## I Етап – Дизайн на интерфейс

Сега ще напишете своя собствена програма на С#.NET - Калкулатор. Като за начало проектът ще бъде опростен минимално и единственото нещо, което ще бъде в състояние да прави, е да събира. След като разберете как работи всичко, ще добавим и останалите операции изваждане, умножение и деление. Версия 1 на вашия калкулатор ще изглежда така:

🔜 Calculator			
1	2	3	+
	5	6	
4			
7	8	9	Clear
	0	Ŀ	

Както виждате, има текстовата кутия за показване на числото, бутони за цифрите от 0 до 9, бутони със символ за точка, плюс и равно, както и бутон "clear"-изчисти.

Така, първото нещо, което трябва да направите, е да проектирате дизайна на вашия калкулатор. Започнете нов проект, като кликнете върху File> New Project.

На новата си форма задайте следните свойства:

Size: 440, 487 Text: Calculator

За да добавите малко цвят на калкулатора, можете да промените свойството BackColor на формата, както на снимката по-долу:

Properties	<b>→</b> 井 >	K			
Form1 System.Windows.Forms.Form					
B 🛃 🖓 🗲	نكر				
AutoSizeMode	GrowOnly 4	*			
AutoValidate	EnablePreventFocusChanc	EnablePreventFocusChang			
BackColor	255, 224, 192 💌				
BackgroundI	Custom Web System				
BackgroundI					
CancelButtor					
CausesValida					
ContextMenu					
ControlBox					
Cursor					
DoubleBuffer					
Enabled					
Font					
ForeColor					
FormBorderS	<u></u>	Ŧ			
BackColor					
The background color of the component.					
	·				

Избрахме цвят оранжев, но не се колебайте да изберете какъвто цвят искате.

Сега добавете текстовата кутия във вашата форма и й задайте следните свойства:

Name: txtDisplay Location: 66, 52 Size: 200, 26 TextAlign: Right

Време е да добавите бутоните. Трябват 10 бутона за числата от 0 до 9. Добавете първия бутон във формата и му задайте следните свойства:

Name: btnZero Font: Microsoft Sans Serif, Bold, 12 Location: 143, 378 Size: 49, 40 Text: 0

Това е бутонът с нула, който отива в дъното на формата. Добавете нов бутон във формата и му задайте следните свойства:

Name: btnOne Font: Microsoft Sans Serif, Bold, 12 Location: 66, 159 Size: 49, 40 Text: 1

По-лесен начин за добавяне на нови бутони, е да ги копирате и поставяте. Кликнете върху **btnOne**, за да го изберете. Кликнете с десния бутон на мишката върху бутона и

изберете **Сору** от контекстното менюто, което се появява. Сега кликнете някъде върху формата. Отново десен клик и изберете **Paste**. Новият бутон ще се появи с числото 1 върху него. Да погледнем, обаче, в прозореца Properties-ще видите, че новият бутон има име button1. Променете го на **btnTwo**. След това променете свойството **Text** на **2**. Плъзнете го, за да се позиционира до бутона с цифра 1.

Добавянето на останалите бутони с цифри става по същия начин: Сору, Paste, промяна на свойствата **Name** и **Text.** За другите бутоните с цифри, използвайте следните стойности на свойството Name: btnThree, btnFour, btnFive и т.н. Поставете вашите бутони, както е показано.

Добавете нов бутон за точката. Дайте му име **btnPoint** и въведете "." в свойство Text. Променете свойството Font (шрифта), ако смятате, че е текстът е твърде малък.

Останаха само три бутона, за да сте готови. Така че, добавете още три бутона и им задайте следните свойства:

Name: btnPlus Font: Microsoft Sans Serif, Bold, 12 Location: 324, 159 Size: 49, 40 Text: +

Name: btnEquals Font: Microsoft Sans Serif, Bold, 12 Location: 324, 230 Size: 49, 40 Text: =

Name: btnClear Font: Microsoft Sans Serif, Bold, 8 Location: 324, 305 Size: 49, 40 Text: Clear

Може да смените местата им, ако не отговарят на подредбата за вашите бутони.

Сега имате завършен дизайн на вашия калкулатор. Запазете тежката си работа ;-) и можем да започнем с кодирането.

# II Етап – Писане на код (1)

# Бутони с цифри

Преди да стигнем до кода, нека да проиграем как ще работи нашия калкулатор:

- 1. Кликване върху бутоните с цифри. Това ще бъде първото число в събирането
- 2. Първото число, което ще се събира трябва да се появи в текстовата кутия
- 3. Кликване върху бутона "плюс" да каже на калкулатора, че ще се извършва действието събиране
- 4. Първото число ще изчезне от текстовата кутия, а тя трябва готова за второто число
- 5. Кликване върху бутоните с цифри отново, за да се добави второ число
- 6. Кликване върху бутона "равно" и отговорът трябва да се появява в текстовата кутия

Първата задача в списъка е да се направи така, че числото да се появи в текстовата кутия, когато един бутон с цифра е бил кликнат. За да направите това, кликнете два пъти върху бутона с цифра 1 за отваряне на прозореца за код.

Цифрите на бутоните бяха сложени чрез свойството Text. Така че, само трябва да използвате това свойство. Можем да използваме текста на бутона като стойност на свойство text за текстовата кутия. Добавете следния ред код за вашия **btnOne**:

txtDisplay.**Text** = btnOne.**Text**;

Това означава: "Направи текста в текстовата кутия, същият като текста, който е на бутона". Запомнете: каквото и да е от дясната страна на знака за равенство ще се съхранява в каквото е от ляво на знака за равенство.

Стартирайте програмата, за да я изпробвате. Кликнете върху бутона с цифра 1 и 1 ще се появи в текстовата кутия. Кликнете няколко пъти върху същия бутон и какво забелязвате? Може би си мислите, че като кликнете върху бутона с цифра 1 два пъти, например, ще доведе до това, в текстовата кутия да се покаже 11, а не 1. В края на краищата, вие го кликнахте два пъти, тогава защо не се появи два пъти 1?

Причината за това е, че не сте казали на С# да запази стойността, която вече сте въвели. Всеки път, когато кликнете върху бутона, С# ще изпълнява действието отначало - той не знае какво е било там преди и ще изчисти числото, което преди това сте въвели.

Спрете програма и се върнете към вашия код. Променете вашия ред код така:

## txtDisplay.Text = txtDisplay.Text + btnOne.Text;

Този ред лесно се чете, ако просто погледнете частта след знака за равенство:

#### txtDisplay.Text + btnOne.Text;

Когато работите с текст, символът плюс не означава "събери", а означава "свържи" (операцията конкатенация, която научихте при работата с низове). Така че С# ще съедини текста в текстовата кутия с текста на бутона. След като свърши тази работа, той ще съхранява отговорът на всичко, което е вляво от знака за равенство. В този случай, това не е променлива, а свойството text на текстовата кутия.

Стартирайте програмата отново. Кликнете върху бутона с 1няколко пъти. Вече 1трябва да се появи в текстовата кутия повече от веднъж.

Спрете програмата и се върнете не към кода, а към самата форма. (Ако не можете да видите вашата форма, кликнете с десния бутон на **Form1.cs** в Solution Explorer отдясно. От менюто, което се появява, изберете **View Designer**.)

Сега кликнете два пъти бутон 2 и добавете следния код:

txtDisplay.Text = txtDisplay.Text + btnTwo.Text;

Единственото нещо, което е различно е името на бутона – **btnTwo**, вместо **btnOne**. Всичко останало е еднакво.

Направете същото и за останалата част от вашите бутони, променяйте името на бутоните всеки път. (Можете да копирате и поставяте кода си, за да се спести време.)

Когато сте готови, прозорецът ви за кодиране трябва да изглежда така,:

```
private void btn0ne Click(object sender, EventArgs e)
ł
    txtDisplay.Text = txtDisplay.Text + btnOne.Text;
}
private void btnTwo Click(object sender, EventArgs e)
    txtDisplay.Text = txtDisplay.Text + btnTwo.Text;
3
private void btnThree_Click(object sender, EventArgs e)
    txtDisplay.Text = txtDisplay.Text + btnThree.Text;
}
private void btnFour_Click(object sender, EventArgs e)
    txtDisplay.Text = txtDisplay.Text + btnFour.Text;
}
private void btnFive_Click(object sender, EventArgs e)
    txtDisplay.Text = txtDisplay.Text + btnFive.Text;
3
private void btnSix_Click(object sender, EventArgs e)
Ł
    txtDisplay.Text = txtDisplay.Text + btnSix.Text;
}
private void btnSeven Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnSeven.Text;
}
private void btnEight Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnEight.Text;
3
private void btnNine_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnNine.Text;
}
private void btnZero Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnZero.Text;
}
```

Стартирайте програмата си отново и кликнете всичките десет от вашите бутони. Уверете се, че всяка цифра се появява в текстовата кутия, когато дадения бутон е натиснат.

Върнете се към вашата форма и кликнете два пъти върху бутона **Clear** и добавете следния ред:

### txtDisplay.Clear();

След точката изпишете думата **Clear**, последвана от чифт кръгли скоби. Ясно е, че е метод, който можете да използвате за текстови кутии. Както подсказва името му, той ще изчисти текстовата кутия, като я остави празна.

Стартирайте отново програмата си, кликнете на няколко цифри и след това тествайте вашия бутон Clear. Числата трябва да изчезнат от текстовата кутия.

В следващата част, ще добавите кода за бутона Plus.

## II Етап – Писане на код (2)

# Бутон "+"

Вече имаме начин, цифрите да се появяват в текстовата кутия, когато бутонът се натисне. Следващото нещо, което трябва да направите, е числото да се съхранява някъде. След това ще използвате числото, когато се натисне бутонът "равно" (бутонът "равно" ще прави събирането, не бутона плюс).

Работата на бутона плюс ще е да вземе числото от текстовата кутия. Тъй като числото ще е текст, а не числова стойност, ще трябва да го превърнете в такава. След което, тази стойност да се съхранява в променлива. Единственият друг ред код, от който ще имате нужда е още един ред код за бутона "+", с който да се изчиства текстовата кутия и да я подготви за второто число.

Първото число трябва да се съхранява в променлива. Ще използваме типа double за променливата. По този начин ще можем да работим с наистина големи числа. За да могат всички бутони в програмата да имат достъп до тази променлива, тя трябва да се декларира извън кода на бутоните. Така че, не можем да направим това:

```
private void btnOne_Click(object sender, EventArgs e)
{
     double total1 = 0;
}
```

Ако се създаде променлива вътре на един бутон, само този бутон ще бъде в състояние да направи нещо с променливата. Това е известно като **обхват на променливите**. Тъй като сме декларирали променливата вътре в кода на бутона, ние сме й дали **локален** обхват: тя ще бъде достъпна само в рамките на фигурни скоби. Нарича се още **локална променлива**. За да направите променливата достъпна за всички бутони, трябва да й дадем глобален обхват или да я направим **глобална променлива**. Това е доста лесно - просто го я декларирайте извън методите за събитие на бутоните. Подобно на това:

```
double total1 = 0;
private void btnPlus_Click(object sender, EventArgs e)
{
}
```

Сега променливата е извън фигурни скоби, което я прави достъпна за всички бутони във формата. По същия начин създайте променлива във вашия код.

В кода на бутона **btnPlus** добавете следните два реда (удебелените в синьо по-долу):

```
double total1 = 0;
private void btnPlus_Click(object sender, EventArgs e)
{
    total1 = total1 + double.Parse( txtDisplay.Text );
    txtDisplay.Clear();
}
```

Това, което се случва тук, е получаване на текста от текстовата кутия, който се превръща в стойност от тип double, и след това се съхранява в променливата total1. Забележете, че ние също сме направили това:

## total1 = total1 +

Така, както направихме за бутоните с цифри, така трябва и да съхраняваме стойностите в променливата total1. Трябва да направите това в случай, че искате да събирате повече от две числа. Ако не се съхранява това, което е в променливата total1 всеки път, С# ще "забрави" за него, и ще започне всичко отначало. Тази техника се използва толкова често в програмирането и даже се използва съкратения запис:

## total1 += double.Parse(txtDisplay.Text);

В този запис, вместо да се повтаря името на променливата, можете просто да използвате символ плюс и за равенство заедно (+ =). Горният ред прави същото нещо като този:

### total1 = total1 + double.Parse(txtDisplay.Text);

Който и начин да изберете за запис на запазване на стойността на дадена променлива, все ще означава за С#: "Пази всичко, което е вече в променливата и добави още нещо".

След запазване на стойността от текстовата кутия, тя трябва да се изчисти:

### txtDisplay.Clear();

След като текстовата кутия е изчистена, второто число може да бъде въведено, като се кликне на бутоните с цифри.

В следващата част, вие ще научите как да пише код за бутона "=".

## II Етап – Писане на код (3)

# Бутон "="

Бутонът Equals-,,=" е мястото, където се развива действието. Това е мястото, където ще се случва всъщност събирането.

За да съхраните отговора от събирането, ще имате нужда от друга променлива (с удебеленото синьо по-долу):

double total 1 = 0;

#### double total2 = 0;

private void btnPlus\_Click(object sender, EventArgs e)
{

total1 = total1 + double.Parse(txtDisplay.Text);
txtDisplay.Clear();

}

По същия начин създайте променливата total2.

Върнете се към вашата форма и кликнете два пъти върху бутона "=",за да се върнете във вашия код. Сега добавете следните три реда:

total2 = total1 + double.Parse( txtDisplay.Text ); txtDisplay.Text = total2.ToString( ); total1 = 0;

Първият ред трябва да ви изглежда познато:

total2 = total1 + double.Parse( txtDisplay.Text );

В дясно от знака за равенство, правим същото нещо, както преди:

#### total1 + double.Parse(txtDisplay.Text);

Разликата е преди знака за равенство: сега стойността се съхранява в променливата total2:

**total2** = total1 + double.Parse(txtDisplay.Text);

С други думи, вземате стойността от текстовата кутия, превръщате я променлива от тип double и я добавяте към това, което е в totall. Когато се извършат действията, отговорът ще се съхранява в променливата наречена total2.

Втората линия код е този:

txtDisplay.Text = total2.ToString( );

Вдясно от знака за равенство, превръщаме променливата **total2** в String. Това е така, за да може да се покаже като текст в текстовата кутия.

Третият ред код нулира променливата total1:

#### total1 = 0;

Това се прави, за да може новата сума може да се изчисли правилно.

Време е да изпробвате вашия калкулатор. Използвайте го, за да изчисли следните суми:

# $\begin{array}{r} 10+25\\ 36+36\\ 10+10+10\end{array}$

Разбира се, тези изчисления можете да направите и наум. Но се уверете, че вашият калкулатор получава точната сума преди да продължи със следващата. Кликнете върху вашия бутон **Clear**, за да започнете ново въвеждане. Когато сте сигурни, че сте разбрали това, което се случва с кода, опитайте това упражнение.

#### Упражнение С

Вие все още не сте написали нито един код за **btnPoint** бутона. Това означава, че не можете да имате номера като 10.5 или 36.7 във вашите изчисления. Напишете кода за решаване на този случай.(Съвет: имате нужда само от един ред код.)