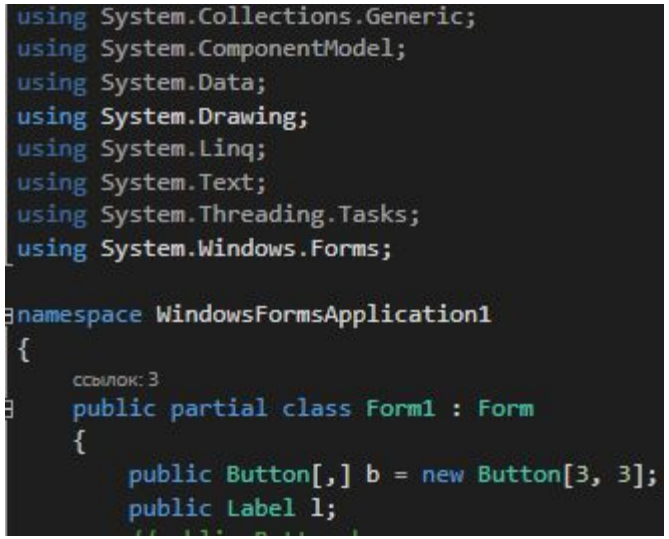
Динамическое создание объектов

Введение

- Довольно часто возникают такие ситуации, когда банально неудобно сидеть и прописывать множество однотипных объектов. Иной раз их проще создать динамически – то есть описать полностью внутри кода без использования конструктора.
- Отличным примером программы, для которой подойдет данный способ, являются крестики-нолики. У нас девять однотипных объектов, при нажатии на которые у нас появляется либо крестик, либо нолик соответственно.

Динамическое создание объекта

- Динамически можно добавить абсолютно любой объект. Просто выберите тот, который вам нужен, и пропишете его добавление в самом начале кода! Всегда используйте сначала ключевое слово public. Обратите внимание, вы можете добавить как один объект, так и целый массив объектов!



Динамическое создание объекта

- На предыдущем слайде вы просто сказали, что у вас будет такой объект, но не создали. Создание объекта всегда помечается словом new и использованием конструктора. Иначе говоря, создание объекта выглядит следующим образом:
- Для одного объекта

```
l = new Label();
```

- Для массива объектов (создание производится строго в цикле!!!):

```
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        b[i, j] = new Button();
```

Динамическое создание объектов

- Дальше вы можете менять свойства так, как вам то будет нужно. После того, как вы сделали все, что вам нужно, вы должны поместить объект на форму!!! Это делается с использованием конструкции следующего вида:

this.Controls.Add(*объект*);

- Например, добавление на форму одного объекта:

```
this.Controls.Add(l);
```

- Массива объектов (все еще в цикле!!!):

```
this.Controls.Add(b[i, j]);
```

События

- Событие – это некоторая функция или процедуры, которая вызывается при каких-то условиях. Конструкция для событий выглядит следующим образом:

public *тип название_события*(object sender, *аргумент*)

- Аргумент может различаться в зависимости от того, для какого объекта и какое событие мы создаем. Например, мы создаем событие, которые будет вызывать по клику, для кнопки. Оно будет выглядеть следующим образом:

```
public void b_Click(object sender, EventArgs e)
```

События

- Просто создать событие – недостаточно. Его нужно связать с нашим объектом. Связь осуществляется также, как и изменение свойств объектов, т.к. через точку. Заметим, что каждая такая связь уникальна и она отличается от объекта к объекту и от события к событию. Например, связь для кнопки для события «клик» будет выглядеть следующим образом:

```
b[i, j].Click += new EventHandler(b_Click);
```

Задача

- Реализовать заполнение полей в игре «крестики-нолики».

Принцип

- Все объекты будут задаваться динамически. Первыми ходят крестики. Динамически нужно будет создать девять кнопок и девять пустых изображений под ними. На эти пустые изображения будут помещаться либо крестик, либо нолик в зависимости от хода.
- Кнопки и изображения будут содержаться в двумерном массиве, где индексы будут отвечать за строки и столбцы соответственно.

Инициализация объектов

```
public Button[,] but = new Button[3, 3]; //создание массива кнопок
public PictureBox[,] pic = new PictureBox[3, 3]; //создание массива изображений
int[,] a = new int[3, 3]; //числовой массив, отвечающий за отображаемый элемент - крестик или нолик
int c = 1; //счетчик ходов, нечетные ходы - крестики, четные - нолики
```

Создание объектов

```
public Form1()
{
    InitializeComponent();
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            a[i, j] = -1; //заполняем массив -1
            //1 - крестик, 0 - нолик
            but[i, j] = new Button(); //создание кнопки
            pic[i, j] = new PictureBox(); //создание изображения
            //задаем одинаковый размер для кнопки и для изображения
            but[i, j].Size = new Size(100, 100);
            pic[i, j].Size = new Size(100, 100);
            //задаем одинаковое положение для кнопки и изображения
            but[i, j].Location = new Point(this.Location.X + 100 * i, this.Location.Y + 100 * j);
            pic[i, j].Location = new Point(this.Location.X + 100 * i, this.Location.Y + 100 * j);
            but[i, j].Click += new EventHandler(play); //привязываем событие хода к кнопке
            //помещаем на форму сначала кнопку, а потом картинку, чтобы картинка была скрыта под кнопкой
            this.Controls.Add(but[i, j]);
            this.Controls.Add(pic[i, j]);
        }
    }
}
```

Создание события

```
public void play(object sender, EventArgs e)
{
    Button btn = (Button)sender; //получаем информацию о том, какая кнопка была нажата
    //получаем информацию о ее номере в массиве
    int i = btn.Location.X / 100;
    int j = btn.Location.Y / 100;
    but[i, j].Visible = false; //скрываем кнопку
    a[i, j] = c % 2; //берем остаток от деление на 2
    //нечетное - крестик
    //четное - нолик
    pic[i, j].Image = Image.FromFile(a[i, j].ToString() + ".png"); //на место изображения вставляем нужную картинку
    //1 - крестик
    //0 - нолик
    c++; //увеличиваем счетчик ходов
}
```

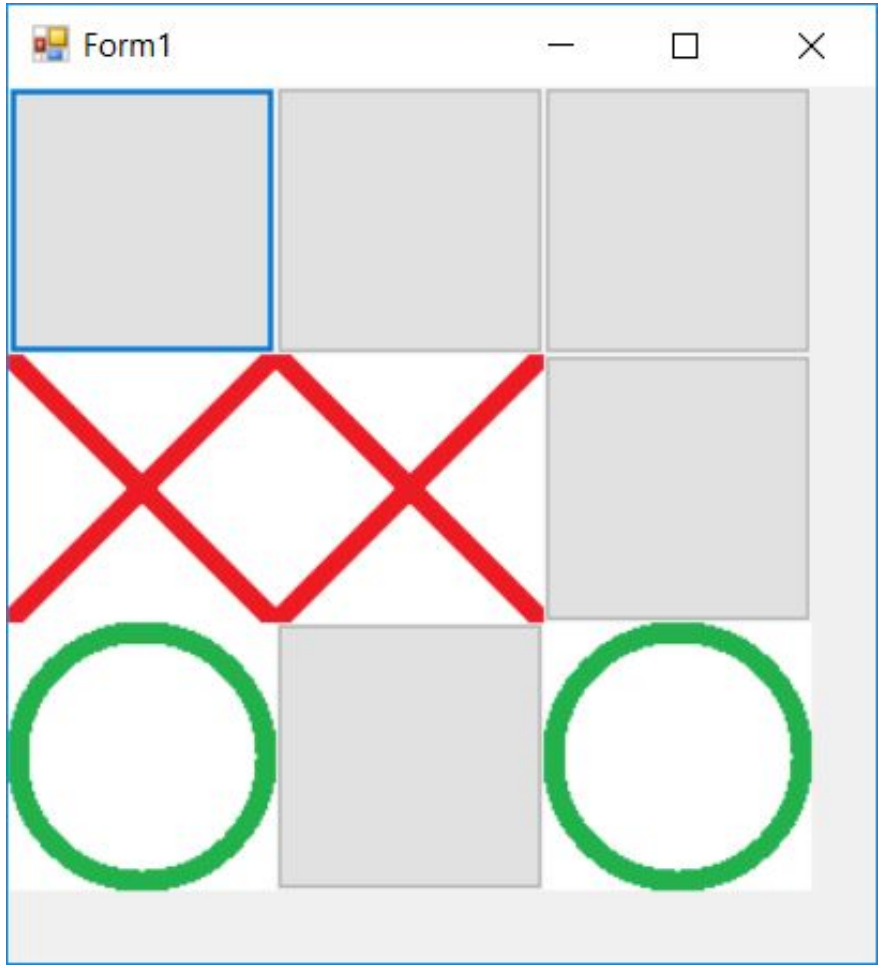
Код

```
public partial class Form1 : Form
{
    public Button[,] but = new Button[3, 3]; //создание массива кнопок
    public PictureBox[,] pic = new PictureBox[3, 3]; //создание массива изображений
    int[,] a = new int[3, 3]; //числовой массив, отвечающий за отображаемый элемент - крестик или нолик
    int c = 1; //счетчик ходов, нечетные ходы - крестики, четные - нолики

    ссылка: 1
    public Form1()
    {
        InitializeComponent();
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                a[i, j] = -1; //заполняем массив -1
                //1 - крестик, 0 - нолик
                but[i, j] = new Button(); //создание кнопки
                pic[i, j] = new PictureBox(); //создание изображения
                //задаем одинаковый размер для кнопки и для изображения
                but[i, j].Size = new Size(100, 100);
                pic[i, j].Size = new Size(100, 100);
                //задаем одинаковое положение для кнопки и изображения
                but[i, j].Location = new Point(this.Location.X + 100 * i, this.Location.Y + 100 * j);
                pic[i, j].Location = new Point(this.Location.X + 100 * i, this.Location.Y + 100 * j);
                but[i, j].Click += new EventHandler(play); //привязываем событие хода к кнопке
                //помещаем на форму сначала кнопку, а потом картинку, чтобы картинка была скрыта под кнопкой
                this.Controls.Add(but[i, j]);
                this.Controls.Add(pic[i, j]);
            }
        }
    }

    ссылка: 1
    public void play(object sender, EventArgs e)
    {
        Button btn = (Button)sender; //получаем информацию о том, какая кнопка была нажата
        //получаем информацию о ее номере в массиве
        int i = btn.Location.X / 100;
        int j = btn.Location.Y / 100;
        but[i, j].Visible = false; //скрываем кнопку
        a[i, j] = c % 2; //берем остаток от деление на 2
        //нечетное - крестик
        //четное - нолик
        pic[i, j].Image = Image.FromFile(a[i, j].ToString() + ".png"); //на место изображения вставляем нужную картинку
        //1 - крестик
        //0 - нолик
        c++; //увеличиваем счетчик ходов
    }
```

Результат работы



Задача

- Закончить игру.