Windows’s windows

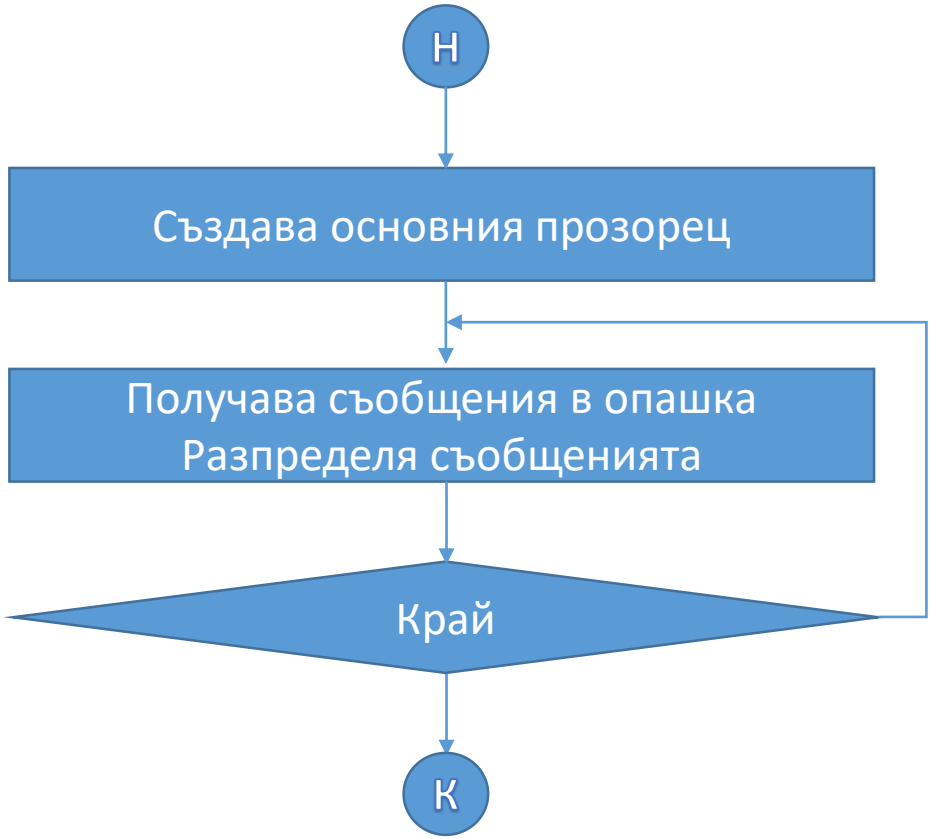
- Основния потребителски обект с който работи Windows е обекта прозорец

- Всяка нишка отговаряща за потребителския интерфейс притежава прозорците създадени от нея и отговаря за тяхната обработка. За целта тя притежава опашка за съобщения и цикъл за обработката им

Message loop (.c)

```
HINSTANCE hinst;
HWND hwndMain;

int PASCAL WinMain(
    HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine,
    int nCmdShow)
{
    MSG msg;
    BOOL bRet;
    WNDCLASS wc;
    UNREFERENCED_PARAMETER(lpszCmdLine);

    // Register the window class for the main window.
    //...

    // Create the main window.
    hwndMain = CreateWindow(
        "MainWndClass",
        "Sample",
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        (HWND)NULL,
        (HMENU)NULL,
        hinst,
        (LPVOID)NULL);

    //..
    // Show the window and paint its contents.
    ShowWindow(hwndMain, nCmdShow);
    UpdateWindow(hwndMain);

    // Start the message loop.
    while ((bRet = GetMessage(&msg, NULL, 0, 0)) != 0)
    {
        if (bRet == -1)
        {
            // handle the error and possibly exit
        }
        else
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    // Return the exit code to the system.
    return msg.wParam;
}
```

Message loop (.c)



Прозорец

- При потребителска активност (работа с мишка, клавиатура и тн.) операционната система изпраща съобщения адресирани към прозорец

- Всеки прозорец е отговорен за обработка на съобщенията към него

Потребителски контроли

- Всеки прозорец може да изрисува изгледа си според предназначението си

- Повечето потребителски контроли (бутони, списъци, панели и тн.) са прозорци разположени едни върху други с предефинирана обработка на част от съобщенията към тях

Windows Forms (.NET)

- В .NET и респективно C# опашката за съобщения и цикъла за разпределение и обработката им са скрити в йерархии от класове

- Съобщенията към контролите (прозорците) пораждат извикването на събития на тези класове (като OnClick ит.)

Windows Forms (.NET)

- Всеки прозорец има подразбираща се имплементация на метода отговарящ за изрисуването му, като тя може да бъде променена при необходимост

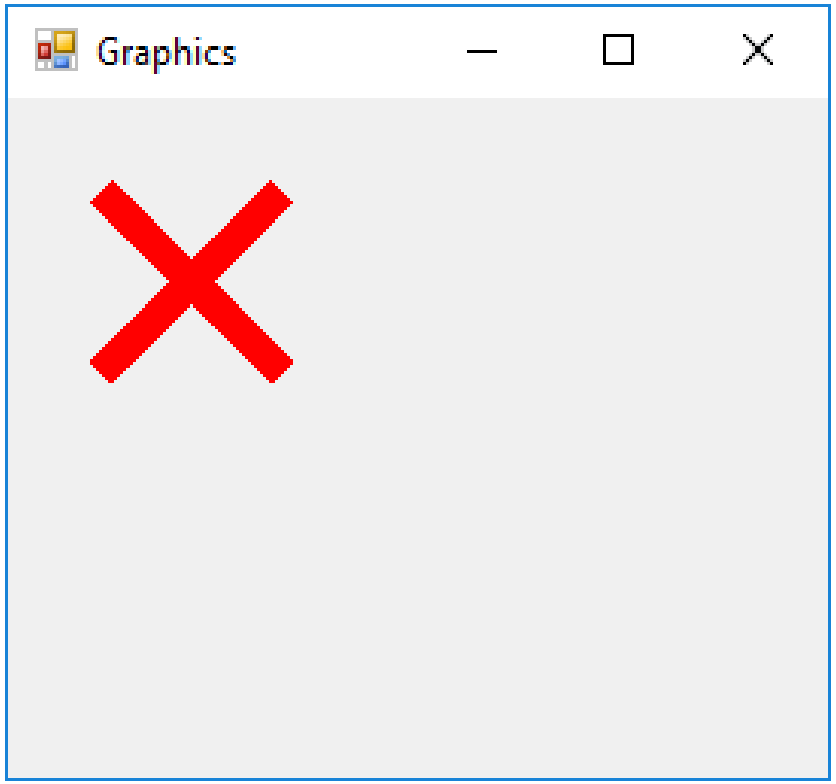
OnPaint

```
namespace System.Windows.Forms
{
    ...public class Control : Component, IDropTarget, ...
    {
        ...public Control();
        //...
        ...protected virtual void OnPaint(PaintEventArgs e);
        //...
```

OnPaint

```
namespace System.Windows.Forms
{
    ...public class Form : ContainerControl
    {
        ...public Form();

        //...

        ...protected override void OnPaint(PaintEventArgs e);

        //...
```

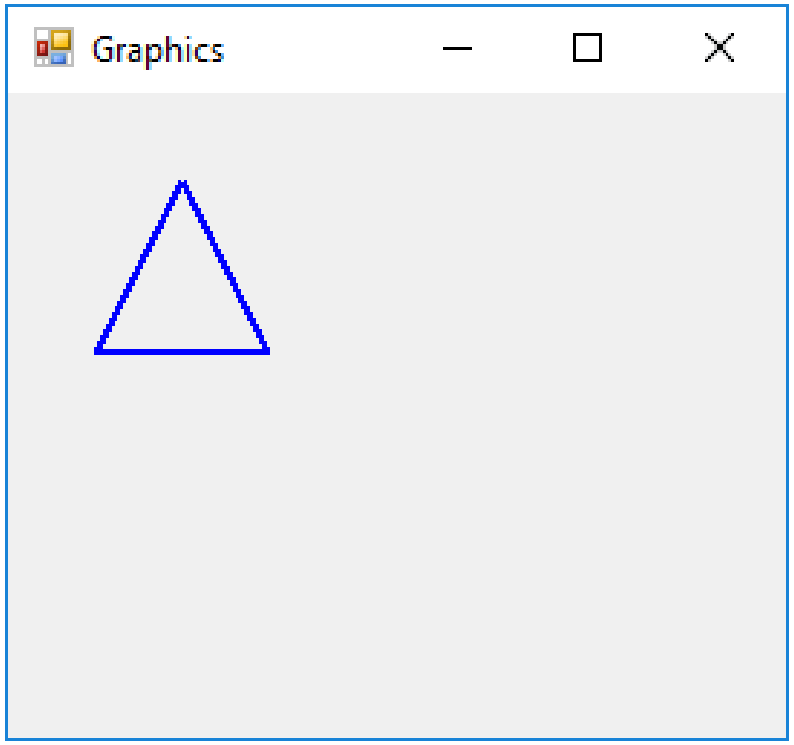
OnPaint



OnPaint

```
public partial class FormMain : Form
{
    public FormMain()
    {
        InitializeComponent();
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);

        Graphics graphics = e.Graphics;

        Pen pen = new Pen(Color.Red, 10);
        graphics.DrawLine(pen, 30, 30, 90, 90);
        graphics.DrawLine(pen, 90, 30, 30, 90);
    }
}
```

Да се нарисува триъгълник



Да се нарисува триъгълник

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);

    Graphics graphics = e.Graphics;

    Pen pen = new Pen(Color.Blue, 2);
    graphics.DrawLine(pen, 60, 30, 90, 90);
    graphics.DrawLine(pen, 60, 30, 30, 90);
    graphics.DrawLine(pen, 30, 90, 90, 90);
}
```

Да се направи клас Rectangle

- Класът да има:
  - Свойство от Point, за позиция
  - Свойства X и Y за дължина на страните
  - Метод Paint с един параметър от тип Graphics и имплементация използваща метода DrawRectangle на параметъра

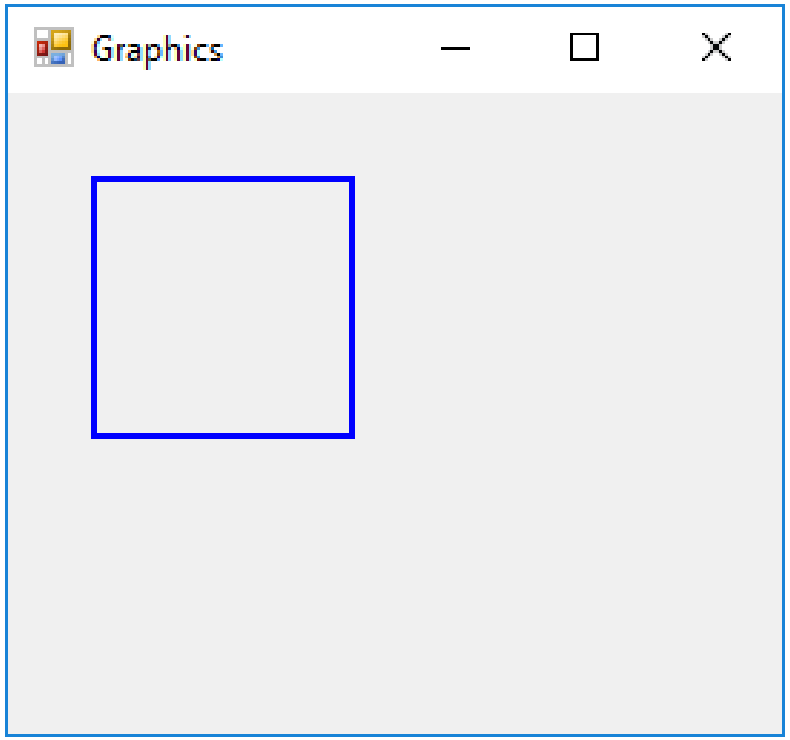
Клас Rectangle

```
public class Rectangle
{

    public Point Position { get; set; }

    public int Width { get; set; }

    public int Height { get; set; }

    public void Paint(Graphics graphics)
    {
        Pen pen = new Pen(Color.Blue, 2);

        graphics.DrawRectangle(pen, Position.X, Position.Y, Width, Height);
    }
}
```

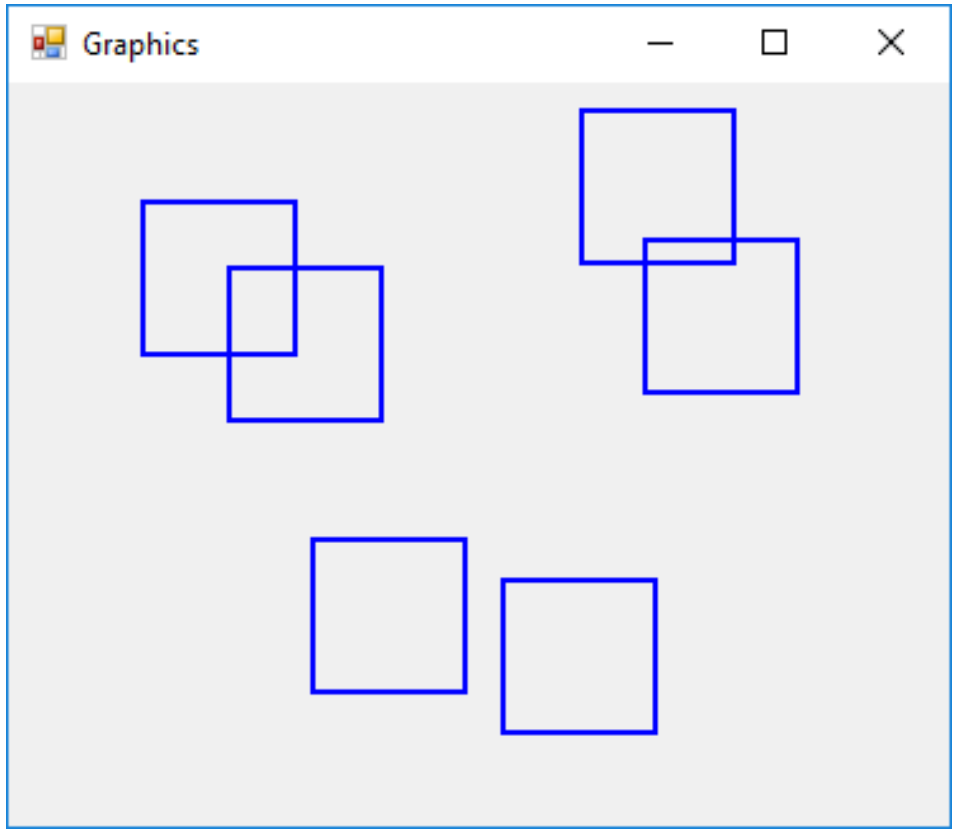
Използване на класа Rectangle

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);

    Rectangle rectangle = new Rectangle();
    rectangle.Position = new Point(30, 30);
    rectangle.Width = 60;
    rectangle.Height = 60;

    rectangle.Paint(e.Graphics);
}
```

Използване на класа Rectangle



Да се изрисува четириъгълник при натискане на ляв бутон на мишката



Да се изрисува четириъгълник при натискане на ляв бутон на мишката

- Изисквания:
  - Да се добави шаблонен списък (Generics) в класа на формата, в който ще се съхраняват четириъгълниците;
  - Да се прихване събитието MouseDown на формата (параметъра „e“ има свойство Location отговарящо на точката в която е била мишката). В събитието се създава нова инстанция на Rectangle и се добавя в списъка. В събитието се изрисува новия четириъгълник, като Graphics обекта се получава чрез функцията CreateGraphics() на формата;
  - Да се изрисуват всички фигури в метода OnPaint.

Да се изрисува четириъгълник при натискане на ляв бутон на мишката

```
public partial class FormMain : Form
{
    private List<Rectangle> _rectangles = new List<Rectangle>();

    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);

        foreach (var rectangle in _rectangles)
        {
            rectangle.Paint(e.Graphics);
        }
    }
```

```
private void FormMain_MouseDown(object sender, MouseEventArgs e)
{
    Rectangle rectangle = new Rectangle();
    rectangle.Position = e.Location;
    rectangle.Width = 60;
    rectangle.Height = 60;

    _rectangles.Add(rectangle);

    using (var graphics = this.CreateGraphics())
    {
        rectangle.Paint(graphics);
    }
}
```