VBA voor Doe het Zelvers deel 9

Handleiding van Helpmij.nl

Auteur: leofact

Augustus 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

VBA voor Doe het Zelvers is een reeks artikelen, bedoelt voor mensen die met VBA in Excel aan de slag willen om taken te automatiseren of om deze toegankelijk te maken voor gebruikers met weinig Excel kennis. VBA is een volwaardige programmeertaal met een woordenschat en een zinsopbouw (syntaxis). Om deze goed te kunnen leren wordt het aangeraden om hierover boeken te lezen. Bijvoorbeeld uit deze willekeurige selectie.

Vorige aflevering

In de vorige aflevering draaide het om het opzetten van lussen of Loops. Onder andere For Next en Do While werden behandeld. Daarbij werden tips gegeven om de Loops te versnellen. Als bijlage was een werkboek toegevoegd met de gebruikte code en een demo waarin het effect zichtbaar werd van de simpele tips om de Loops te versnellen.

In deze aflevering

Dit keer zetten we de foutafhandeling in het spotlicht. Misschien niet het meest sexy onderwerp, maar noodzakelijk wanneer je een toepassing wilt maken die stabiel, gebruiksvriendelijk en zonder onprettige verassingen werkt. De code en een demo vind je in dit werkboek.

Waarom foutafhandeling?

Een goede vraag. VBA heeft een ingebouwde foutafhandeling. Bij een fout stopt de compiler en wordt er een Messagebox weergegeven met een melding van het foutnummer en een korte omschrijving van de fout. Vanuit dit venster kan direct naar de fout worden genavigeerd en dat helpt bijzonder goed bij het opsporen en verhelpen van de fout. Gebruikers zullen echter niet echt geïnteresseerd zijn in mogelijkheden om een fout op te sporen of op te lossen. Zij willen met de toepassing werken, zonder dat er onverwachte dingen gebeuren met archaïsche meldingen die als abracadabra ervaren kunnen worden. De beste oplossing hiervoor is natuurlijk om een 100% perfecte toepassing te maken die nooit crasht en altijd juist reageert. Helaas is dat natuurlijk een utopie. Zelfs een bedrijf als Apple lukt dat niet. Bovendien heb je vaak met variabelen te maken waar je geen invloed op hebt. Dat kan allerlei verschillende oorzaken hebben, zoals onjuiste beveiligingsinstellingen, de verkeerde Excelversie of het missen van een noodzakelijke invoegtoepassing. Met foutafhandeling kan er voor gezorgd worden dat de gebruiker weet wat er aan de hand is en wat hij daar vervolgens aan kan doen. Verder kan de foutafhandeling er voor zorgen dat er correct kan worden doorgewerkt na een onverwachte fout. Als laatste kan met behulp van foutafhandeling feedback aan de gebruiker worden gegeven over verkeerd gebruik of foutieve invoer.

Soorten foutafhandeling

Voor mijzelf onderscheid ik drie soorten foutafhandeling:

1. Geprogrammeerd
   Hierbij wordt er een routine geprogrammeerd die in werking treedt bij een bepaalde fout en die er voor zorgt dat deze fout zo min mogelijk gevolgen heeft. Dat kan bijvoorbeeld gegevensvalidatie zijn of een Excelversie check. Verderop volgt een voorbeeld waarin uitgewerkt wordt dat een gebruiker een bepaald werkboek alleen kan gebruiken als de macro's zijn ingeschakeld en de beveiligingsinstellingen goed ingesteld staan.
2. Fout-gestuurde foutafhandeling
   Bij deze vorm wordt de foutafhandeling routine ingezet om te reageren op optredende fouten. Bijvoorbeeld het vergelijken van twee verschillende type variabelen of het zoeken naar iets wat niet aanwezig is.
3. Programma-gestuurde foutafhandeling
   Hierbij wordt bij een bepaalde gebeurtenis een fout gegenereerd welke vervolgens door de foutafhandeling wordt afgehandeld. Dit kan bijvoorbeeld worden ingezet om gebruikersinvoer te controleren. Voldoet de invoer niet aan de vereiste voorwaarden, dan kan VBA er toe gezet

worden om een fout te genereren. De fout kan dan via de foutafhandeling worden afgehandeld.

Van genoemde soorten foutafhandeling volgen er verderop voorbeelden.

MZ-Tools

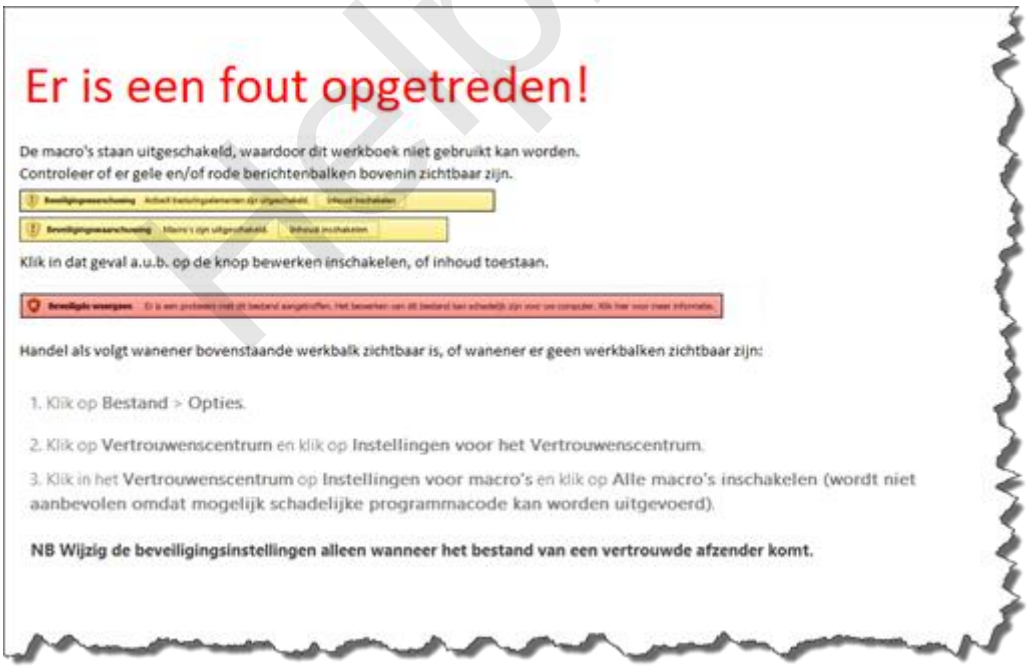
Dit handige hulpmiddeltje maakt het leven van een VBA-programmeur een stuk eenvoudiger. Deze tool is eerder aan de orde geweest in aflevering drie van deze reeks. Nu gaan we de tool intensief inzetten om het opzetten van de foutafhandelingsprocedures makkelijker te maken.
Wanneer je MZ-tools nog niet hebt geïnstalleerd hebt kun je het hier downloaden, of volg de aanwijzingen in deel 3.

Voorbeeld 1; uitgeschakelde macro's

Wanneer macro's worden ingezet in een werkboek kan de werking van het werkboek volledig verstoord worden als VBA zijn werk niet kan doen omdat de macro's uitgeschakeld zijn. In de nieuwere versies van Excel (vanaf 2007) kan dit ook veroorzaakt worden door de beveiligingsinstellingen. Zo kan bewerken uitgeschakeld zijn wanneer een werkboek wordt gedownload vanaf internet of vanuit een email.

Om te voorkomen dat er allerlei fouten in een werkboek sluipen zonder de ingeschakelde macro's is het mogelijk om het werkboek op het oog onbruikbaar te maken. Een methode hiervoor is om alle werkbladen onzichtbaar te maken voor het openen en een speciaal werkblad te tonen met informatie over de opgetreden fout en hoe te handelen om het werkboek in gebruik te nemen. Dat kan als volgt:

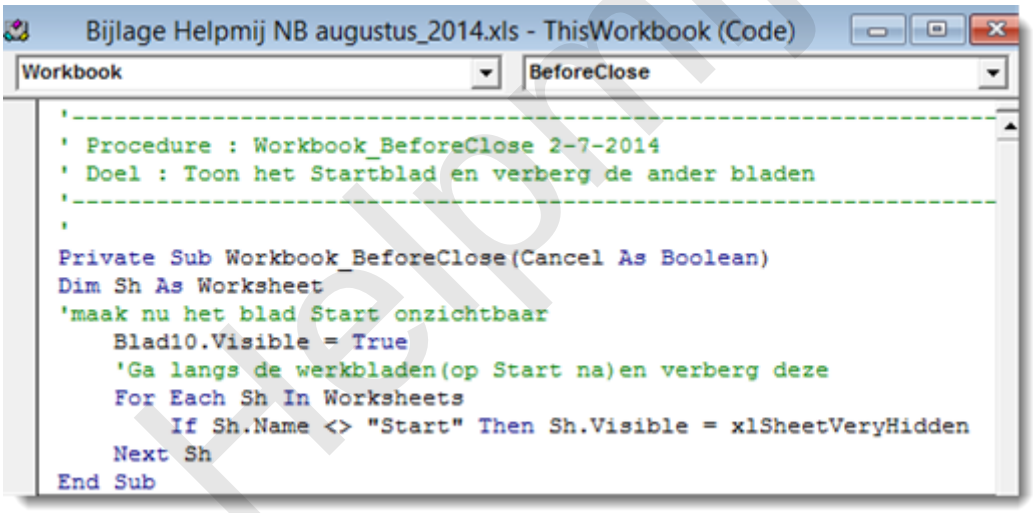
Begin met het maken van een blad waarin de gebruiker gewaarschuwd wordt en noem dit blad Start. Dat kan er zo uit zien:



Nu dienen bij het openen van het werkboek alle andere bladen te worden getoond en het blad start moet worden verborgen. Deze volgorde is essentieel. Excel staat niet toe dat alle bladen tegelijk onzichtbaar zijn. Het ligt voor hand dat dit in de Work_Open geregeld wordt. Bijvoorbeeld met deze code:

Nu worden mooi alle werkbladen getoond en de gebruiker kan aan de slag.

Er ligt echter nog één puntje. De bladen moeten natuurlijk wel onzichtbaar gemaakt worden bij het afsluiten. Anders kan er alsnog in de werkbladen worden gewerkt bij de volgende start van het werkboek. Daarvoor maken we eerst het Startblad zichtbaar en verbergen vervolgens de andere bladen met de volgende routine:



Wanneer er eenmaal een keer op macro's inschakelen is geklikt merk je de volgende keer weinig van de routines. Alleen moet er bij het afsluiten opgeslagen worden omdat er nog een verandering optreedt in het werkboek. Natuurlijk kan dit ook geautomatiseerd worden. Dan wordt er echter altijd opgeslagen. Ook als de gebruiker dit niet wil. Dat is dan weer te ondervangen met een keuzevenster, maar dit wordt nu verder aan de eigen fantasie overgelaten.

EZ-tools

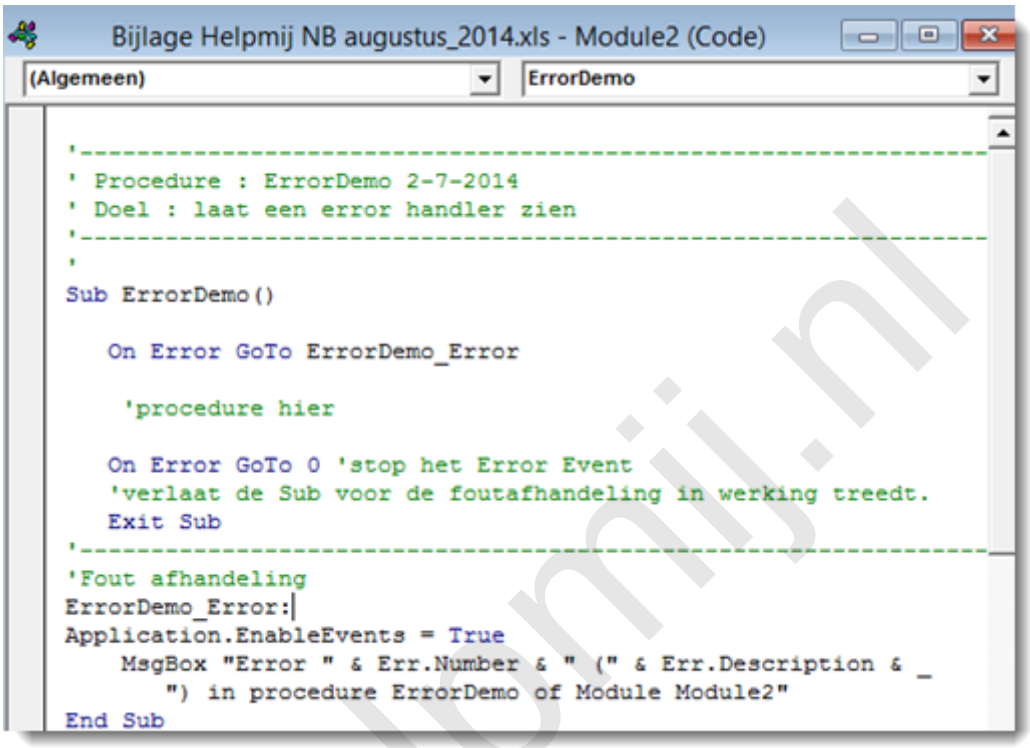
Nu is het tijd om EZ-Tools erbij te pakken.

Deze tool heeft de mogelijkheid om de foutafhandeling met één klik per routine te regelen. De tool kent een format welke is aan te passen via het optiemenu >Error Handler. Standaard wordt er bij een opgetreden fout een Messagebox weergegeven met daarin als variabelen het foutnummer, de korte omschrijving en de procedure waarin de fout is opgetreden. In plaats van een Messagebox kun je ook verwijzen naar een routine waarin de foutafhandeling wordt geregeld. Die routine kun je bijvoorbeeld een mail laten versturen naar jezelf, zodat je op de hoogte wordt gebracht van de foutmelding. Het versturen van mail wordt de volgende aflevering behandeld. Voor dit moment laten we de foutafhandeling zoals deze standaard staat voorgeprogrammeerd. Op één punt na. We zorgen ervoor

dat de events weer ingeschakeld worden met:

Application.EnableEvents = True

Hiermee zorgen we ervoor dat ook als de Events staan uitgeschakeld een fout er niet toe kan leiden dat het werkboek niet meer reageert. De code die nu door MZ-Tools wordt gegenereerd ziet er als volgt uit:



On Error is een event welke hier met een sprong door middel van Goto wordt gecombineerd. On Error kan ook losstaand worden gebruikt. Het wordt dan:

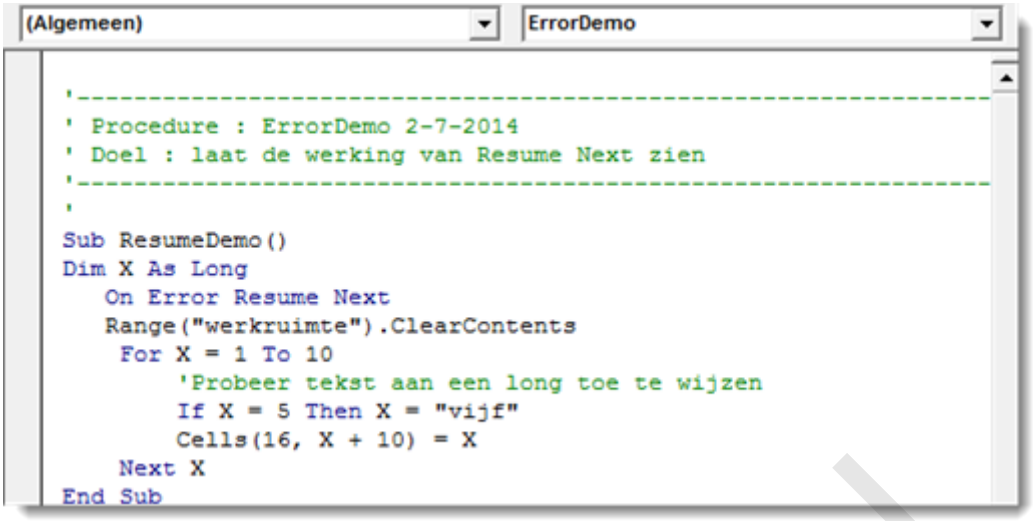
On Error Resume Next

Dit zorgt er voor dat VBA gewoon door gaat naar de volgende code stap na een fout. De procedure blijft gewoon door lopen nadat er fout is opgetreden. Nu wordt de foutieve stap echter overgeslagen. Of dit gewenst is hangt af van de functie van de procedure en het ervan.

Fout-gestuurde foutafhandeling

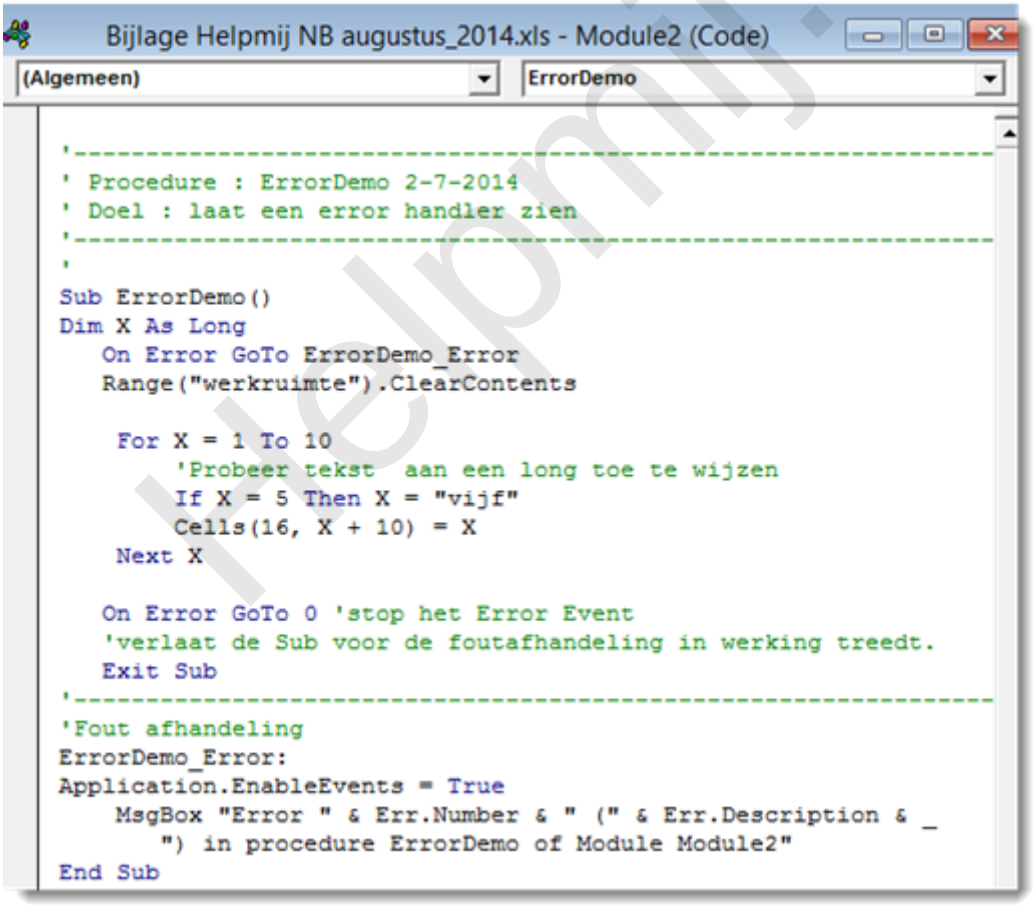
De Resume Next is een actie op een fout en dus een vorm van Fout-gestuurde fout afhandeling. Bijvoorbeeld:

Helpmij.nl



Nu loopt de code gewoon door. De stap X = “vijf” wordt overgeslagen en X blijft dus 5

Een voorbeeld met Goto is de volgende code:

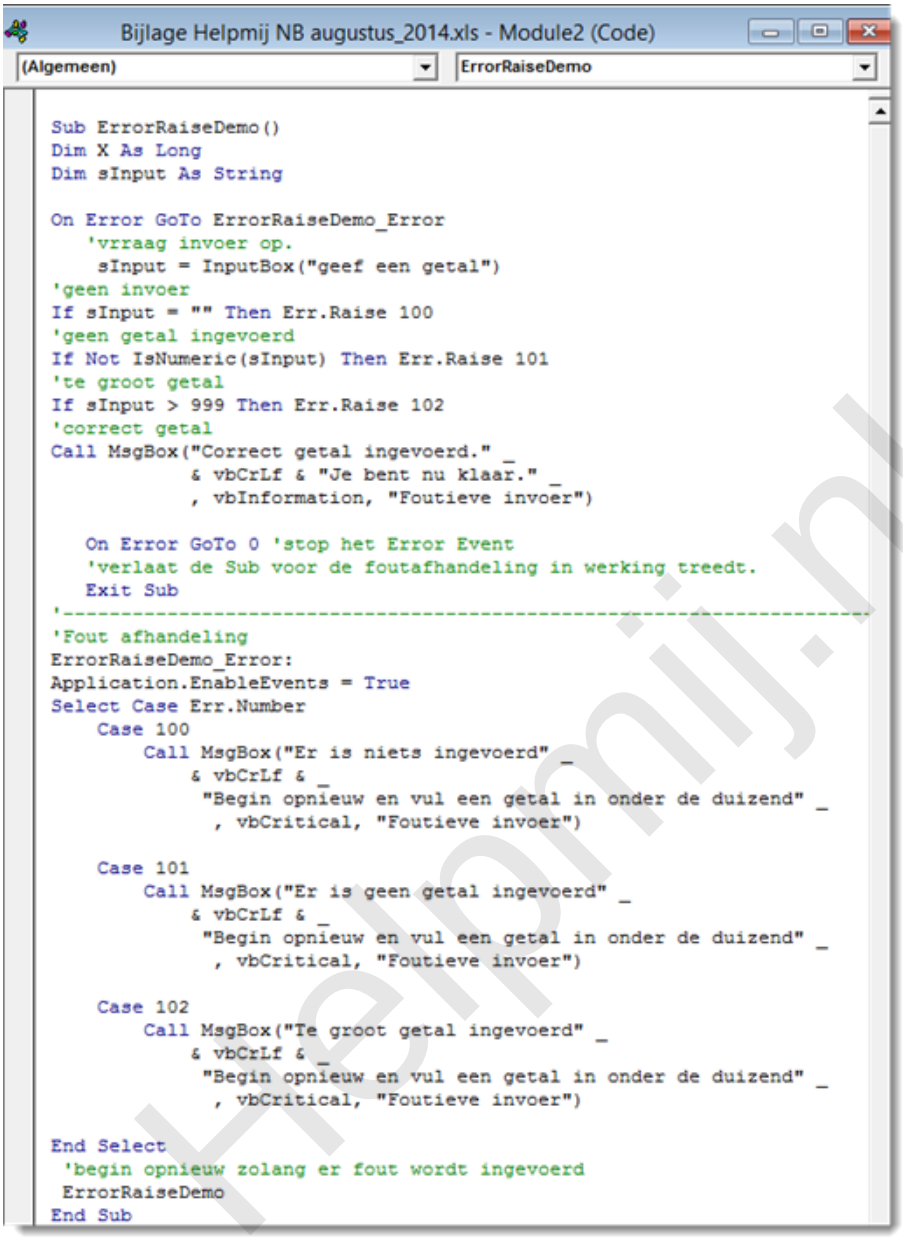


De code stopt bij de fout, maar de melding die volgt is nu zelf geconfigureerd in plaats van de standaard foutmelding.

Programma-gestuurde Foutafhandeling

Hierbij test je op bepaalde situaties en laat VBA de foutafhandeling dan afvuren door middel de

opdracht Err.Raise. Met Err.Number kun je dan de fout uitlezen:




```
Sub ErrorRaiseDemo()
Dim X As Long
Dim sInput As String

On Error GoTo ErrorRaiseDemo_Error
   'vrraag invoer op.
    sInput = InputBox("geef een getal")
'geen invoer
If sInput = "" Then Err.Raise 100
'geen getal ingevoerd
If Not IsNumeric(sInput) Then Err.Raise 101
'te groot getal
If sInput > 999 Then Err.Raise 102
'correct getal
Call MsgBox("Correct getal ingevoerd." _
            & vbCrLf & "Je bent nu klaar." _
            , vbInformation, "Foutieve invoer")

   On Error GoTo 0 'stop het Error Event
   'verlaat de Sub voor de foutafhandeling in werking treedt.
   Exit Sub
'---------------------------------------------------------------------
'Fout afhandeling
ErrorRaiseDemo_Error:
Application.EnableEvents = True
Select Case Err.Number
    Case 100
        Call MsgBox("Er is niets ingevoerd" _
            & vbCrLf & _
             "Begin opnieuw en vul een getal in onder de duizend" _
              , vbCritical, "Foutieve invoer")

    Case 101
        Call MsgBox("Er is geen getal ingevoerd" _
            & vbCrLf & _
             "Begin opnieuw en vul een getal in onder de duizend" _
              , vbCritical, "Foutieve invoer")

    Case 102
        Call MsgBox("Te groot getal ingevoerd" _
            & vbCrLf & _
             "Begin opnieuw en vul een getal in onder de duizend" _
              , vbCritical, "Foutieve invoer")

End Select
 'begin opnieuw zolang er fout wordt ingevoerd
 ErrorRaiseDemo
End Sub
```

Dit zijn natuurlijk maar eenvoudige voorbeelden, maar de mogelijkheden zijn legio.

Samenvatting.

Foutafhandeling zorgt voor een betere gebruikservaring. Er zijn verschillende soorten foutafhandeling. De geprogrammeerde waarbij je met een routine voorziene fouten opvangt. De fout-gestuurde afhandeling, waarbij er op een doorgaans onverwachte door VBA gegenereerde fout wordt gereageerd. En als laatste de programma-gestuurde afhandeling waarbij er in de procedure een fout wordt gegenereerd, welke vervolgens wordt afgehandeld met behulp van de foutafhandeling.

Volgende aflevering

De volgende aflevering behandelt het geautomatiseerd versturen van mail in verschillende vormen, met en zonder bijlages.