



Programmeren in Microsoft Visual Basic 2010 Express,
lessenserie voor het voortgezet onderwijs HAVO/VWO

© David Lans, Emmauscollege, Vespucci College, Marnix Gymnasium
Rotterdam, december 2011

Hoofdstuk 6: Zelf functies maken

6.0 Leerdoel

In dit zesde hoofdstuk van de cursus “Programmeren in Visual Basic” leer je:

- kennis maken met de volgende onderdelen van een Visual Basic project:
 - zelf opgegeven functies zonder invoerparameter
 - zelf opgegeven functies met invoerparameter
- nieuwe gebeurtenissen in een project beschrijven met programmacode, zoals:
 - zelf opgegeven functies gebruiken in een programma
- enkele algemene regels voor het programmeren
 - testen van een programmaonderdeel

We doen dat aan de hand van de voorbeeldprojecten:

- Dobbelsteen3
- Dobbelsteen4
- Yahtzeedobbelstenen1
- Yahtzeedobbelstenen2

Met dit hoofdstuk ben je, afhankelijk van je tempo, ongeveer 2 lessen van 50 minuten bezig.

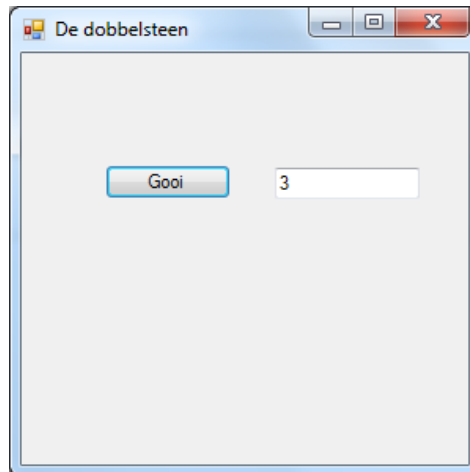
6.1 Een eenvoudige functie dobbelsteen()

In hoofdstuk 5 heb je kennis gemaakt met in Visual Basic ingebouwde functies voor getallen (“Int”, “Rnd”, “/”, “\”, “Mod”, “Div”) en teksten (“+”, “Len”, “Mid”).

In Visual Basic kan je ook zelf functies maken. Dat doe je als je in een programma vaak eenzelfde handeling verricht. In dit hoofdstuk bouwen een Visual Basic programma waarin meerdere malen met een dobbelsteen gegooid moet worden. We maken dus een functie voor het gooien van een dobbelsteen.



Maak een project “Dobbelsteen3” met het formulier “frmDobbelsteen” met daarop een knop “btnGooi” en een tekstveld “txtDobbelsteen”:



De bijbehorende programmacode is:

```
btnGooi Click
Public Class frmDobbelsteen
    Private Sub frmDobbelsteen_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Randomize()
    End Sub
    Function dobbelsteen() As Integer
        dobbelsteen = 1 + Int(6 * Rnd())
    End Function
    Private Sub btnGooi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGooi.Click
        txtDobbelsteen.Text = dobbelsteen()
    End Sub
End Class
```

Dubbelklik op het formulier om de “Randomize” instructie aan het formulier (frmDobbelsteen_Load) toe te voegen. Deze instructie zorgt ervoor dat bij het laden van het formulier de random-number generator geïnitieerd wordt.

Voeg vervolgens de functie dobbelsteen() aan de programmacode toe. De functie dobbelsteen() levert een geheel getal op als uitvoerwaarde: Function dobbelsteen() As integer.

Dubbelklik op de knop “btnGooi” om een door de functie dobbelsteen() gekozen waarde in het tekstveld txtDobbelsteen te plaatsen.

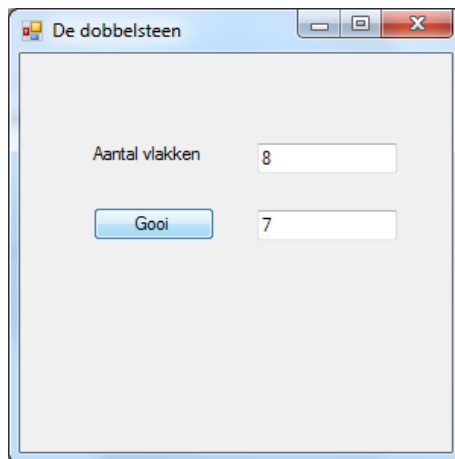
Het is handig om, als je functies in een programma wilt gaan gebruiken, deze eerst uit te **testen**. Zodra je weet dat de functie werkt, is hij klaar voor gebruik !

6.2 Een eenvoudige functie dobbelsteen() met invoerparameter

Een functie hoeft niet alleen een uitvoer te leveren maar kan ook invoer nodig hebben. Stel je voor dat je Visual Basic niet alleen met een dobbelsteen met 6 vlakken wilt kunnen laten gooien maar met dobbelstenen met een willekeurig aantal vlakken.



De functie dobbelsteen heeft dan als invoer nodig hoeveel vlakken de dobbelsteen heeft:



Maak een project Dobbelsteen4 bestaand uit een formulier “frmDobbelsteen” met een label “lblVlakken”, een tekstveld “txtVlakken”, een knop “btnGooi” en een tekstveld “txtDobbelsteen”.

De programmacode luidt als volgt:

```
frmDobbelsteen - dobbelsteen
Public Class frmDobbelsteen
    Private Sub frmDobbelsteen_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Randomize()
    End Sub
    Function dobbelsteen(ByVal vlakken As Integer) As Integer
        dobbelsteen = 1 + Int(vlakken * Rnd())
    End Function
    Private Sub btnGooi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGooi.Click
        Dim aantalvlakken As Integer
        aantalvlakken = txtVlakken.Text
        txtDobbelsteen.Text = dobbelsteen(aantalvlakken)
    End Sub
End Class
```

De functie dobbelsteen heeft nu een invoerwaarde “vlakken” nodig. Deze waarde bepaalt de mogelijke uitkomsten van de dobbelsteen en wordt dus opgenomen ter vervanging van de “6” (vlakken).

Als nu de functie dobbelsteen bij een druk op de knop “cmdGooi” aangeroepen wordt, dan dient het aantalvlakken uit het tekstveld “txtAantalvlakken” ingevoerd te worden in de functie dobbelsteen().

Zowel de invoerwaarde als de uitvoerwaarde van de functie dobbelsteen() zijn van het type Integer. We noemen een waarde die als invoer aan een functie meegegeven dient te worden ook wel een **parameter** van de functie.

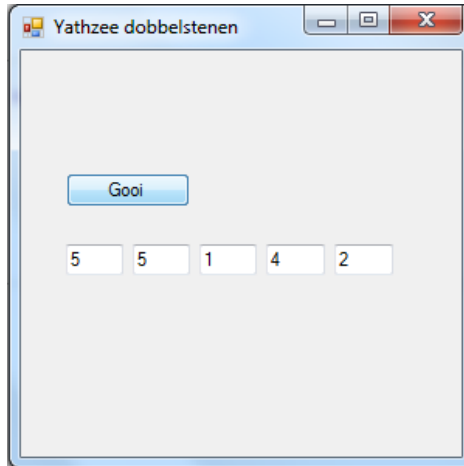


Sla het project op als “Dobbelsteen4”.

6.3 Meervoudig aanroepen van een functie

Zoals al eerder vermeld is een functie pas dan handig als je deze in een programma meerdere malen aanroept. Bij het spelletje Yahtzee wordt met vijf dobbelstenen tegelijk gegooid:

- Maak een project bestaand uit een formulier “frmYahtzeedobbelstenen” met daarop een knop “btnGooi” en vijf tekstvelden “txtDobbel1” t/m “txtDobbel5”:



De bijbehorende programmacode is:

```
(General) (Declarations)
Public Class frmYahtzeedobbelstenen
    Private Sub frmDobbelsteen_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Randomize()
    End Sub
    Function dobbelsteen() As Integer
        dobbelsteen = 1 + Int(6 * Rnd())
    End Function
    Private Sub btnGooi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGooi.Click
        txtDobbel1.Text = dobbelsteen()
        txtDobbel2.Text = dobbelsteen()
        txtDobbel3.Text = dobbelsteen()
        txtDobbel4.Text = dobbelsteen()
        txtDobbel5.Text = dobbelsteen()
    End Sub
End Class
```

- Sla het project op onder de naam “Yahtzeedobbelstenen1”.

Bij het spel Yahtzee krijgt een speler na de eerste worp nog twee keer de mogelijkheid om dobbelstenen naar keuze opnieuw te gooien.



Maak een kopie van het project “Yahtzeedobbelstenen1” en breid hierin het formulier uit met een aantal knoppen “btnHold1” t/m “btnHold5”, een label “lblWorp” met daarnaast een label “lblAantalworpen” dat het aantal worpen bijhoudt en een knop “btnNieuw” waarmee na elk aantal worpen een nieuwe beurt kan beginnen.



Bovenin de programmacode worden de variabelen hold1 t/m hold5 en beurt opgegeven, zodat ze in meerdere subroutines kunnen worden gebruikt.

```
frmYathzeedobbelstenen (Declarations)
Public Class frmYathzeedobbelstenen
    Dim hold1, hold2, hold3, hold4, hold5 As Boolean
    Dim beurt As Integer
    Private Sub frmDobbelsteen_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Randomize()
        hold1 = False
        hold2 = False
        hold3 = False
        hold4 = False
        hold5 = False
    End Sub
    Function dobbelsteen() As Integer
        dobbelsteen = 1 + Int(6 * Rnd())
    End Function
End Class
```

Dubbelklik op het formulier om de variabelen hold1 t/m hold5 bij het laden van het formulier de waarde “False” te geven (dat wil zeggen dat de dobbelstenen 1 t/m 5 in principe niet worden geblokkeerd).

Ook hier programmeren we de functie dobbelsteen().

Bij de “gooi” knop moet nu worden toegevoegd dat de beurten worden bijgehouden en dat er alleen maar met een dobbelsteen wordt gegooid als iemand niet op een hold-knop heeft gedrukt. De code wordt:

```
frmYathzeedobbelstenen (Declarations)
Private Sub btnGooi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGooi.Click
    beurt = beurt + 1
    lblAantalworpen.Text = beurt
    If hold1 = False Then txtDobbel1.Text = dobbelsteen()
    If hold2 = False Then txtDobbel2.Text = dobbelsteen()
    If hold3 = False Then txtDobbel3.Text = dobbelsteen()
    If hold4 = False Then txtDobbel4.Text = dobbelsteen()
    If hold5 = False Then txtDobbel5.Text = dobbelsteen()
    If beurt = 3 Then btnGooi.Visible = False
End Sub
```

Bij elk van de “hold” knoppen hoort de code:

```
frmYathzeedobbelstenen (Declarations)
Private Sub btnHold1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHold1.Click
    Select Case hold1
        Case Is = False
            txtDobbel1.ForeColor = Color.Red
            btnHold1.Text = "unhold"
            btnHold1.ForeColor = Color.Red
            hold1 = True
        Case Is = True
            txtDobbel1.ForeColor = Color.Black
            btnHold1.Text = "hold"
            btnHold1.ForeColor = Color.Red
            hold1 = False
    End Select
End Sub
```

Deze code zorgt ervoor dat de waarde van “hold” verandert op het moment dat er op de hold-knop wordt geklikt (zodat er niet meer gegooid kan worden of juist wel) en de kleuren veranderen.

Tenslotte hoort bij de “nieuw” knop de code die alles weer instelt als bij het begin:

```
frmYathzeedobbelstenen (Declarations)
Private Sub btnNieuw_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNieuw.Click
    btnGooi.Visible = True
    beurt = 0
    txtDobbel1.Text = ""
    txtDobbel1.ForeColor = Color.Black
    txtDobbel2.Text = ""
    txtDobbel2.ForeColor = Color.Black
    txtDobbel3.Text = ""
    txtDobbel3.ForeColor = Color.Black
    txtDobbel4.Text = ""
    txtDobbel4.ForeColor = Color.Black
    txtDobbel5.Text = ""
    txtDobbel5.ForeColor = Color.Black
    lblAantalworpen.Text = ""
    hold1 = False
    hold2 = False
    hold3 = False
    hold4 = False
    hold5 = False
    btnHold1.Text = "hold"
    btnHold1.ForeColor = Color.Black
    btnHold2.Text = "hold"
    btnHold2.ForeColor = Color.Black
    btnHold3.Text = "hold"
    btnHold3.ForeColor = Color.Black
    btnHold4.Text = "hold"
    btnHold4.ForeColor = Color.Black
    btnHold5.Text = "hold"
    btnHold5.ForeColor = Color.Black
End Sub
End Class
```

☞ Sla het project op als “Yahtzeedobbelstenen2”.

We hebben nu gezien dat we met de functie dobbelsteen() het programma in eerste instantie overzichtelijker hebben gemaakt maar dat het programma door het gebruik van de vele knoppen en tekstvelden minder overzichtelijk is geworden.

Niet alleen bij het vaker gebruiken van eenzelfde functie maar ook bij het vaker gebruiken van gelijksoortige objecten zoals tekstvelden en knoppen is het mogelijk compacter te programmeren. In hoofdstuk 7 en 8 wordt uitgelegd hoe je met behulp van arrays en indices (enkelvoud: index) de programmacode bij de Yahtzee dobbelstenen kunt inkorten.

6.4 Tel de ogen (functie zonder invoerparameter)

- ☞ Gebruik de functie `dobbelsteen()` om een project “Tel de ogen” te maken. Dit project telt het aantal keren dat een bepaald ogen aantal wordt gegooid.



6.5 De woordenteller (functie met invoerparameter)

- ☞ Schrijf een functie `woordenteller()` die wordt gebruikt om te tellen uit hoeveel woorden een zin bestaat. Noem het project “Tel de woorden”.

