



Programmeren in Microsoft Visual Basic 6.0, lessenserie voor het voortgezet onderwijs HAVO/VWO

© David Lans, Emmauscollege, Marnix Gymnasium
Rotterdam, januari 2004

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:

- De Dobbelsteen
- De Uitgebreide Dobbelsteen
- Yahtzee Dobbelstenen 1
- Yahtzee Dobbelstenen 2

Met dit hoofdstuk ben je, afhankelijk van je tempo, ongeveer 2 lesuren 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 “De Dobbelsteen” met het formulier “frmDobbelsteen” met daarop een knop “cmdGooi” en een tekstveld “txtDobbel”:



De bijbehorende programmacode is:

```
Project1 - frmDobbelsteen (Code)
(General) | dobbelsteen
Private Function dobbelsteen() As Integer
dobbelsteen = 1 + Int(6 * Rnd())
End Function
Private Sub cmdGooi_Click()
txtDobbel.Text = dobbelsteen()
End Sub
Private Sub Form_Load()
Randomize
End Sub
```

Zoals je ziet wordt de functie “dobbelsteen” bij de (General) declaraties opgegeven. Dat betekent dat je de functie overal in je programma kunt gebruiken. Als de functie dobbelsteen ergens in je programma wordt aangeroepen dan levert de functie als uitvoerresultaat een Integer, namelijk een getal 1 t/m 6.

Bij een druk op de knop “cmdGooi” wordt in het tekstveld “txtDobbel” een waarde geplaatst die de functie dobbelsteen op dat moment geeft.

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 invoer

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



☞ Maak een formulier “frmUitgebreideDobbelsteen” met een label “lblAantalvlakken”, een tekstveld “txtAantalvlakken”, een knop “cmdGooi” en een tekstveld “txtDobbel”.

```
Project1 - frmDobbelsteen (Code)
(General) | dobbelsteen
Private Function dobbelsteen(vlakken As Integer) As Integer
dobbelsteen = 1 + Int(vlakken * Rnd())
End Function
Private Sub cmdGooi_Click()
Dim aantalvlakken As Integer
aantalvlakken = txtAantalvlakken.Text
txtDobbel.Text = dobbelsteen(aantalvlakken)
End Sub
Private Sub Form_Load()
Randomize
End Sub
```

De programmacode luidt als volgt:

De functie dobbelsteen heeft nu een invoerwaarde “vlakken” nodig. Deze waarde bepaalt de mogelijke uitkomsten van de dobbelsteen.

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 het voorbeeld zijn zowel de invoerwaarde als de uitvoerwaarde van het type Integer. Elk ander type is natuurlijk mogelijk, afhankelijk van wat een functie moet doen.

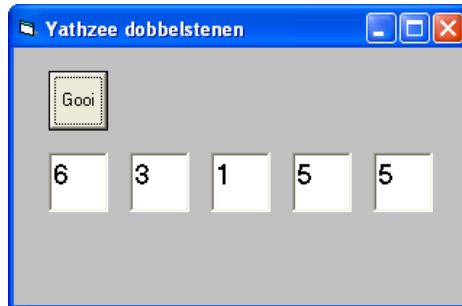
We noemen een waarde die als invoer aan een functie meegegeven dient te worden ook wel een **parameter** van de functie.

☞ Sla het project op als “De Uitgebreide Dobbelsteen”.

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 formulier “frmYahtzeedobbelstenen” met daarop een knop “cmdGooi” en een vijf tekstvelden “txtDobbel1” t/m “txtDobbel5”:



De bijbehorende programmacode is:

```
Project1 - frmDobbelsteen (Code)
(General) | dobbelsteen
Private Function dobbelsteen() As Integer
doppelsteen = 1 + Int(6 * Rnd())
End Function
Private Sub cmdGooi_Click()
txtDobbel1.Text = dobbelsteen()
txtDobbel2.Text = dobbelsteen()
txtDobbel3.Text = dobbelsteen()
txtDobbel4.Text = dobbelsteen()
txtDobbel5.Text = dobbelsteen()
End Sub
Private Sub form_load()
Randomize
End Sub
```

- Sla het project op onder de naam “De Yahtzee Dobbelstenen 1”.

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

- ☞ Breid het formulier uit met een aantal knoppen “cmdHold1” t/m “cmdHold5”, een label “lblWorp” met daarnaast een label “lblAantalworpen” dat het aantal worpen bijhoudt en een knop “cmdNieuw” waarmee na elk aantal worpen een nieuwe beurt kan beginnen.



Als (General) variabelen moeten worden opgegeven:

```
Dim hold1, hold2, hold3, hold4, hold5 As Boolean  
Dim beurt As Integer
```

Als (General) functie moet weer worden opgegeven:

```
Private Function dobbelsteen() As Integer  
dobbelsteen = 1 + Int(6 * Rnd())  
End Function
```

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

```
Private Sub cmdGooi_Click()  
beurt = beurt + 1  
lblAantalworpen.Caption = beurt  
If hold1 = False Then  
txtDobbel1.Text = dobbelsteen()  
End If  
..... idem voor hold2 t/m hold5  
If beurt = 3 Then  
cmdGooi.Visible = False  
beurt = 0  
End If  
End Sub
```

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

```
Private Sub cmdHold1_Click()  
If hold1 = False Then  
hold1 = True  
txtDobbel1.ForeColor = &HFF&  
cmdHold1.Caption = "unhold"  
Else  
hold1 = False  
txtDobbel1.ForeColor = &H80000008  
cmdHold1.Caption = "hold"  
End If  
End Sub
```

Bij de “nieuw” knop hoort de code:

```
Private Sub cmdNieuw_Click()  
cmdGooi.Visible = True  
txtDobbel1.Text = ""  
txtDobbel1.ForeColor = &H80000008  
txtDobbel2.Text = ""  
txtDobbel2.ForeColor = &H80000008  
txtDobbel3.Text = ""  
txtDobbel3.ForeColor = &H80000008  
txtDobbel4.Text = ""  
txtDobbel4.ForeColor = &H80000008  
txtDobbel5.Text = ""  
txtDobbel5.ForeColor = &H80000008  
lblAantalworpen.Caption = ""  
hold1 = False  
hold2 = False  
hold3 = False  
hold4 = False  
hold5 = False  
cmdHold1.Caption = "hold"  
cmdHold2.Caption = "hold"  
cmdHold3.Caption = "hold"  
cmdHold4.Caption = "hold"  
cmdHold5.Caption = "hold"  
End Sub
```

En ook nu ontbreekt de Randomize opdracht natuurlijk niet:

```
Private Sub form_load()  
Randomize  
End Sub
```



Sla het formulier op als “frmYahtzeedobbelstenen 2” en sla het project op als “Yahtzee Dobbelstenen 2”.

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”.

