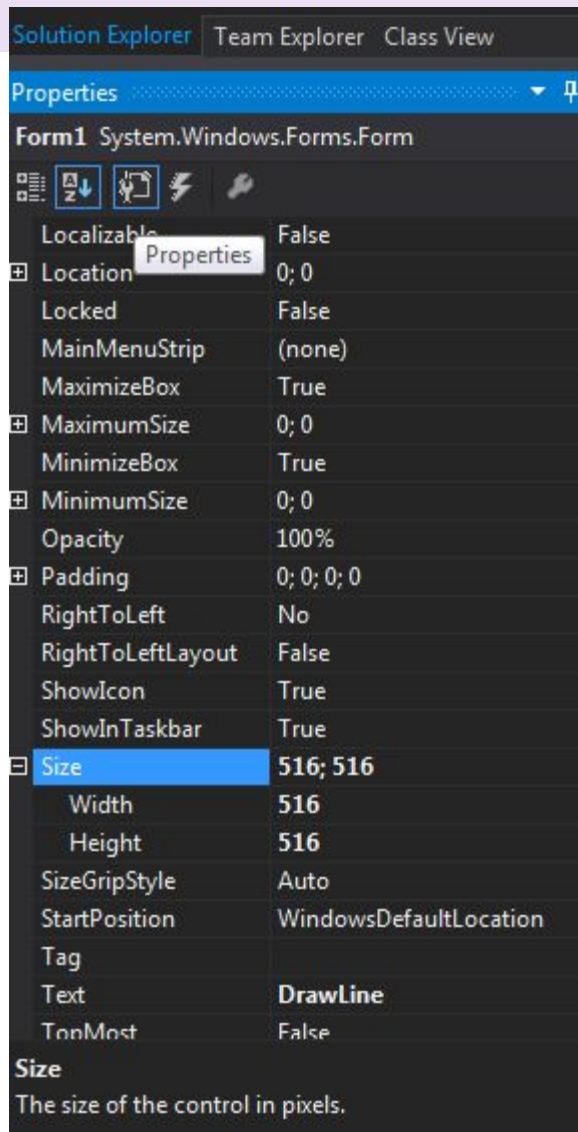
Окно Solution Explorer



Form1.cs [Design]

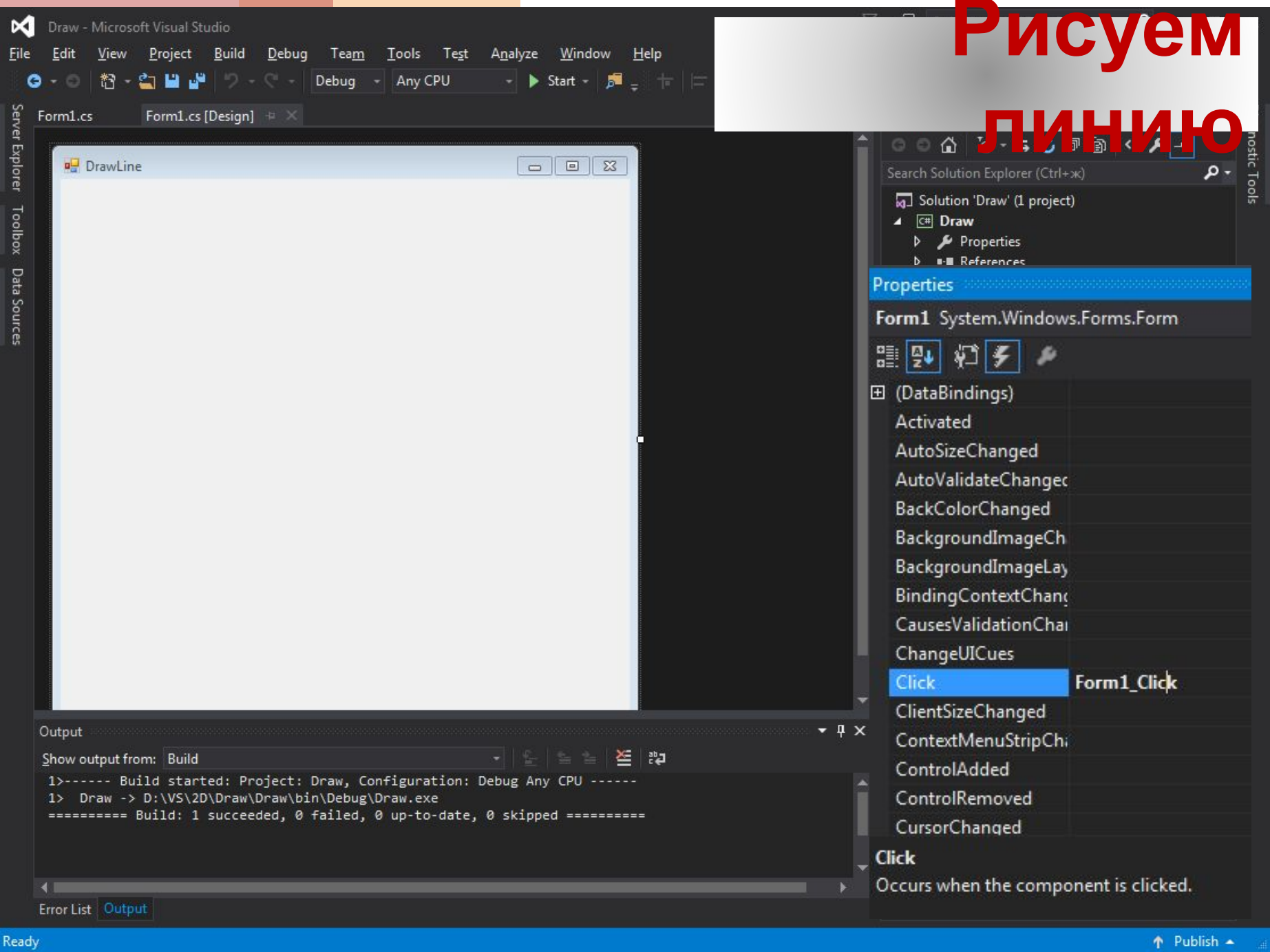
- Окно Solution Explorer позволяет просматривать состав проектов, входящих в решение, в виде иерархической структуры, а также связи между проектами и их компонентами.
- Компонентами проектов могут быть формы, классы, модули, а также другие файлы, которые требуются для создания приложения.

Окно Properties (F4)



**Предназначено
для изменения
свойств
элементов
управления
создаваемого
приложения**

Рисуем линию



Рисуем линию

```
C# Draw Draw.Form1
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Draw
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void Form1_Click(object sender, EventArgs e)
21         {
22
23         }
24     }
25 }
26
```

Перед тем как рисовать линии и фигуры, отображать текст, выводить изображения и управлять ими в GDI+ необходимо создать объект **Graphics**.

Объект **Graphics** представляет поверхность рисования GDI+ и используется для создания графических изображений.

Работа с графикой

- ▶ **Этап 1:** Создание объекта **Graphics**. Вывозите метод `CreateGraphics` формы или элемента управления, на котором необходимо отобразить графику:

`Graphics g = this.CreateGraphics();`

- ▶ **Этап 2.** Использование объекта **Graphics** для рисования линий и фигур, отображения текста или изображения и управления ими. После создания объекта `Graphics` его можно использовать для рисования линий и фигур, отображения текста или изображения и управления ими. Основные объекты, используемые с объектом `Graphics`:

Класс Pen — служит для рисования линий, контуров и отрисовки других геометрических объектов.

Класс Brush — служит для заливки областей, например фигур, изображений или текста.

Рисуем линию

Solution Explorer Team Explorer Class View

Properties ▾

Form1 System.Windows.Forms.Form

Localizable False

Location 0; 0

Locked False

MainMenuStrip (none)

MaximizeBox True

MaximumSize 0; 0

MinimizeBox True

MinimumSize 0; 0

Opacity 100%

Padding 0; 0; 0; 0

RightToLeft No

RightToLeftLayout False

ShowIcon True

ShowInTaskbar True

Size 516; 516

Width 516

Height 516

SizeGripStyle Auto

StartPosition WindowsDefaultLocation

Tag

Text **DrawLine**

TopMost False

Size

The size of the control in pixels.

```
_Click(object sender, EventArgs e)
```

```
this.CreateGraphics();
```

```
White);
```

```
Pen(Color.Red);
```

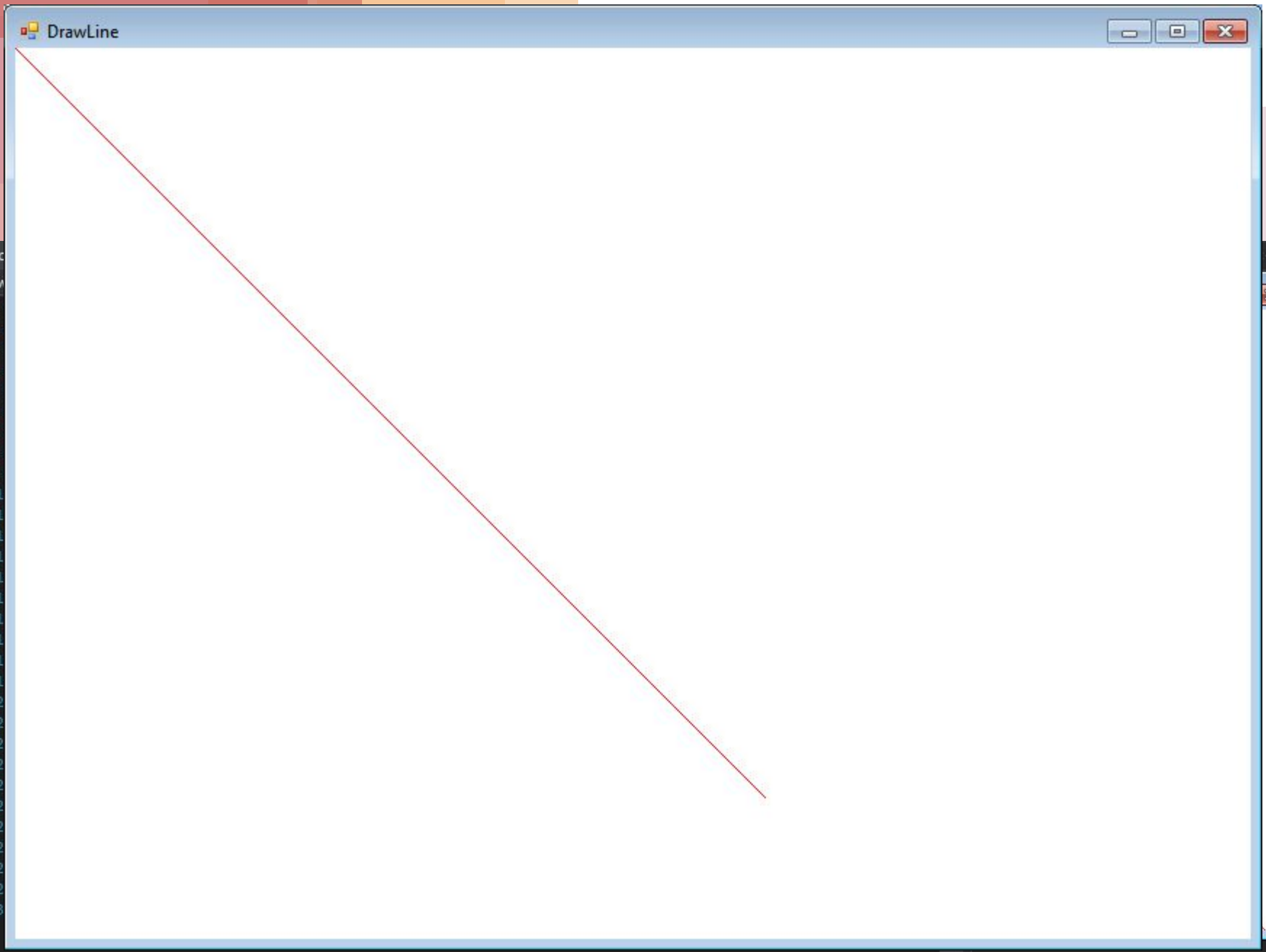
```
, 0, 0, 516, 516);
```

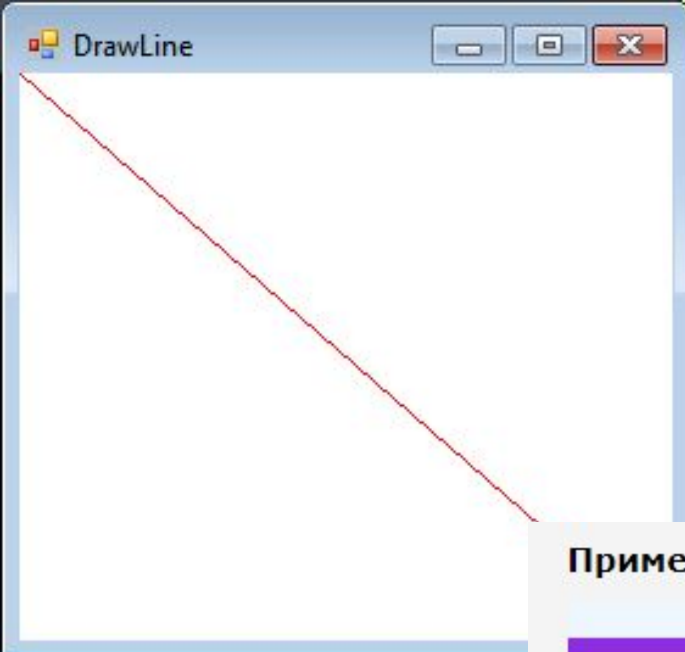
```
void Graphics.DrawLine(Pen pen, int x1, int y1, int x2, int y2) (+ 3 overloads)
```

Draws a line connecting the two points specified by the coordinate pairs.

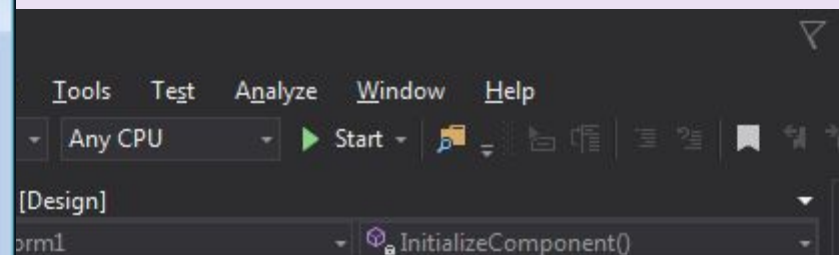
Exceptions:

ArgumentNullException





Рисуем линию



```

15 public Fo
16 {
17     Init
18 }
19
20 private v
21 {
22     Graph
23     g.Cle
24     // P
25     Pen p
26     g.Dra
27     g.Dis
28 }
29 }
30 }
31

```

Пример	X11 Обозначение цвета	HEX	RGB
	AliceBlue	#F0F8FF	240,248,255
	BlueViolet	#8A2BE2	138,43,226
	CadetBlue	#5F9EA0	95,158,160
	CadetBlue1	#98F5FF	152,245,255
	CadetBlue2	#8EE5EE	142,229,238
	CadetBlue3	#7AC5CD	122,197,205
	CadetBlue4	#53868B	83,134,139
	CornflowerBlue	#6495ED	100,149,237
	DarkBlue	#00008B	0,0,139
	DarkCyan	#008B8B	0,139,139
	DarkSlateBlue	#483D8B	72,61,139
	DarkTurquoise	#00CED1	0,206,209
	DeepSkyBlue	#00BFFF	0,191,255
	DeepSkyBlue1	#00BFFF	0,191,255
	DeepSkyBlue2	#00B2EE	0,178,238
	DeepSkyBlue3	#009ACD	0,154,205
	DeepSkyBlue4	#00688B	0,104,139
	DodgerBlue	#1E90FF	30,144,255
	DodgerBlue1	#1E90FF	30,144,255
	DodgerBlue2	#1C86EE	28,134,238
	DodgerBlue3	#1874CD	24,116,205
	DodgerBlue4	#104E8B	16,78,139

Рисуем линию

