

-TNT Post
Port betaald
Port Payé
Pays-Bas

Nieuwsbrief

15^{de} jaargang december 2008

Nummer 4



HCC BASIC
gebruikersgroep

HCC - Postbus 6360 - 2001 HH Haarlem



Inhoud

Onderwerp

blz.

Wat zijn objecten?	4
In BASIC werken buiten de deuren.	7
Methoden (OOP)	8
Liberty Basic: MainWin en NoMainWin	10
Change	17
Verenigingswerk is mensenwerk!	19
Het Email verkeer van de BASIC- interessegroep	19
Meer mogelijkheden met tekst.	21
Werken met de printer en de bitmap.	24

Deze uitgave kwam tot stand met bijdragen van:

Naam	Blz
Marco Kurvers	4, 7, 8, 10, 21, 24
Gordon Rahman	10
Het bestuur	17, 19
Jan van der Linden, Webmaster	19



Contacten

Functie	Naam	Telefoonnr.	E-mail
Voorzitter	Willem Gobel	0118-850837	voorz@basic-gg.hcc.nl
Secretaris a.i.	Willem Gobel Heemskerckplein 27 4384 BD Vlissingen	0118-850837	secr@basic-gg.hcc.nl
Penningmeester	Piet Boere	0348-473115	penm@basic-gg.hcc.nl
Bestuurslid	Titus Krijgsman	075-6145458	t.krijgsman8@upcmail.nl
Redacteur	M.A. Kurvers Schaapsveld 46 3773 ZJ Barneveld	0342-424452	m.a.kurvers@hccnet.nl
Ledenadministratie	Fred Luchsinger	0318-571187	f.luchsinger@kader.hcc.nl
Webmaster	Jan van der Linden	071-3413679	j.vd.linden@kader.hcc.nl

<http://www.basic.hccnet.nl>



Redactioneel

In deze laatste nieuwsbrief op papier wil ik u laten zien dat Basic met de tijd is meegegaan. Niet alleen om de Basic versies zelf, maar om al het software waar Basic is terug te vinden. Het begon met VBA (Visual Basic for Applications) in de Microsoft Office programma's en later ook een BASIC versie om games te kunnen maken - DarkBASIC. Maar niet alleen om games te maken werden Basic versies bedacht, ook wilde men websites gaan programmeren zonder de 'Click en Drag to' methode te leren. Nu zijn er website editors waarmee we Basic kunnen gebruiken, o.a. in Visual Studio .NET en in de Crimson Editor. Meer daarover volgend jaar.

Marco Kurvers

Wat zijn objecten?

In Basic kunnen we eigen typen maken die werken als recordtypen. Onderstaand voorbeeld is zo'n type. Visual Basic .NET kent echter geen type, maar om toch zulke typen te kunnen maken wordt gebruik gemaakt van het sleutelwoord `Structure`.

```
Public Type TPlant  
    PlantNaam As String  
    PlantHoogte As Integer  
    PlantKleur As String  
End Type
```

De variabelen in een type worden velden genoemd en deze velden vormen de structuur van het (record)type.

Maar wat is nu eigenlijk een object?

Wanneer bovenstaand type TPlant vergeleken zou worden met een object, zal het enorme verschil dat er lijkt te bestaan tussen een (record)type en een object zijn teruggebracht tot wat het werkelijke verschil is: een kleine uitbreiding met grote functionele gevolgen. Het zijn juist deze gevolgen waardoor de uitleg wordt bemoeilijkt en een object zo ingewikkeld lijkt te zijn.

Een klasse

Om een object te kunnen gebruiken hebben we een klasse nodig. Onderstaand voorbeeld laat hetzelfde type zien, maar nu in vorm van een klasse. Bekijk eens het verschil tussen een type- en klasse declaratie.

Een aantal velden en methoden achter elkaar: FPlantNaam, FPlantHoogte en FPlantKleur.

- de methoden welke werken met (de inhoud van) deze velden, en de properties (eigenschappen) welke de toegang tot de inhoud van de velden regelen vormt de structuur van een klasse type.
- Het voorbeeld hieronder werkt niet in Visual Basic 6, maar het is mogelijk om soortgelijk zo'n klasse in versie 6 te kunnen maken door het klasse geraamte weg te laten en de properties te splitsen met de sleutelwoorden `Let` en `Get`. Lees meer daarover in de help van MSDN.

```
Public Class TPlant  
    Private FPlantNaam As String  
    Private FPlantHoogte As Integer  
    Private FPlantKleur As String
```

```

Public Sub New() 'andere programmeertalen noemen dit een constructor.
    FPlantNaam = ""
    FPlantHoogte = 0
    FPlantKleur = ""
End Sub

```

```

Public Property PlantNaam() As String
    Get 'in VB 6.0: noem dit Property Get en Property Let
        Return FPlantNaam
    End Get
    Set(ByVal Value As String)
        FPlantNaam = Value
    End Set
End Property

```

```

Public Property PlantHoogte() As Integer
    Get
        Return FPlantHoogte
    End Get
    Set(ByVal Value As Integer)
        FPlantHoogte = Value
    End Set
End Property

```

```

Public Property PlantKleur() As String
    Get
        Return FPlantKleur
    End Get
    Set(ByVal Value As String)
        FPlantKleur = Value
    End Set
End Property

```

```

End Class

```

In deze vergelijking valt duidelijk op dat een klasse type een grote uitbreiding is van het normale type.

Het grootste verschil dat opvalt, zijn de properties en de methode `New()` welke het klasse type zijn toegevoegd. Het klasse type heeft namelijk een probleem: de velden kunnen niet rechtstreeks benaderd worden.

Om toch gegevens aan een klasse toe te kunnen voegen, moet er een object worden gemaakt. We zouden dus kunnen zeggen:

```

Dim Plant As TPlant

```

Maar dat betekent niet dat we nu meteen het object Plant kunnen gebruiken. Het object werkt namelijk als een **pointer** die geheugenruimte reserveert zodat we de inhoud van de klasse kunnen gebruiken. We moeten dus de declaratie veranderen als:

```
Dim Plant As New TPlant 'voor VB 6.0
Dim Plant As TPlant = New TPlant 'werkt alleen in VB .NET
```

Bijvoorbeeld:

```
Dim MijnPlant As TPlant = New TPlant
MijnPlant.PlantNaam = "Roosje"
MijnPlant.PlantHoogte = 35
MijnPlant.PlantKleur = "Rood"
MijnPlant = Nothing
```

Wanneer er een waarde van een klasse wordt gebruikt dan wordt er van een object gesproken. Ofwel, wanneer er een instantie van een klasse wordt gemaakt dan wordt er een object gemaakt. Het object wordt echt gemaakt en daarvoor is er een speciale procedure nodig, de constructor `New()`. Deze speciale procedure doet het volgende: er wordt een geheugenruimte klaargezet, dan wordt er een kopie gemaakt van alle velden uit de declaratie van de klasse en vervolgens worden er verwijzingen, ofwel pointers, aangemaakt en wel één voor elke methode in de klasse.

Het resultaat hiervan is, dat elk gemaakt object andere gegevens kan opslaan maar dat voor het werken met deze gegevens wordt doorverwezen, door middel van de pointers, naar de in de klasse gedefinieerde methoden. Deze methoden kunnen ook de eigenschappen zijn, dus de properties. Deze methoden worden niet telkens weer opnieuw gedeclareerd en uitgewerkt. Dit heeft het grote voordeel dat de code optimaal hergebruikt wordt.

Een ander voordeel is, dat de inhoud van de methoden, dus in feite de werking van de procedures en functies, niet zichtbaar zijn in een gemaakt object. De werking wordt verborgen voor de gebruiker van het object, dit heet *information hiding* en is een van de kenmerken van Objectgericht Programmeren.

Naast het aanmaken van een geheugenruimte voor het object, is het noodzakelijk om ook een opruimer voor de geheugenruimte te hebben, wanneer het object niet meer nodig is. Wanneer dit niet gebeurt, dan is er al vrij snel zoveel geheugenruimte in gebruik dat er een tekort ontstaat waardoor een programma zal vastlopen. Een goed programma zal dan ook werken in drie stappen: eerst wordt er een object gemaakt dat nodig is om een actie uit te voeren, daarna wordt die actie uitgevoerd, en tenslotte wordt het object weer vernietigd zodat het gebruikte geheugen weer beschikbaar komt.

Het opruimen gebeurt met een andere speciale procedure, de destructor `Dispose()`. Deze is in Visual Basic in sommige klassen aan te roepen, maar het kan gebeuren dat het niet mogelijk is, zoals bovenstaand voorbeeld laat zien. De enige mogelijkheid is dan toch de oude manier, door gebruik te maken van het sleutelwoord `Nothing`.

Marco Kurvers

In BASIC werken buiten de deuren.

Steeds meer BASIC versies worden object georiënteerd. De naam BASIC, die we nu Basic noemen, lijkt daardoor groter te zijn geworden met alle mogelijkheden die we vroeger niet hadden.

Maar we moeten zeker niet op onze tong bijten met al die objecten. Haal eens al die object georiënteerde mogelijkheden eruit en zie wat er van Basic overblijft: de BASIC programmeertaal waar we altijd mee gewerkt hebben. We kunnen dus zeggen dat er met BASIC niets is veranderd. Alleen, door de techniek die de laatste tijd enorm is uitgebreid, heeft BASIC commando's en functies erbij gekregen om buiten de deuren toch nog de hardware aan te kunnen sturen. Deze commando's en functies zijn zeker geen standaard BASIC statements. Al die extra's kunnen we alleen vinden in de objecten, zoals een printer object of een beeldscherm object.

Poke statement en Peek functie

Wie zich nog de oude BASIC programmeertalen kan herinneren, weet misschien nog dat de hardware direct was aan te sturen. Geen objecten gebruiken, maar direct met de standaard BASIC statements en functies geluiden laten horen of een kleuren tintje op het beeldscherm weergeven. De twee mogelijkheden, POKE en PEEK(), waren in die tijd zeer krachtig en werden veel gebruikt.

Het nadeel van deze twee mogelijkheden was het gebruik van de geheugenadres die als parameter opgegeven moest worden. Onderstaande regel gaf bijvoorbeeld een kleuren vakje linksboven op het scherm in BASIC versie 2 van de Commodore 64.

```
POKE 1024,4
```

De volgende regel, die hetzelfde deed, was echter op de Commodore VIC 20 in dezelfde BASIC versie.

```
POKE 7680,4
```

Met de functie PEEK() konden we te weten komen wat de kleur was van een vakje op het beeldscherm. We moesten zelf de X en Y berekenen met de linker bovenhoek om niet per ongeluk het verkeerde geheugenadres te krijgen. Onderstaand voorbeeld geeft zo'n berekening:

```
A = PEEK(1024 + (40 * Y + X))
```

Deze formule berekent het resultaat van de beeldscherm coördinaat. Onthoud wel dat wanneer u de linker bovenhoek gebruikt, de X en Y allebei op nul zet. Anders blijft u niet linksboven op 1024. Dat geldt ook wanneer u op de eerste rij wilt blijven, maar niet linksboven wilt zijn. Voor Y geeft u nul op en voor X geeft u groter dan nul, maar kleiner dan 40.

In BASIC van de VIC 20 had u minder kolommen dan de C64, namelijk maar 22 kolommen. In BASIC versie 7.0 van de Commodore 128 kon zelfs met 80 kolommen worden gewerkt, maar een goed beeldscherm was dan wel aan te bevelen.

En tekst weergeven? Moesten we dan telkens maar weer die nare bovenstaande commando en functie gebruiken? Natuurlijk niet. BASIC had toen al commando's als PRINT, INKEY\$, INPUT, LINE INPUT en INPUT\$. Het nadeel was dat niet al deze commando's en functies in alle BASIC versies konden werken. De versies waren ontwikkeld als dialecten en dat gaf vaak problemen.

De techniek van nu

Nu hebben we BASIC op visueel gebied. Een omgeving om gemakkelijk Windows programma's te schrijven, maar hadden we het wel zo gewild? De programmeer problemen blijven bestaan en de BASIC code wordt moeilijker. Ook al kunnen we op een formulier eenvoudig de besturingselementen plaatsen; we moeten het alsnog programmeren om het te laten werken. Steeds meer werken we buiten de deuren van BASIC om de formulieren te programmeren.

Met 'buiten de deuren' bedoel ik dat we methoden (commando's en functies) hebben die niet van de Basic programmeertaal zelf zijn. Er zijn bestanden waar al deze methoden in worden bewaard. Door een object daarvan te gebruiken kunnen we die methoden aanroepen.

In de map System32 van Windows staan er een heleboel van in. We kunnen dus zeggen dat we met puzzel stukjes zitten te werken om een goed programma te kunnen maken

Resultaat

Ook al is het zo veranderd, toch kunnen we programmeren in Windows ook leren. We mogen van geluk spreken dat het de programmeertaal Basic is gebleven en geen ingewikkelde cryptische symbolentaal is geworden. Werken met de Windows onderdelen en formulieren is heus niet alleen maar voor gevorderden. Beginners kunnen hier ook mee aan de slag.

Marco Kurvers

Methoden (OOP)

In het objectgeoriënteerd programmeren is methode een synoniem voor member-functie. Het is dus een functie die behoort tot een klasse. Alle methodes van een object/klasse samen beschrijven zijn interface en zijn gedrag.

De term 'methode' is oorspronkelijk afkomstig uit de taal Smalltalk, maar is daarna veralgemeend voor al het objectgeoriënteerd programmeren.

Constructor

Een constructor is vergelijkbaar met een methode maar het levert geen waarde op en het kan niet overgeërfd worden. De scope van een constructor kan ook anders zijn: in bepaalde talen, zoals Java, kan de constructor niet zelf aangeroepen worden. Constructors hebben vaak dezelfde naam als de klasse waartoe ze behoren. In Basic is dat niet het geval, daar wordt een constructor altijd `New` genoemd. Het doel van een constructor is het initialiseren van de velden van een klasse en het vastleggen van een invariant voor de klasse (indien mogelijk, anders faalt het aanmaken van het object). Een correcte constructor levert een object in een 'geldige' toestand op.

In de meeste programmeertalen is overloading van de constructor mogelijk: een klasse kan meerdere constructors hebben met dezelfde naam maar met verschillende parameters. Dat is in Basic ook toegestaan. Onthoud dat het alleen in de .NET versie mogelijk is!

De tegenhanger van de constructor is de destructor die uitgevoerd wordt wanneer het object wordt vernietigd. Zoals in het eerste hoofdstuk beschreven, is het de destructor `Dispose`.

Methoden gebruiken

De methoden kunnen uit drie onderdelen bestaan: members, eigenschappen en gebeurtenissen (ook wel events genoemd).

Een member kan een procedure of een functie zijn.

De methoden kunnen op verschillende manieren worden aangemaakt in een klasse:

- door ze `private` te declareren;
- door ze `public` te declareren;
- door ze `protected` te declareren;
- door ze `friend` te declareren;
- door ze `protected friend` te declareren.

De meeste keren worden de methoden `private` en `public` gemaakt. De andere drie zijn alleen nodig wanneer de klassen elkaar erven of uit een andere library vandaan komen. Het is dan alleen maar mogelijk om `friend` methoden aan te roepen, die in een klassen library aanwezig zijn.

Wanneer een methode `protected` wordt gedeclareerd, is de methode alleen bekend in een klasse als de klasse, waar de methode in aangemaakt is, geërfd wordt.

Bij `protected friend` methoden geldt het voor allebei. Ze kunnen alleen via een geërfde klasse aangeroepen worden die aanwezig is buiten het project, zoals een klassen library in een DLL. `Friend` methoden kunnen dat ook, in een DLL staan, maar zonder het sleutelwoord `protected`. Een klasse in een project kan ook een klasse erven die in een ander bestand staat, zoals een DLL.

U kunt de methoden net zo gebruiken als dat u met uw eigen procedures en functies doet. Vergeet geen objectinstanties te creëren voordat u de objecten gaat gebruiken en vergeet ze ook niet te vernietigen, wanneer u ze niet meer nodig hebt.

Methoden die nergens anders in horen dan in een bepaalde klasse, en ook niet door objectinstanties aangeroepen hoeven te worden, kunt u private maken.

Marco Kurvers

Liberty Basic: MainWin en NoMainWin.

Liberty Basic kent twee manieren om weergave te gebruiken: tekstschermen en grafische schermen. Onderstaand scherm is bijvoorbeeld een tekstwindow, zonder grafisch, geen iconen en ook geen aparte fonts.



Om de tekst op het scherm te krijgen kunnen we gewoon `print` gebruiken.

```
print "Hello, World!"
```

Ook kan er om invoer worden gevraagd, waarbij een variabele nodig is.

```
Input "What is your name?"; name$
```

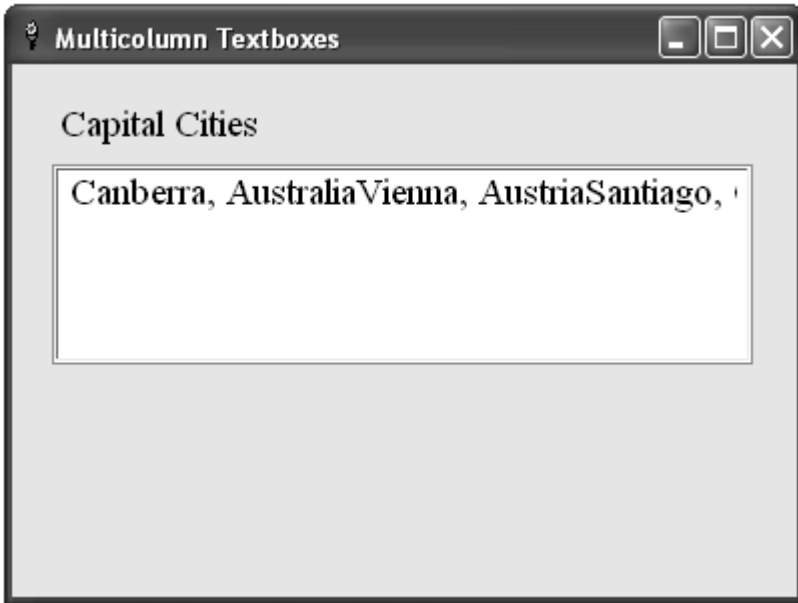
Willen we echter werken met de besturingselementen van Windows, dus grafisch bezig zijn, dan hebben we het statement `NoMainWin` nodig.

Onderstaand voorbeeld laat zien hoe we een tekstbox kunnen gebruiken met meer kolommen. Het eerste scherm laat een single line textbox zien.

Advertentie !

Handboek voor MS-DOS QBasic programmeurs
Microsoft Press, Kluwer 1991, 667 bladzijden.
Gratis af te halen of tegen verzendkosten.
Neem contact op via email: j.vd.linden@kader.hcc.nl

De VISUAL BASIC voor DOS programmeergids van Chris Arnoldus
Uitgever Addison Wesley 1993, 388 bladzijden



Figuur 1: Een singel tekstregel Textbox

Het type van Liberty Basic is de weergave van een eenlijnige tekst. Met wat veranderingen kan de textbox #main.col de tekst in kolomformaat weergegeven worden.

```
NoMainWin
```

```
'Ontwerp een GUI met een textbox
```

```
WindowWidth = 400
```

```
WindowHeight = 300
```

```
UpperLeftX = Int((DisplayWidth - WindowWidth) / 2)
```

```
UpperLeftY = Int((DisplayHeight - WindowHeight) / 2)
```

```
Statictext #main, "Capital Cities", 24, 20, 350, 24
```

```
Textbox #main.col, 20, 50, 350, 100
```

```
Open "Multicolumn Textboxes" for Window as #main
```

```
#main, "Trapclose XbyTrap"
```

```
#main, "Font Times_New_Roman 14"
```

```
'Creëer gegevens voor de kolom uitvoer
```

```
Dim capital$(20)
```

```
capital$ = ""
```

```
For i = 1 to 20
```

```
Read c$
capital$(i) = c$
capital$ = capital$;capital$(i);Chr$(13);Chr$(10)
Next i
```

```
#main.col, capital$
```

```
Wait
```

```
Sub XbyTrap handle$
```

```
Close #handle$
```

```
End
```

```
End Sub
```

```
Data "Canberra, Australia"
Data "Vienna, Austria"
Data "Santiago, Chile"
Data "Copenhagen, Denmark"
Data "Cairo, Egypt"
Data "London, England"
Data "Helsinki, Finland"
Data "Paris, France"
Data "Athens, Greece"
Data "Rome, Italy"
Data "Tokyo, Japan"
Data "Amsterdam, Netherlands"
Data "Abuja, Nigeria"
Data "Oslo, Norway"
Data "Lisbon, Portugal"
Data "Mogadishu, Somalia"
Data "Madrid, Spain"
Data "Bern, Switzerland"
Data "Damascus, Syria"
Data "Ankara, Turkey"
```

Advertentie !

Microsoft QuickBASIC compiler versie 2.01

Originale compiler met handboek inclusief twee originele 5,25 inch diskettes met de compiler.

Iets voor een verzamelaar of museum?

Gratis af te halen of tegen verzendkosten.

Neem contact op via email: j.vd.linden@kader.hcc.nl



Figuur 2: Meerregelige Textbox

De tekstuitvoer met meerdere regels worden verzorgd door twee functies. Elke regel krijgt aan het eind een carriage return, `Chr$(13)`, en een line feed, `Chr$(10)`. Zonder deze implementatie achter de array variabele `capital$(i)` krijgt men figuur 1, anders figuur 2.

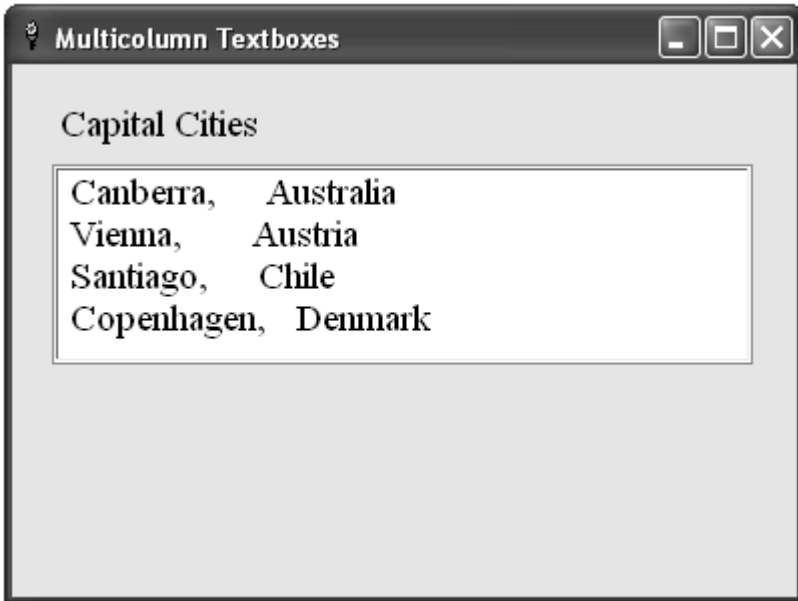
Het is ook mogelijk tabs te simuleren door gebruik te maken van spaties.

1. Bepaal de aantal spaties om de tweede kolom te plaatsen.
2. Bepaal de aantal tekens in de tekst van de eerste kolom.
3. Tel die aantal tekens met de aanwezige tweede kolompositie op.

Het resultaat van de gewijzigde `For` lus ziet u bij figuur 3.

```
Dim capital$(20)
capital$ = ""
For i = 1 to 20
  Read c$
  city$ = Word$(c$, 1)
  nation$ = Word$(c$, 2)
  capital$(i) = city$;Space$(14 - Len(city$));nation$
  capital$ = capital$;capital$(i);Chr$(13);Chr$(10)
Next i

#main.col, capital$
```



Figuur 3: Verschillende Font Breedtes

Toch geeft figuur 3 niet de juiste weergave zoals we willen. Wat scheelt eraan? Het probleem met gebruik van de regel `Space$(20 - Len(city$))` ligt bij de keuze van de font. Deze berekende formule is niet geschikt voor fonts die verschillende breedtes hebben, zoals het verschil tussen de breedte van een letter 'I' en een letter 'W'. Een font als 'Times New Roman' is niet acceptabel om de formule te gebruiken; wel fonts die vaste font breedtes hebben, zoals 'Courier New'. Het veranderen van de font naar een vaste font breedte zal meer duidelijke uitvoer geven.

Verander de regel: `#main, "Font Times_New_Roman 14"`

In de regel: `#main, "Font Courier_New 14"`

Figuur 4 geeft hiervan een voorbeeld.

Advertentie !

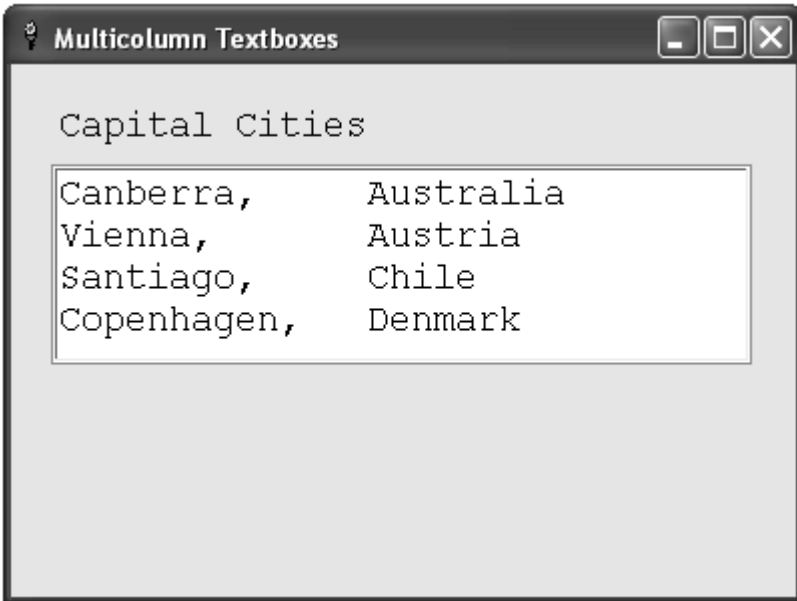
Microsoft BASIC Professional Development System versie 6.0

Wie heeft er belangstelling voor de originele handboeken van deze compiler?

Het zijn drie handboeken met alle informatie over deze professionele BASIC compiler.

Unieke naslagwerken voor de liefhebber. De originele diskettes met het programma zijn er niet meer bij. Ik doe ze gratis weg. Ze kunnen worden afgehaald of ze worden u toegezonden als u de portokosten betaalt.

Neem contact op via email: j.vd.linden@kader.hcc.nl



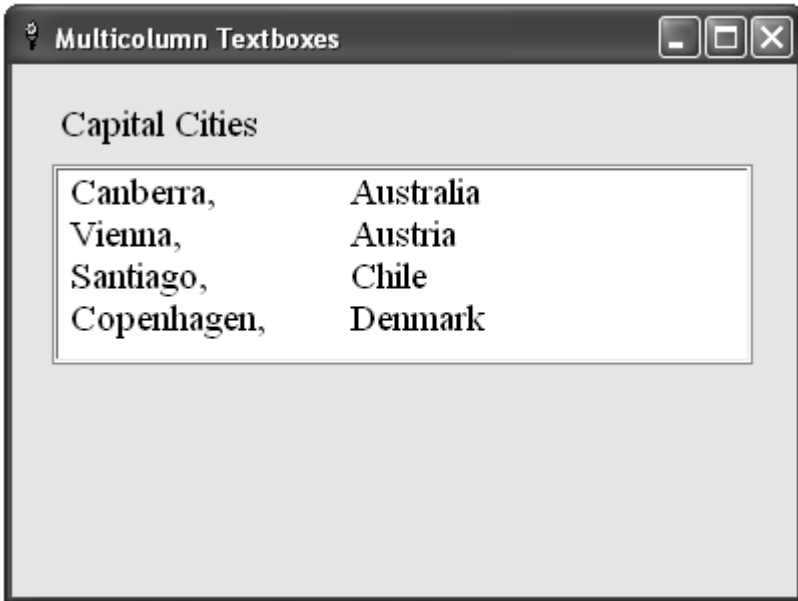
Figuur 4: Vaste Font Breedte

Een TabStop plaatsen met `_EM_SETTABSTOPS`

In Liberty Basic kan een tabstop worden gedefinieerd door een `_EM_SETTABSTOPS` bericht te zenden naar de textbox. Het plaatsen van een tabstop laat kolom formattering toe met fonts, van verschillende fonts zoals 'Times New Roman'. Tabstops zijn gedefinieerd als een set, de set zit in elkaar als een array. Liberty Basic ziet de array als een structuur. In dit makkelijke voorbeeld is slechts één tabstop toegekend, en die ene tabstop moest in elkaar als een structuur.

Plaats onderstaande code vóór de `For` lus, die bovendien ook een kleine wijziging krijgt. Zie het resultaat: figuur 5.

```
#main, "Font Times_New_Roman 14"  
  
'Vraag om de handle van de textbox  
hTB = hWnd(#main.col)  
  
'Definieer de structuur  
Struct TabStop, tabl as Long  
  
'Ken de waarde toe aan de variabele structuur  
TabStop.tabl.struct = 56
```



Figuur 5: Verschillende Font Breedte

```
'Definieer de tab toets [ Tab = Chr$(9) ]
LCTab$ = Chr$(9)

'Zend het bericht
CallDLL #user32, "SendMessageA", _
    hTB as ULONG, _ 'Handle van textbox
    _EM_SETTABSTOPS as Long, _ 'Windows Constant 203
    1 as Long, _ 'Aantal te zetten tabstops
    TabStop as Struct, _ 'De array van tabstops
    SetTabs as Long 'Geen-nul vergelijkbaar met succes

'Creëer gegevens voor kolom uitvoer
Dim capital$(20)
capital$ = ""
For i = 1 to 20
    Read c$
    city$ = Word$(c$, 1)
    nation$ = Word$(c$, 2)
    capital$(i) = city$;LCTab$;nation$
    capital$ = capital$;capital$(i);Chr$(13);Chr$(10)
Next i
```


#main.col, capital\$

Samengevat

Er zijn nog veel meer mogelijkheden, zoals meer tabstops instellen of een scrollbar in de textbox gebruiken.

Wilt u meer weten? Kijk eens op: <http://lbpe.wikispaces.com/Textbox+with+Columns>.
(Liberty BASIC Programmer's Encyc >> Textbox with Columns)

De website over deze encyclopedie is in het Engels.

Wanneer u alleen maar een tekst programma wilt schrijven zonder poespas, dan kunt u veel sneller in `mainwin` werken. Programma's zullen veel sneller draaien, omdat in `nomainwin` in Windows gewerkt wordt. Een handle bijvoorbeeld gebruiken kost ook tijd en geheugen. Als u geen Windows formulier nodig hebt om besturingselementen te plaatsen en alleen maar tekst, raad ik u aan om `nomainwin` niet te gebruiken.

Marco Kurvers (vertaald in het Nederlands)
Gordon Rahman (voor meer informatie)

Change

Het is een woord wat na de overwinning van Obama in Amerika de hele wereld over gaat.

Change is ook binnen de HCC al meer dan een jaar aan de gang.

De structuur en de bezetting is binnen de HCC volledig veranderd. Mocht er tot aan 1 september niet al te veel over gezegd worden in verband met de verbintenis met IDG de uitgever van Computer!Totaal, Tips&Trucs en Zoom, nu is dat over.

Zoals bekend is op die datum de samenwerking tussen IDG en de HCC beëindigd en gaan ze ieder weer hun eigen gang.

IDG probeert met een paar extra bladzijden de abonnees bij zich te houden voor de zelfde prijs maar zonder HCC lidmaatschap.

De HCC heeft het lidmaatschapsgeld nu vastgesteld op 24,50 waarnaast men een flinke korting kan krijgen bij een aanvullend computerblad gebaseerd op eigen interesse. Uit maar liefst 23 titels kan – of kon – gekozen worden. Vele leden van de HCC hebben inmiddels een proefblad binnen het profiel van hun opgegeven interesse ontvangen.

Maar er waren meer veranderingen:

- De afdelingen werden regio's
- De gebruikersgroepen binnen de HCC werden interessegroepen.

Maar het logo van de BASIC-interessegroep gaan wij vooralsnog niet veranderen, dat blijft



Aangezien de BASIC groep net als de andere groeperingen – bijna 60 bij elkaar – onderdeel zijn van de HCC, moeten wij mee veranderen. De hele financiële vlechting heeft hier een belangrijk aandeel in.

De BASIC groep geeft sinds vele jaren 4x per jaar een eigen BASIC blad uit, de Nieuwsbrief. Daarvoor betaalde u jaarlijks een extra bijdrage aan de HCC.

Met wat extra inkomsten en vergoedingen bleek het tot nu toe mogelijk om daarmee rond te komen.

Het nieuwe abonnementstarief van de HCC gaat uit van de eigen HCC organisatie met daarbij ondermeer hun digitale netwerkmogelijkheden voor alle leden. Extra uitgaven nodig voor bijvoorbeeld de nieuwsbrief van de BASIC-interessegroep moet door die groep nu geheel zelf gefinancierd worden. Dat betekent dat wij zelf van u een abonnementbijdrage moeten vragen, en innen, om door te kunnen gaan met de verspreiding van ons blad. Gezien alle veranderingen en de nieuwere mogelijkheden verwachten wij dat wat minder leden, dan in het verleden, zich op onze nieuwsbrief zullen abonneren.

Met een kostenbegroting voor volgend jaar zijn wij uitgekomen op 10 Euro per jaar wat wij nodig hebben om de nieuwsbrief op papier overeind te houden.

Als u van mening bent dat de nieuwsbrief als papieren uitgave moet blijven bestaan laat het ons dan weten.

Dat kan op 3 manieren:

1. Per E-mail aan penm@basic-gg-hcc.nl, dit heeft onze grote voorkeur:
Onder vermelding van uw naam, adres, woonplaats,
per E-mail ontvangt u dan ook informatie over het verdere verloop.
2. Telefonisch aan de penningmeester Piet Boere.
Normaal nummer: 0348-473115.
Mobiel nummer: 06-33981999.
3. Per brief aan de penningmeester BASIC-interessegroep,
Graaf Jan de Derde straat 4,
3417 TS MONTFOORT

Als er uiterlijk vóór 15 januari 2009 minimaal 250 bevestigende opgaven bij ons binnengekomen zijn, gaan wij door met de nieuwsbrief op papier. Dan ontvangt u een nota van 10 Euro per jaar en na betaling krijgt u de toekomstige nieuwsbrieven toegestuurd.

Bij minder belangstelling gaan wij geheel digitaal.

De inhoud van dit onderdeel, namelijk het wel en wee van onze website, staat elders in deze nieuwsbrief.

Het bestuur van de BASIC-gg

Verenigingswerk is mensenwerk!

Dit geldt ook voor de BASIC-groep.

Voor in dit boekje staan de namen van de mensen achter de BASIC-groep.

Daarnaast zijn er nog enkele leden die voor dit blad stukjes aanleveren.

Al met al is het wat mager met de bemensing van die taken.

Graag zouden wij er enige mensen bij hebben die meedenken en/of meewerken met het BASIC gebeuren, zodat wij met een wat gevarieerder aanbod aan onderwerpen ons blad en de site kunnen vullen.

Zonder regelmatige verversing van mensen en onderwerpen licht herhaling op de loer.

Daarom vragen wij aan u om eens bij uzelf na te gaan of u iets voor de BASIC-groep zou kunnen betekenen. Graag horen wij van U.

Op termijn ontkomen wij niet aan een herschikking van functies binnen onze club en hoe meer mensen hoe makkelijker wordt de verdeling.

Laat dit keer eens van u horen !

Een telefoontje of een mail is welkom.

0118-490168 of 0348-473115

Het bestuur van de BASIC-gg

Het Email verkeer van de BASIC- interessegroep

Begin november hebben alle leden van de BASIC interessegroep een e-mail gehad over de BASIC-cursus die in Haarlem wordt gegeven.

HEEFT U DEZE EMAIL NIET GEHAD? Leden die een ander emailadres hebben dan via hccnet.nl zullen waarschijnlijk geen e-mail ontvangen hebben. Dan kunt u zich opgeven bij de webmaster.

j.vd.linden@kader.hcc.nl

In het vervolg zullen wij u vaker een e-mail sturen over activiteiten van de BASIC interessegroep.

Door de bezuinigingen die de HCC heeft doorgevoerd zullen we meer via email met onze leden moeten gaan communiceren. Elders in de Nieuwsbrief kunt u lezen dat de HCC geen geld meer beschikbaar stelt voor de Nieuwsbrief. Mocht de papieren Nieuwsbrief niet meer haalbaar blijken dan gaan we over tot verspreiding van een versie via e-mail en de BASIC-site. Maar we doen ons uiterste best om de papieren versie voort te zetten.

Als u een profiel hebt aangemaakt op de site van de HCC ontvangt u alle berichten als BASIC-gg lid op uw e-mail adres. Wilt u die niet meer ontvangen dan kunt u zich binnenkort afmelden via uw profiel op de site www.hcc.nl

U logt in op: Mijn profiel.
Dan gaat u naar: Mijn gegevens.

Links onderin kunt u bij nieuwsbrieven groeperingen uw voorkeuren opgeven en aanpassen. Alleen is de optie tijdelijk buiten werking, maar deze komt binnenkort weer beschikbaar.

WEBSITE www.basic.hccnet.nl

Op onze vertrouwde website houden wij u op de hoogte van de BASIC interessegroep.

U kunt daar de listings downloaden van ons kwartaalblad en kennis nemen van andere zaken.

We beschikken nog over vraagbaken hoewel het aantal vragen aan hen sterk is afgenomen.

We willen in de toekomst meer interactief met de leden in contact treden met deze website.

We gaan een recente literatuurlijst over BASIC op de site plaatsen. Ook willen we de leden de gelegenheid geven zelf wat op de site te laten plaatsen. Een programmevoorbeeld of iets wat een lid zelf bedacht heeft. We zullen u op de hoogte houden van de ontwikkeling rond de website. Heeft u wensen of tips voor de website dan horen we dat graag van u via e-mail: j.vd.linden@kader.hcc.nl

**De webmaster
Jan van der Linden**

Advertentie !

Microsoft QuickBASIC compiler versie 4.5
Geheel compleet met diskettes en handboek.
Voor de liefhebber.
Gratis af te halen of tegen verzendkosten.

Werken met Visual BASIC 2.0 en 3.0
Een boek uitgegeven door Sybex, 1993 met 505 blad-
zijden
Gratis af te halen of tegen verzendkosten.

Neem contact op via email: j.vd.linden@kader.hcc.nl

Meer mogelijkheden met tekst.

Werken met tekst is informatie weergeven voor de computer gebruiker – of ook wel genoemd: de computeraar.

Tekst lezen en gebruiken, typen of schrijven, is niet alleen maar woorden en zinnen. Tekst kan uit hele alinea's bestaan. Een alinea dat dus uit meerdere zinnen bestaat. Meerdere alinea's staan weer in meerdere pagina's als de alinea's niet meer bij elkaar passen. Meerdere pagina's zijn bladzijden in een tijdschrift of zelfs uit een dik boek.

Als we op een Windows formulier een label component of een textbox component plaatsen, dan zetten we daar tekst in of laten we de gebruiker zelf tekst invoeren. Elke tekst component heeft zijn eigen kwaliteit. Het is bijvoorbeeld niet mogelijk in een label te schrijven, maar dat geldt ook voor de textbox. We kunnen daar wel in typen, maar we kunnen ook de kwaliteit veranderen. Zo is het mogelijk om de textbox op alleen-lezen te zetten, en kunnen we de textbox instellen op meerdere regels. Waarom het instellen op meerdere regels nodig zou zijn, is een onderwerp waar ik nu dieper op in ga.

Een voorbeeld stap voor stap proberen

Start BASIC die formulieren en componenten ondersteunt. Ik gebruik voor het voorbeeld Visual Basic.

Maak of gebruik een formulier en maak of plaats een textbox op het formulier. Gebruik als voorbeeld 'Figuur 1' uit het hoofdstuk 'Liberty Basic' - maak dus de textbox wijd genoeg om meerdere woorden in te kunnen typen.

Start het programmaatje en het formulier met het tekstvak verschijnt op het scherm. Typ wat tekst in het ruime tekstvak en druk daarna op de Enter toets.

Wat gebeurt er? Helemaal niets! De cursor blijft op zijn plaats staan en gaat niet naar de volgende regel. Sterker nog, die volgende regel is er helemaal niet en dat kunt u ook zien door de regel tekst langer te maken dan de lengte van het tekstvak, want de tekst zal verschoven worden. Ook kunt u zien dat er ook automatisch geen schuifbalk zal verschijnen.

Gelukkig zijn al deze nadelen te veranderen. De textbox heeft eigenschappen die we zo kunnen veranderen, dat we haast zelf een tekstverwerker kunnen bouwen. In Visual Basic

6.0 kunnen we dat al programmeren, maar in Visual Basic .NET heeft de textbox nog veel meer mogelijkheden. Eén van die mogelijkheden heb ik al in een paar voorgaande nieuwsbrieven behandeld: de `Lines` eigenschap. Maar eerst gaan we kijken hoe we de standaard eigenschappen kunnen gebruiken.

De eigenschappen van de textbox

Laten we eens wat eigenschappen bekijken.

Naam	Omschrijving
Alignment	Hiermee kunt u tekst links, gecentreerd of rechts op het tekstvak uitlijnen.
BackColor	Het is mogelijk het tekstvak in te kleuren. U kunt de standaard Windows kleuren gebruiken, maar u kunt ook zelf een kleur definiëren.
BorderStyle	Als u deze instelt op None dan heeft het tekstvak geen diepte.
Font	Kies hier een ander lettertype. U kunt maar één font per tekstvak instellen.
ForeColor	Kies hier een andere tekstkleur.
Locked	Bepaal hier of de gebruiker in het tekstvak mag typen of niet.
MaxLength	Geef op hoeveel tekens de gebruiker in mag typen. Lengte nul betekent een oneindige lengte.
MultiLine	Bepaal hier of de gebruiker meerdere regels mag gebruiken. De Enter toets zal dan ook functioneren.
ScrollBars	Wanneer de tekst niet in het tekstvak past dan kunt u de schuifbalken instellen. Zolang er ruimte is staan de schuifbalk(en) uit. Ze zullen automatisch ingeschakeld worden zodra de tekst gaat schuiven.
Text	De tekst die ingevoerd wordt, zal worden toegekend aan die eigenschap. Hebt u echter de eigenschap Locked op True gezet, dan bent u als programmeur de enige die de tekst mag invoeren. De gebruiker zal dan uw tekst lezen.
ToolTipText	Voor een duidelijke info, over wat het doel is van het tekstvak, zal de gebruiker daardoor meer weten en de juiste tekst invoeren.

Door deze eigenschappen te gebruiken tijdens het programmeren, kunnen we een goed tekstprogramma maken voor de gebruiker. Het nadeel is dat we toch een eigenschap missen, en dat is (Custom). Als we die willen gebruiken, hebben we een ander component nodig: de `RichTextBox`.

Met de `RichTextBox` kan de tekst op vet, schuin en onderstrepen worden gezet, en niet alleen de hele tekst, nee, zelfs deels van de tekst. We kunnen een menu maken met die opties of een werkbalk en we kunnen de `RichTextBox` laten reageren op de gebruikersacties. De gebeurtenissen die in de menu-items of knoppen uitgevoerd worden, moeten de juiste instellingen geven op de `RichTextBox` eigenschappen.

Visual Basic .NET en meer mogelijkheden met Lines

De vorige keer heb ik verteld wat de eigenschap doet en heb ik laten zien hoe we de eigenschap kunnen gebruiken.

Hoewel het gebruik van meerdere regels en het zoeken naar tekst wat beter is geworden, heeft de eigenschap helaas niet de functionaliteit die in een echt tekstvenster thuishoort, zoals het direct wijzigen per regel en deels kunnen zoeken naar tekst.

Onderstaand voorbeeld laat de code zien met een formulier en een textbox en het gebruik van de eigenschap `Lines()`, en eigenlijk kunnen we het bijna geen eigenschap noemen!

```
Dim I As Integer
Dim s(25) As String
Dim sget, f As String
```

```
For I = 0 To 25
    s(I) = "Het getal is: " & (I + 2).ToString()
Next
TextBox1.Lines() = s
sget = TextBox1.Lines().GetValue(2)
f = TextBox1.Lines().IndexOf(s, "Het getal is: 24") _
    .ToString()
TextBox1.Lines().SetValue("Gevonden: " & f, 0)
```

Plaats het voorbeeld in de procedure `Form_Load`, plaats een breakpoint op de laatste regel en start het programma.

Wijs met de muis variabele `f` aan en de waarde is 22. Dat is juist.

Wijs met de muis variabele `sget` aan en de waarde is "Het getal is: 4". Dat is ook juist.

Ga verder met debugging...

Er wordt keurig een lijstje van getallen in het tekstvak gezet, maar zonder de 'Gevonden' regel die we op de eerste regel willen hebben. Fout gedacht! Wat is er aan de hand? Laten we eens de code goed bekijken.

Ten eerste: we moeten eerst een array van strings declareren en in een `For` lus vullen. Pas daarna kunnen we de `Lines()` eigenschap vullen met de elementen van de array. Dat is het eerste nadeel: het is niet toegestaan direct in de `For` lus regel voor regel aan de `Lines()` eigenschap toe te kennen. U kunt alleen array's toekennen.

Ten tweede: de functie `GetValue()` werkt prima en hiermee kunnen we elke regel opvragen en aan een variabele toekennen. Dan zou de methode `SetValue()` zeker het tegenovergestelde werken, namelijk dat we een regel tekst in de opgegeven index kunnen plaatsen. Echter zien we nergens de waarde "Gevonden: 22" in de textbox. De methode doet dus niet wat de functie doet: werken met elementen, nee, erger nog: de methode vraagt om de

index van de array. De waarde 0 die in de tweede parameter van `SetValue()` staat, is dus het eerste array index, ofwel: de eerste dimensie.

Ten derde: we kunnen met de methode `IndexOf()` alleen zoeken naar complete regels. Verander de code van die methode in:

```
f = TextBox1.Lines().IndexOf(s, "24").ToString()
```

Zet weer een breakpoint op de laatste regel en start het programma.

Wijs met de muis weer op variabele `f`. Nu is de waarde: -1

De variabele, of beter gezegd: de methode, doet nu net alsof de tekst niet in het tekstvak bestaat. Als complete tekstregel is dat ook zo, maar het was toch veel mooier geweest als we ook naar delen van tekst konden zoeken.

Resultaat

Ook al zijn er meer eigenschappen en methoden bijgekomen, toch kunnen we nog lang niet alles direct doen wat we willen doen. Voor hergebruik van de code is het zelfs beter om nieuwe onderdelen zelf te maken. Probeer bijvoorbeeld eens een `Find()` methode te maken die ook naar tekst zal zoeken, maar die dan ook delen van tekst zal ondersteunen. Ik kan u vertellen: programmeren is leuk, want u bent eigen meester van de computer. Het is een gebouw dat u zelf groter kunt maken, kunt opmaken met nieuwe kleuren, andere vloeren en plafonds, een zolder erbij! Experimenteer eens met de tekst en ontdek de vele mogelijkheden die er zijn.

Marco Kurvers

Werken met de printer en de bitmap.

In Visual Basic is er een mogelijkheid om te kunnen programmeren en uitvoer naar de printer te sturen. Het enige wat u nodig heeft is het object `Printer`.

Het object heeft eigenschappen waarmee we de uitvoer kunnen bepalen. Zie meer daarover op <http://www.devarticles.com/c/a/Visual-Basic/Printing-With-Visual-Basic/3/>

De site is helaas wel Engelstalig, maar ik zal toch hier het een en ander samenvatten in het Nederlands.

De Print methode

Met de `Print` methode plaatsen we tekst op het scherm. Meestal met het object `Debug`. Dit object kent de `Print` methode ook, want als we na het object de punt typen, zien we twee methoden verschijnen: `Assert` en `Print`. Het object `Printer` lijkt de methode `Print` niet

te kennen, omdat die niet in de lijst staat. Toch is het mogelijk om tekst af te kunnen drukken, door gewoon code te typen als:

```
Printer.Print "Hallo allemaal"
```

Ook is het mogelijk grafisch te tekenen en dat naar de printer te sturen. De oude bekende methoden, zoals `Line` en `PSet` zijn algemeen bekend. Vergeet niet het printer object ervoor te zetten.

Er kan ook gewerkt worden met de pagina's van de printer. Het object kent de methoden `NewPage`, `Page` en `EndPage`. Kijk op de aangegeven website voor meer informatie.

De methode `PaintPicture`

Met de methode `PaintPicture` kunnen we bitmaps afdrukken op papier. Het kunnen bitmaps zijn die uit een `Picture` object vandaan komen, maar het kan ook een bitmap bestand zijn dat geladen wordt en direct naar de printer wordt gestuurd.

De regel:

```
Printer.PaintPicture Picture1.Picture, 1000, 1000
```

Stuurt de `Picture` van het opgegeven `Picture` object naar de printer met 1000 twips, als dit de juiste scale mode is.

De regel:

```
PaintPicture LoadPicture ("Voorbeeld.jpg"), 1000, 1000
```

Laadt het bestand `Voorbeeld.jpg` en plaatst het in een pagina. Zonder gebruik van het `Printer` object zal alles namelijk in de pages van het object worden bewaard en niet direct worden afgedrukt.

De regel:

```
Printer.PaintPicture Picture1.Picture, 1000, 1000, 2000, 2000
```

Verandert de grootte van de `Picture` in `Picture1` naar 2000x2000 punten en drukt het af.

De methode heeft zoveel mogelijkheden dat we prima fototabellen en fotolijsten af kunnen drukken. Zoals u kunt zien hebben we alleen maar `PaintPicture` nodig om de bitmap in een pagina op te slaan of direct naar de printer te sturen.

Onthoud dat `PaintPicture` altijd om een bitmap vraagt, directe bitmap uit het bitmap object of door een bestandsnaam op te geven. De methode ondersteunt niet de bestandsnaam zelf, daar zorgt `LoadPicture` voor.

Nogmaals: voor meer informatie over printen met Visual Basic; raadpleeg aangegeven website. Kunt u de website niet vinden; ga dan naar de hoofdsite www.devarticles.com. Bovendien kunt u via de hoofdsite vele andere programmeertechnieken vinden en veel hulp. Helaas is alles wel Engelstalig, maar geen nood: u kunt alle vragen aan mij stellen.

Marco Kurvers

Inhoud kwartaaldiskette

2008 – 4

De kwartaaldiskette bestaat uit:

Bestandsnaam	Type
Wat zijn objecten	RTF
BASIC buiten de deuren	RTF
Methoden OOP	RTF
LB mainwin nomainwin	DOC
Change	RTF
Verenigingswerk mensenwerk	RTF
Email verkeer BASIC groep	RTF
Mogelijkheden tekst	RTF
Printer en de bitmap	RTF

Indien niet alle bestanden op één diskette passen, kan het uit meerdere diskettes bestaan.

Zie aanwijzing op bladzijde 27 voor het bestellen van bovenstaande diskette

De cursussen worden uitsluitend geleverd met cursusboek
Alle prijzen zijn inclusief verzendkosten voor Nederland en België:

Cursussen

Qbasic: Cursus, lesmateriaal en voorbeelden op CD-ROM € 6,00 voor leden. Niet leden € 10,00.

QuickBasic: Cursusboek en het lesvoorbeeld op diskette,
€ 11,00 voor leden. Niet leden € 13,50

Visual Basic 6.0: Cursus, lesmateriaal en voorbeelden op CD-ROM,
€ 6,00 voor leden. Niet leden € 10,00

Basiscursus voor senioren, Windows 95/98,

Word 97 en internet voor senioren, (geen diskette). € 11,00 voor leden. Niet leden € 13,50

Software

Catalogusdiskette,

€ 1,40 voor leden. Niet leden € 2,50

Overige diskettes,

€ 3,40 voor leden. Niet leden € 4,50

CD-ROM's,

€ 9,50 voor leden. Niet leden € 12,50

Hoe te bestellen

De cursusboeken en/of diskettes of CD-ROM kunnen worden besteld door storting van het verschuldigde bedrag op :

Postbanknummer 7227036

HCC BASIC GG

Montfoort

Onder vermelding van de juiste cursus. Wanneer u elektronisch bankiert vermeld dan ook uw adres bij opmerkingen. Uw adres wordt namelijk niet automatisch meegezonden. Houdt u rekening met een leveringstijd van ongeveer twee weken.

Zoals reeds eerder vermeld kunt u vanaf uitgave 1 van 1998, de teksten en broncodes van de Nieuwsbrieven bestellen op diskette. Ze zijn ook te downloaden van onze website. De diskettes worden bij tijd en wijlen aangevuld met bruikbare hulpprogramma's, informatieve artikelen en voorbeeldprogramma's.

Aanvullingen zijn altijd in subdirectories geplaatst.

Op de catalogusdiskette staat een korte, maar duidelijke beschrijving van elk programma.



Vraagbaken



De volgende personen zijn op de aangegeven tijden beschikbaar voor vragen over programmeerproblemen. Respecteer hun privé-leven en bel alstublieft alleen op de aangegeven tijden.

Waarover	Wie	Wanneer	Tijd	Telefoon	Email
Liberty Basic	Gordon Rahman	De hele week	19-23	(023) 5334881	grahman@planet.nl
MSX-Basic	Erwin Nicolai				basic@lordthanatos.com
PowerBasic CC	Fred Luchsinger				f.luchsinger@kader.hcc.nl
QBasic	Jan v.d. Linden				j.vd.linden@kader.hcc.nl
QuickBasic					

Waarover	Wie	Wanneer	Tijd	Telefoon	Email
Visual Basic voor Windows Visual Basic .NET	Jeroen v. Hezik	ma. t/m zo.	19-21	(0346) 214131	j.a.van.hezik@kader.hcc.nl
Basic algemeen, zoals VBA Office Web Design, met XHTML en CSS	Marco Kurvers	ma. t/m vr. za. t/m zo.	19-23 13-17	(0342) 424452	m.a.kurvers@hccnet.nl m.a.kurvers@tronicasoftware.nl