

Samenvatting

Een belangrijk deel van de programmeertaak bestaat eruit besturingselementen op een formulier te zetten en hieraan bepaalde initiële properties (eigenschappen) mee te geven. Met de faciliteiten van de IDE van C# is dat een koud kunstje, al moet je er wel veel mee oefenen.

Opgaven

2.1

Creëer een nieuw project genaamd **Demo** en plaats drie knoppen op het formulier. Zet de tekst ervan op respectievelijk **1**, **2** en **3**. Maak ook drie labels, met respectievelijk **A**, **B** en **C** erop. Zet nu in de methoden behorend bij de knoppen een code die ertoe leidt dat:

- (a) het aanklikken van button1 de tekst van alle labels op **Ja** zet;
- (b) het aanklikken van button2 de tekst van alle knoppen op **Nee** zet;
- (c) het aanklikken van button3 de 'tekstwaarden' terugzet op **A**, **B** en **C**.

2.2

Schrijf een programma met twee knoppen en één label. Geef het label een gele kleur. Klik je op de ene knop, dan wordt het label onzichtbaar, klik je op de andere knop, dan wordt het label weer zichtbaar.

Tip: deze opgave maakt gebruik van de `Visibility`-property (die de zichtbaarheid van een besturingselement bepaalt). Deze property kan op **Visibility.Visible** (zichtbaar) of **Visibility.Collapsed** (verborgen) staan. De volgende code maakt `resultLabel` bijvoorbeeld onzichtbaar:

```
resultLabel.Visibility = Visibility.Collapsed;
```

2.3

Schrijf een programma dat eerst je naam toont in een `MessageBox` en vervolgens, na het aanklikken van een knop, je leeftijd in een nieuwe `MessageBox`.

Tip: maak hierbij gebruik van het `Loaded`-event van het formulier. In het property-venster voor het formulier (Window) klik je op het bliksem-icoon om de events te tonen en vervolgens dubbelklik je op 'Loaded'. Figuur 2.25 toont het property-venster met de events die op een formulier van toepassing zijn.

FIGUUR 2.25 Een event selecteren in het property-venster



2.4

We gaan een eenvoudige teksteditor maken. Plaats een tekstvak (TextBox) op het formulier en maak dit bijna net zo groot als het formulier zelf. Geef het de volgende properties:

- VerticalScrollBarVisibility = "Visible"
- TextWrapping = "Wrap"
- AcceptsReturn = "True"

Run het programma. Typ wat tekst in het tekstvak. Merk op dat je na een rechtermuisklik kunt knippen en plakken in de tekst. Start nu buiten de IDE een willekeurige tekstverwerker en verplaats eens wat tekst van en naar je C#-teksteditor.

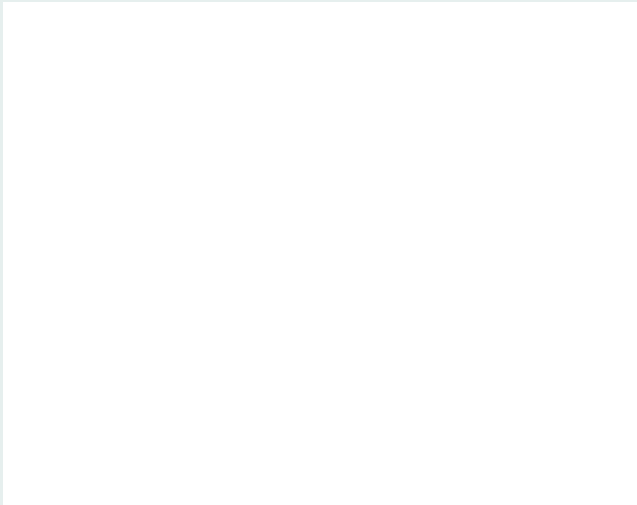
2.5

Het event `MouseEnter` treedt op wanneer de cursor over een besturingselement wordt bewogen. Of beter gezegd: wanneer de muis van buiten het element naar binnen komt (*enter*). Schrijf nu een programma dat een `MessageBox` laat zien met de tekst **Over Button** zodra je de muis boven een knop houdt.

2.6

Heel vaak kun je de Label-control vervangen door een TextBlock-control. Er is echter één groot verschil tussen deze twee controls. Een Label heeft een Target-property waarmee je kunt aangeven welke control de focus krijgt wanneer een bepaalde toetsencombinatie gebruikt wordt. Gebruik de Help-functie en zoek deze property op. Bouw vervolgens een formulier met een TextBlock in het groen, een Label en een TextBox. Zorg dat de cursor knippert in deze TextBox als de gebruiker de combinatie Alt-N indrukt.

FIGUUR 2.26 Een Label of een TextBlock



Antwoorden op de testvragen

2.1 Een voorbeeld van hoe je code op het einde van de oefening eruit zou kunnen zien is:

→ **Project: h02\TestVraag2_1 | Bestand: MainWindow.xaml**

```
<Grid>
    <Label Content="Naam: Kris Hermans" HorizontalAlignment="Left"
        Margin="100,60,0,0" VerticalAlignment="Top"
        FontFamily="Times New Roman"
        Background="Green"
        Name="personLabel"/>
    <TextBlock Text="Leeftijd: 40" HorizontalAlignment="Left"
        Margin="100,100,0,0" VerticalAlignment="Top"
        FontFamily="Times New Roman"
        Background="Orange"
        Name="ageLabel"/>
```