# Osnove programskog jezika C#

Čas 4.

Dizajn kontrola, događaji

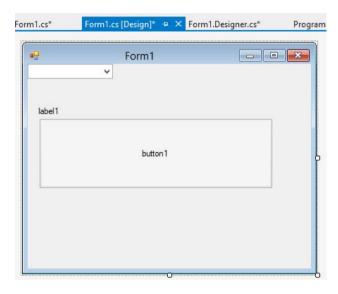
# Windows aplikacije

- Podsećanje:
- 1. Kako od konzolne aplikacije do Windows aplikacije?
  - Nova referenca u projektu za korscenje Windows.Forms biblioteke
- 2. Korišćenje projekta tipa Windows application

## Nasleđivanje na delu

```
musing System;
 using System.Collections.Generic;
 using System.ComponentModel;
 using System.Data;
 using System.Drawing;
 using System.Text;
using System.Windows.Forms;
□ namespace MyWindowsApp
     public partial class Form1 : Form
         public Form1()
              InitializeComponent();
```

# (i) Faza dizajna i kodovanja(ii) Faza izvršavanja (F5, Ctrl+F5)



 Izmene u dizajnu se ogledaju u metodi InitializeComponent koja se nalazi u fajlu Form1.Designer.cs.  Svoj kod u vezi klase Form1 unosite u fajl Form1.cs.

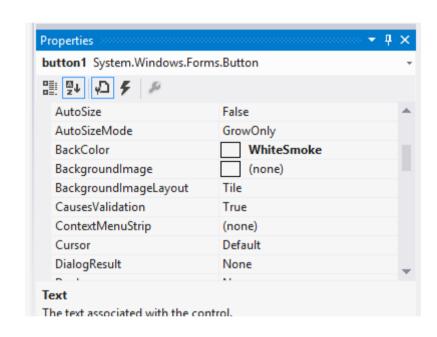
```
Form1.cs* * X Form1.cs [Design]*
                                     Form1.Designer.cs*
                                                           Program.

₱ WindowsFormsApplication1.Form1

   ∃using System;
     using System.Collections.Generic;
     using System.ComponentModel;
     using System.Data;
     using System.Drawing;
     using System.Ling;
     using System. Text;
     using System. Threading. Tasks;
     using System.Windows.Forms;
   namespace WindowsFormsApplication1
         public partial class Form1 : Form
             public Form1()
                 InitializeComponent();
                 //this.button1.BackColor = System.Drawing.Color
```

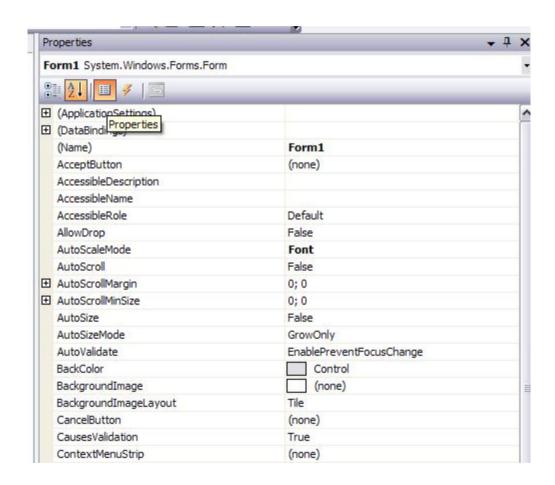
# Svojstva kontrola (1)

 Kontrole imaju svojstva koja se mogu podešavati u fazi dizajna, ali takođe i kodovati neposredno tj. u fazi izvršavanja.



## Svojstva kontrola (2)

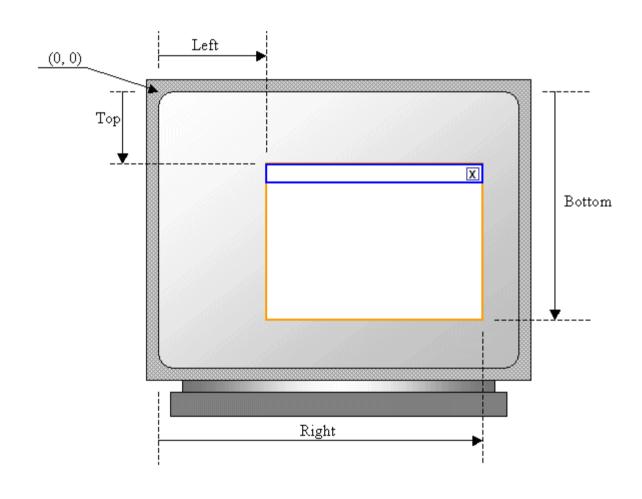
- Najveći broj svojstava su tipična:
- Name
- Location
- Size
- Visible
- Enabled
- TabIndex i TabStop
- Text
- BackColor
- ForeColor
- •



#### DIZAJN Windows kontrola

- Osobine Windows kontrola
  - Zajedničko za sve kontrole je da su po svojoj prirodi takođe prozori.
  - Imaju većinu sličnih svojstava.
  - Prihvatanje određenih događaja je tipično za kontrole.
  - Neki događaji su specifični samo za određenu kontrolu dok su drugi zajednički.

## Koordinatni sistem

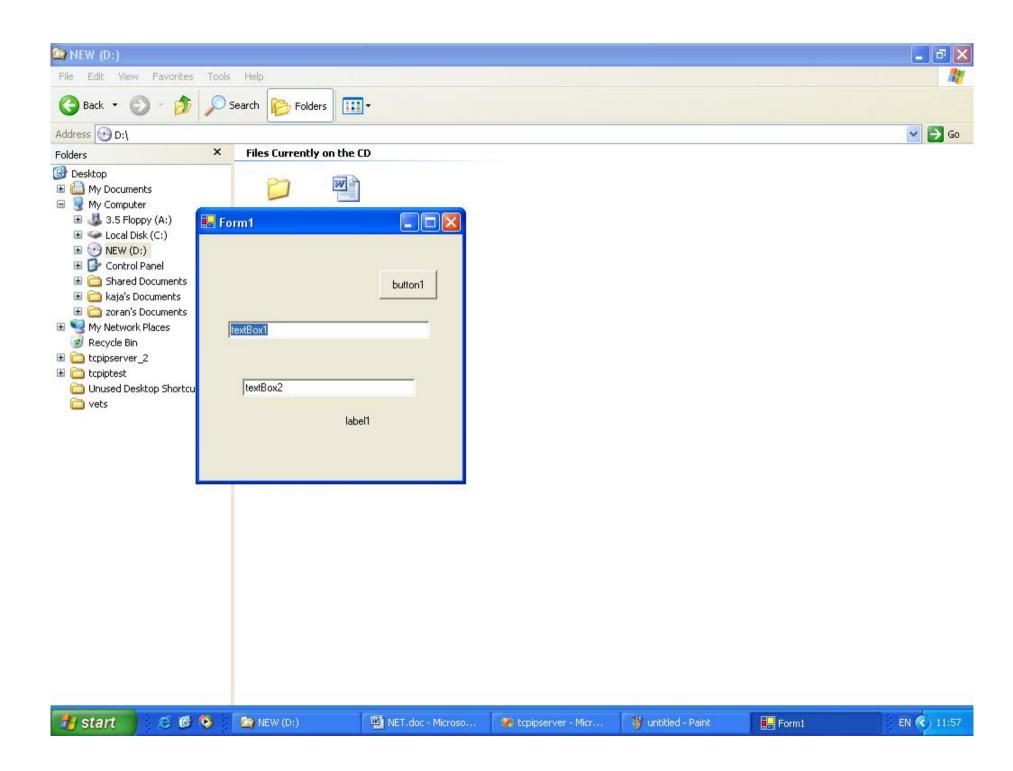


# Relacija roditelj-dete

- Parent-child relationship
- Ovo je podrazumevana relacija kada se kontrole postaljaju i manifestuje se "lepljenjem" kontrole za tu formu.
   Pomeranjem forme pomeraju se sve kontrole na njoj, tj. sva deca prozori.

## Fokus kontrole i aktivan prozor

- U fokusu ako kontrola može da primi ulaz sa tastature. Ono se može posebno grafički označiti što se u podrazumevanom vizuelnom podešavanju i radi.
- Pojam aktivnog prozora se ne vezuje za kontrolu koja pripada nekoj formi već za samu formu. Kontrola koja ima fokus se nalazi na aktivnom prozoru. Samo jedan prozor je aktivan u jednom trenutku.

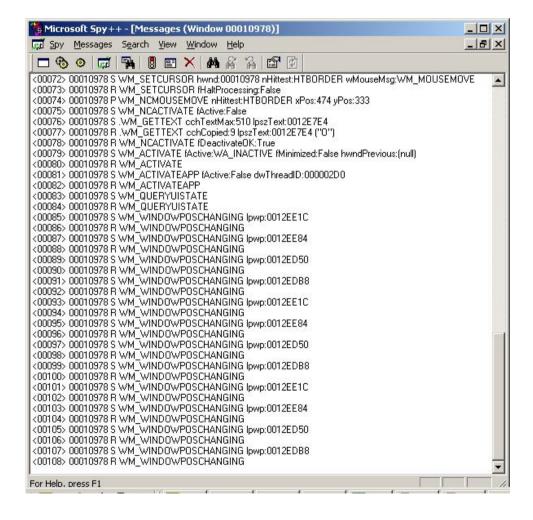


## DOGAĐAJI (I deo)

- Il deo ide sledećeg časa.
- Događaji (*events*) obezbeđuju aplikacijama, odnosno formama odnosno kontrolama da prepoznaju i da reaguju na akcije bilo korisnika bilo drugih aplikacija.
- Spadaju u najvažnije osobine višeprocesnih operativnih sistema kao što su Windows.
- U toku rada neke aplikacije, kontrole i forme ( preciznije - prozori ) neprekidno šalju poruke operativnom sistemu o svim promenama i akcijama korisnika.

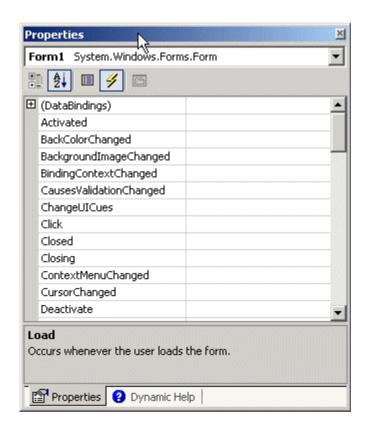
# Spy++

 Spy++ je poseban programski alat, deo Visual Studia, namenjen praćenju poruka nekog procesa ili prozora (I za još po nešto). Spy++ možete da pokrenete iz menija Tools ili sa spiska programa



## Događaji i IDE

- Ime svakog događaja je prikazano na levoj strani.
- Sa desne strane je kombo polje gde možete uneti novo ime i time kreirati novu metodu koja će biti vezana za taj događaj, ili izabrane neku od već postojećih sa liste.
- Kratak opis se nalazi na samom dnu i on vam može pomoći oko lakšeg snalaženja.



#### Metode koje reaguju na događaje...

- ...imaju potpis:
- private void myHandler (object sender, System.EventArgs e)

- sender objekat koji šalje događaj
- e argumenti tj. podaci koji se šalju uz događaj

#### Da li možemo praviti naše događaje?

- Naravno.
- Za to će nam biti potrebno znanje o delegatima i tipu podataka event. (II deo)

## Neke vrste događaja

- Događaji miša
  - Click, Right-Click, MouseUp, MouseDows,...
- Događaji tastature
  - KeyDown, KeyUp, KeyPress,...
- Događaji vezani za formu i kontrole
  - Paint, Resize, ....

Više o njima na vežbama!

#### Kreiranje događaja u kodu koristeći IDE...

```
Button okbut = new Button();
okbut.Click +=

new EventHandler(okbut_Click); (Press TAB to insert)

Button okbut = new Button();
okbut.Click += new EventHandler(okbut_Click);

Press TAB to generate handler 'okbut_Click' in this class

void okbut_Click(object sender, EventArgs e)
{
    throw new Exception("The method or operation is not implemented.");
}
```

#### Primer:

- Zadatak:
  - Napisat W. apl. koristeći VS2012 koja će prikazivati poziciju miša u naslovnoj liniji forme.