

# Упражнения : Въведение в събитийното програмиране


## 1. Ослушващи се за събития

Да се създаде **графично приложение** (GUI application), което позволява да изследвате как и кога се случват различните събития и каква информация ни носи всяко от тях.

### Решение

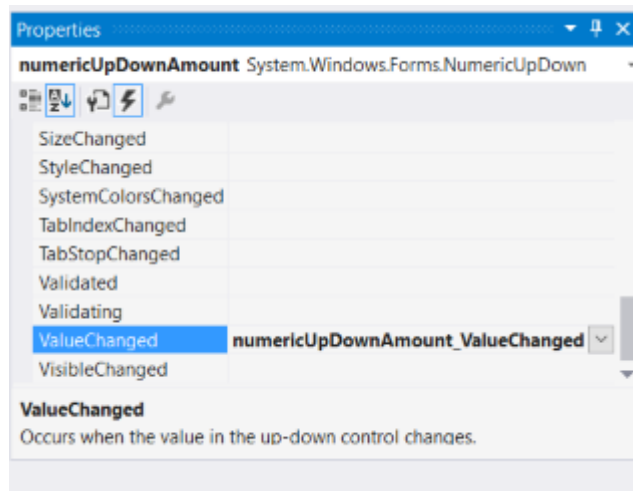
Добавете в ново Visual Studio решение (solution) един проект - **Windows Forms** приложение с име "**Event-Listeners**". Добавете в него един **Label** с име **labelResult**. Той ще показва резултата от обработените събития. Добавете и най-различни други графични контроли, които ще изследваме - по ваш избор. Бъдете експериментатори!

Задайте следните настройки на етикета:

Настройка	Снимка
<b>labelResult:</b> AutoSize = False, BackColor = PaleGreen, TextAlign = MiddleCenter, Font.Size = 14, Font.Bold = True	

Сега дефинирайте най-различни методи-обработчици на събития по контролите. За целта ползвайте иконката със събитията (**Events**) в **Properties** прозореца във Visual Studio. Там с двойно щракване срещу избрани от вас събития генерирайте методи за обработката им. Например:

- **Form1.Load** (като кликнем върху формата 2 пъти с мишката)
- **numericUpDownAmount.ValueChanged** (като кликнем върху **NumericUpDown** контролата 2 пъти)
- **Edit1.KeyUp** (избираме **Events** от таблото **Properties** и кликаме 2 пъти на **KeyUp**)



Събитието **Form1.Load** се изпълнява при стартиране на програмата, преди да се появи прозореца на приложението. Събитието **NumericUpDown.ValueChanged** се изпълнява при промяна на стойността в полето

за въвеждане на число. Събитието **NumericUpDown.KeyUp** се изпълнява след натискане на клавиш в полето за въвеждане на число. При всяко от тези събития ще извеждаме какво се е случило в **labelResult**. Например:

```
private void labelResult_Click(object sender, EventArgs e)
{
    this.labelResult.Text = sender + " clicked";
}
```

Или може да сте още по-конкретни:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.labelResult.Text = "Формата е заредена и готова за работа";
}
```

Опитайте се да използвате и допълнителната информация, която ви се подава с някои събития, например:

```
private void button1_MouseMove(object sender, MouseEventArgs e)
{
    this.labelResult.Text = string.Format("Mouse is at ({0}, {1})", e.X, e.Y);
}
```

Забележете, че тук ползваме допълнителната информация, която ни се подава за това събитие в параметъра **e**. Какви са възможните му свойства, които можете да използвате, ще разберете, като кликнете (с натиснат клавиш Ctrl) върху типа на параметъра, в случая **MouseEventArgs**. Така ще видите неговата декларация.

От имената на повечето събития можете да се досетите кога се генерират. Погледнете и помощната информация в дъното на страницата **Events** в **Properties** прозореца във Visual Studio.

Освен текста на **labelResult** можете да промените при различни събития неговия цвят, позиция и т.н. Бъдете креативни! Имената на свойствата можете да разберете от **Properties** прозореца във Visual Studio.

Накрая **стартирайте проекта** с [Ctrl+F5] и тествайте крайния резултат. Изследователи в действие! 😊

## 2. Хвани бутона!

Създайте забавно графично приложение „хвани бутона“: една форма съдържаща един бутон. При преместване на курсора на мишката върху бутона той се премества на случайна позиция. Така се създава усещане, че „бутонът бяга от мишката и е трудно да се хване“. При „хващане“ на бутона се извежда съобщение-поздрав.



## Подсказка:

Добавете обработчик за събитието **Button.MouseEnter** и премествайте бутона на случайна позиция. Използвайте генератор за случайни числа **Random**. Позицията на бутона се задава от свойството **Location**. За да бъде новата позиция на бутона в рамките на формата, можете да направите изчисления спрямо размера на формата, който е достъпен от свойството **ClientSize**. Можете да ползвате следния примерен код:

```
private void buttonCatchMe_MouseEnter(object sender, EventArgs e)
{
    Random rand = new Random();
    var maxWidth = this.ClientSize.Width - buttonCatchMe.ClientSize.Width;
    var maxHeight = this.ClientSize.Height - buttonCatchMe.ClientSize.Height;
    this.buttonCatchMe.Location = new Point(
        rand.Next(maxWidth), rand.Next(maxHeight));
}
```

Опитайте се да си обясните как точно прави изчисленията на позицията на бутона този програмен код.

Добавете и съобщението-поздрав при успешно натискане на бутона:

```
private void buttonCatchMe_Click(object sender, EventArgs e)
{
    MessageBox.Show("You win!!!");
}
```