



WinForm opdrachten

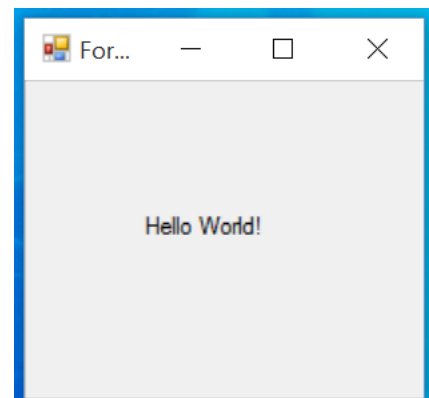


Vragen

1. Schrijf op welk C# component een programma scherm representeert en die andere componenten kan bevatten.
2. Schrijf op waar het "Label" component voor gebruikt wordt.
3. Schrijf op wat het meest algemene grafische onderdeel is van een Windows programma?
4. Leg uit wat "Form controls" (venster componenten) zijn.
5. Schrijf twee verschillende Windows componenten op en leg uit wat hun functies zijn.
6. Schrijf op wat "Event-Driven" programmeren inhoudt.
7. Schrijf op welke eigenschap (property) van een Textbox component je zou moeten lezen of instellen om de tekst ervan te kunnen controleren.
8. Schrijf op waar scroll bars voor gebruikt worden.
9. Leg uit waarom het belangrijk is om de componenten op een form duidelijke namen te geven.
10. Leg uit of een Textbox component standaard "single-line" of "multi-line" is.
11. Schrijf op welke eigenschap je moet aanpassen in de "Form Designer" om meer keuzes toe te voegen aan een List box of Combo Box component.
12. Leg uit hoeveel Radio Buttons tegelijkertijd kunnen worden geselecteerd binnen 1 groep.
13. Leg uit hoeveel Check Boxen tegelijkertijd geselecteerd kunnen worden op een venster.
14. Schrijf op welke eigenschap aangeeft of een Radio button of een Check Box geselecteerd is.
15. Schrijf op welke eigenschap je moet aanpassen om de titel van een form te wijzigen.

Opdrachten

1. Start Visual studio, kies Visual C# en selecteer daar Windows Forms Application. Geef het programma de naam *WinFormOpdracht-001*. Kies het pad en de plaats waar je het project wilt opslaan. Maak nu het "Hello World!" programma door gebruik te maken van het component "Label". Pas de eigenschap Tekst van het "Label" component aan en voer het programma uit. De uitvoer is als hier naast.

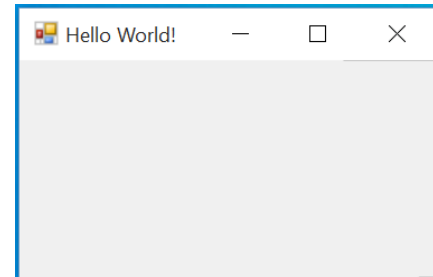




WinForm opdrachten



- Schrijf het "Hello World!" programma waar bij "Hello World!" als titel in het venster komt te staan. Noem het WinFormOpdracht-002. Pas de eigenschap Tekst van Form1 aan met "Hello World!" Als het programma uitgevoerd wordt dan ziet het er zoals hiernaast uit.



- Maak een Windows form programma dat de naam heeft WinFormOpdracht-003.

Plaast een Label op het venster en pas de eigenschap Text van het label aan met "" (een lege tekst string). Plaats ook een Button op het venster. Verander de waarde van het Text eigenschap in "Klik mij."

Nu moet er een event gemaakt worden voor wanneer de gebruiker op de knop drukt. Doe dit door te dubbel klikken op de net gemaakte knop. Visual Studio gaat over naar de tekst editor en maakt het event aan.

```
private void button1_Click(object sender, EventArgs e)
{
}

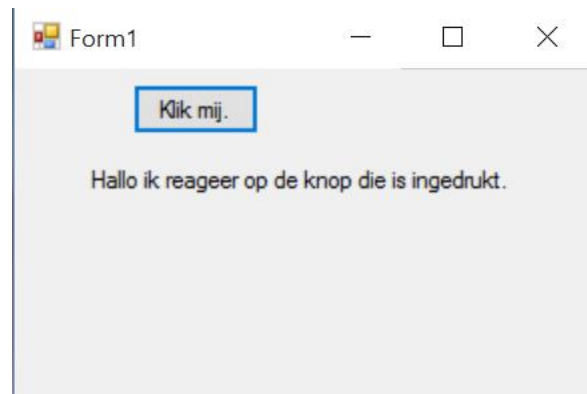
```

Nu moet er tussen de haken een code worden geschreven om een tekst in het label component te laten zien. Pas de event method als volgt aan:

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = "Hallo ik reageer op de knop die is ingedrukt.";
}

```

Voer het programma uit en kijk wat er gebeurt als je op de knop klikt.





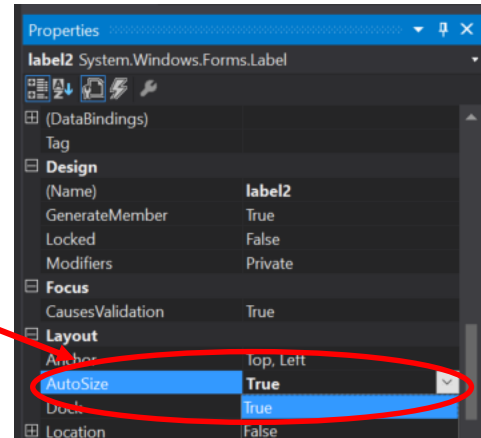
WinForm opdrachten



4. Maak een Windows form programma dat de naam heeft WinFormOpdracht-004.

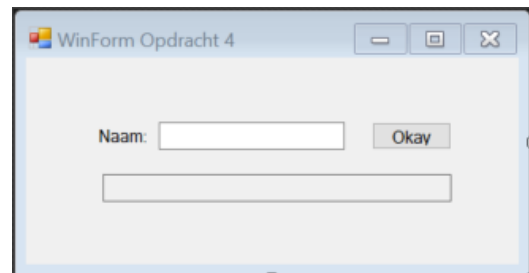
Bij dit programma gaan we gebruik maken van de componenten:

- label1
 - o Text: Naam:
- label2
 - o Zet eigenschap AutoSize op false
 - o Pas de grootte van het label aan door te slepen
 - o Zet de eigenschap Tekst op ""
 - o BorderStyle: FixedSingle
- TextBox1
- Button1
 - o Text: Okay
- Form1
 - o Tekst: WinForm Opdracht 4



Het venster ziet er dan als volgt uit:

Het doel is dat de gebruiker een naam in de TextBox invult en wanneer hij/zij op de knop okay drukt er de tekst "Hallo xxxxx, welkom bij programmeren in C#!" in label2 laat zien.



Er moet nu nog broncode geschreven worden voor twee Events. Eén om de invoer van het TextBox1.Text veld op te slaan in een string variabele naam. De ander om de tekst in label2 te verschijnen.

Dubbel klik op het TextBox component zodat Visual Studio het TextBox event TextChanged aanmaakt.

Plaats net boven de method `public Form1()` de declaratie van de variabele die als volgt is `string naam = "";`

In het `textBox1_TextChanged` event plaats de code `naam = textBox1.Text;`



WinForm opdrachten

Deze zorgt ervoor dat de waarde die de gebruiker in de textbox invult bewaard wordt in de variabele naam.

Vervolgens dubbel klik op de Button1 om het event button1_Click aan te laten maken door visual studio. Voeg daar de code

```
label2.Text = "Hallo " + naam + ", welkom bij programmeren in C#!";
```

toe aan het event.

Voer het programma uit en bekijk hoe het werkt.

De uitvoer zal er zo uit zien:



Probeer een andere naam in te vullen en weer op de okay knop te klikken. Wat gebeurt er dan in het label2?

5. De MessageBox

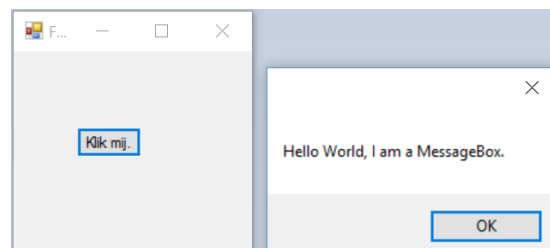
Maak een nieuw Windows Forms Applicatie, noem het WinFormOpdracht-005

Plaats een button op de form en verander de eigenschap Text in "Klik mij".

Maak een button click event die een MessageBox venster opent. Tik de volgende code in het event zodat het er als volgt uit ziet:

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello World, I am a MessageBox.");
}
```

Voer het programma uit en bekijk het resultaat.





WinForm opdrachten

6. Simpel Calculator

Hieronder volgt de verschillende componenten en de aanpassingen die aan de eigenschappen gedaan moeten worden”:

Form1

(Name): CalculatorForm

Text: "Calculator"

Label

(name): NumberLabel

BackColor: White

BorderStyle: FixedSingle

AutoSize: False

Button1, etc.

Text: 1,2,etc

ClearButton

Text: C

AddButton

Text: "+"

SubtractButton

Text: "-"

EqualButton

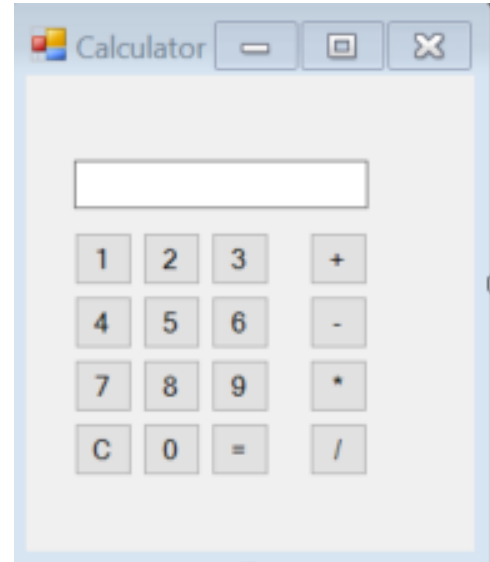
Text: "="

MultiplyButton

Text: "*"

DivideButton

Text: "/"



Na dat de form gemaakt is in de Form Designer, dubbel klik op alle buttons om de events te laten genereren.

Methods en Events schrijven

Dubbel klik op het button met het cijfer 1. Visual Studio maakt het klik event voor de button aan. Vul de code als volgt in voor dit event:

```
private void button1_Click(object sender, EventArgs e)
{
    NumberLabel1.Text += "1";
}
```

Doe hetzelfde voor de andere cijfer knoppen 2,..,0.



WinForm opdrachten

Maak het klik event voor de ClearButton en vul de code als volgt in:

```
private void ClearButton_Click(object sender, EventArgs e)
{
    NumberLabel.Text += "";
}
```

Nu moeten er boven in de Form code een aantal variabelen worden gedeclareerd zoals hier onder:

```
public partial class CalculatorForm : Form
{
    int firstNum = 0;
    int result = 0;
    bool addButtonFlag = false;
    bool subtractButtonFlag = false;
```

firstNum bevat de waarde van het eerste getal dat de gebruiker invoert. De variabele result zal het resultaat van de berekening opslaan.

De twee booleaanse waarden worden gebruikt om te controleren welke bewerkingsknop de gebruiker heeft ingedrukt.

Nu kunnen we de code voor bijvoorbeeld de AddButton invoeren. Deze is als volgt:

```
private void AddButton_Click(object sender, EventArgs e)
{
    if (NumberLabel.Text != "")
    {
        firstNum = int.Parse(NumberLabel.Text);
        NumberLabel.Text = "";
        addButtonFlag = true;
    }
    else
    {
        MessageBox.Show("You must enter a number!");
    }
}
```

De eerste regel controleert of er een waarde is ingevoerd. Is dat niet het geval dan wordt er een melding getoond. Als dat het geval is dan kan die waarde in de variabele firstNum worden opgeslagen. Tevens wordt de addButtonFlag waarde op true



WinForm opdrachten

gezet want de add knop was door de gebruiker ingedrukt. Ook wordt het NumberLabel weer leeg gemaakt zodat de gebruiker een tweede getal kan gaan invoeren. De code die in het *e/se* blok staat wordt uitgevoerd wanneer de gebruiker géén waarde heeft ingevoerd.

De code voor het SubtractButton event is bijna hetzelfde als die van de AddButton. Hier onder volgt de code die daar ingevoerd dient te worden.

```
private void SubtractButton_Click(object sender, EventArgs e)
{
    if (NumberLabel.Text != "")
    {
        firstNum = int.Parse(NumberLabel.Text);
        NumberLabel.Text = "";
        subtractButtonFlag = true;
    }
    else
    {
        MessageBox.Show("You must enter a number!");
    }
}
```

Het event van de ClearButton moet nu verder worden uitgebreid met het terug zetten van de waarde van de variabelen. De code wordt als volgt:

```
private void ClearButton_Click(object sender, EventArgs e)
{
    result = 0;
    firstNum = 0;
    NumberLabel.Text += "";
    addButtonFlag = false;
    subtractButtonFlag = false;
}
```



WinForm opdrachten

Het EqualsButton event wordt gebruikt om de berekening uit te voeren en het resultaat te tonen in NumberLabel.Text. De code voor dit event is als volgt:

```
private void EqualButton_Click(object sender, EventArgs e)
{
    if (NumberLabel.Text != "")
    {
        int secondNum = int.Parse(NumberLabel.Text);

        if (addButtonFlag == true)
        {
            result = firstNum + secondNum;
        }
        else if (subtractButtonFlag == true)
        {
            result = firstNum - secondNum;
        }
        NumberLabel.Text = result.ToString();
    }
}
```

Nadat alle handelingen zijn uitgevoerd en de broncodes zijn ingetikt voer het programma uit en kijk hoe het werkt.

Twee berekening events, vermenigvuldigen en delen zijn nog niet gemaakt. Probeer deze zelf toe te voegen.

Maak de event methoden voor de twee knoppen. Vul de onderstaande broncode bij de events in als volgt:

```
private void MultiplyButton_Click(object sender, EventArgs e)
{
    if (NumberLabel.Text != "")
    {
        firstNum = int.Parse(NumberLabel.Text);
        NumberLabel.Text = "";
        multiplyButtonFlag = true;
    }
    else
    {
        MessageBox.Show("You must enter a number!");
    }
}
```



WinForm opdrachten

```
private void DivideButton_Click(object sender, EventArgs e)
{
    if (NumberLabel.Text != "")
    {
        firstNum = int.Parse(NumberLabel.Text);
        NumberLabel.Text = "";
        divideButtonFlag = true;
    }
    else
    {
        MessageBox.Show("You must enter a number!");
    }
}
```

Pas het EqualButton event aan met de volgende code, na het blok van de multiplyButtonFlag code:

```
else if (multiplyButtonFlag == true)
{
    result = firstNum * secondNum;
}
```

Voor het berekenen van een deling moet wat meer werk gedaan worden, daar er niet door nul gedeeld mag worden. Dit moet worden opgevangen met een **Exception handling** instructie die heet "try - catch"

De code wordt als volgt:

```
else if (divideButtonFlag == true)
{
    // Gebruik de try instructie om potentiële
    // fouten op te vangen bij een deling
    try
    {
        result = firstNum / secondNum;
    }
    catch (System.Exception ex)
    {
        String error = ex.ToString();
        MessageBox.Show("Oops! That calculation is not
            allowed. Please try again!");
    }
}
```



WinForm opdrachten

In de clearButton event moet de volgende code worden toegevoegd:

```
multiplyButtonFlag = false;  
divideButtonFlag = false;
```

Test het programma opnieuw, kijk of de foutmelding goed wordt gegeven.