

Inleiding

Visual Basic 2013 is een objectgeoriënteerde programmeertaal. De basiselementen van de programmeertaal zijn de begrippen klasse en object. De taal ondersteunt inkapseling (encapsulation), enkele overerving (inheritance) en polymorfisme (polymorphism). Het zijn noodzakelijke voorwaarden om te kunnen spreken van een volwaardige objectgeoriënteerde programmeertaal. En Visual Basic 2013 is een volwaardige objectgeoriënteerde programmeertaal.

In het eerste deel *Leren programmeren in Visual Basic 2013 1/2* hebben we de begrippen klasse en object intuïtief aangebracht. We hebben in dat deel niet de klemtoon gelegd op het objectgeoriënteerd programmeren maar op de structuren die in elke programmeertaal worden gebruikt.

Dit tweede deel van de cursus vertrekt van objectgeoriënteerd programmeren en illustreert de verschillende principes van objectgeoriënteerd programmeren in Visual Basic 2013, bv. het aanmaken van eigen klassen, overerving en polymorfisme. We gaan in dit deel ook in op meerdere gegevensstructuren, het werken met bestanden, de communicatie met andere programma's, ...

We gebruiken ook in dit deel de geïntegreerde ontwikkelomgeving Microsoft Visual Studio 2013 met het .NET Framework 4.5.1.

Net als het eerste deel is dit een doe-boek. Je leert de materie terwijl je voor de computer zit. Je werkt de voorbeelden mee uit en je maakt de oefeningen. Het is geen leesboek, ook geen manual...

Ik hoop dat je na deze cursus de programmeermicrobe te pakken hebt en dat je verder duikt in de wereld van programmeren, Visual Basic 2013 én .NET.

Indien je opmerkingen hebt die een latere uitgave kunnen verbeteren, kan je die steeds kwijt aan de auteur.

Roger Frans
februari 2015

Afspraken

In de volgende hoofdstukken leren we aan de hand van concrete voorbeelden Visual Basic 2013 kennen. We veronderstellen dat je de voorbeelden op de computer uittest.

De acties die je als gebruiker stapsgewijs moet volgen, worden als volgt genoemd.

- Je geeft je wachtwoord in. Dit wachtwoord wordt niet op het scherm getoond. Je krijgt enkel sterretjes (*) te zien.
- Je klikt op de knop *OK* om te bevestigen.

Het pijltje geeft aan dat er van jou iets verwacht wordt. Na het pijltje wordt omschreven wat je moet doen.

Nieuwe begrippen worden **vet** afgedrukt. Namen van menukeuzen, knoppen, e.d. worden *schuin* afgedrukt.

Indien je meerdere menukeuzen moet maken, duiden we dit als volgt aan:

- Kies *File / New Project*.

Dit betekent dat je klikt op de menukeuze *File* en dat je daarna klikt op de menukeuze *New Project*. Je mag de menukeuzen natuurlijk ook met sneltoetsen openen.

Indien je tekst letterlijk moet intypen, wordt dit in een ander lettertype weergegeven:

- Typ Dit moet je letterlijk intypen.

Je kan de voorbeeld- en oefenbestanden downloaden op de website van de uitgeverij van dit boek, Campinia Media.

- Surf naar <http://www.campiniamedia.be>.
- Kies *Fondslijst*.
- Typ `Visual Basic` in het zoekvak en klik op *Zoek*.
- Selecteer het boek door op de naam van het boek te klikken.

Je ziet bij de beschrijving van het boek een sectie *Downloads*. Je vindt er enkele zip-bestanden. Zip-bestanden kunnen meerdere bestanden groeperen.

- Download de voorbeeldbestanden door op de koppeling *Voorbeelden Visual Basic 2013-2.zip* te klikken.

Je volgt verder de instructies van je browser om het bestand op te slaan. Daarna pak je het bestand uit. Je kan dit bv. doen door in het linkerdeel van Verkenner op het bestand te klikken en de bestanden in het rechterdeel naar een map te kopiëren.

Je plaatst de bestanden in de map *Vb2013_2_Vbn*, bv. rechtstreeks onder de hoofdmap `C:\`. De map bevat bestanden die je nodig hebt om de voorbeelden en oefeningen in deze cursus te maken. We noemen deze map verder de **voorbeeldmap**.

De voorbeelden en oefeningen die je in deze cursus zelf uitwerkt, plaats je in een map met de naam *Vb2013_2_Oef*. We spreken in het boek van de **oefeningenmap**.

Inhoudsopgave

INLEIDING	1
AFSPRAKEN	2
INHOUDSOPGAVE	3
1 OOP EN .NET	11
1.1 Inleiding	11
1.2 Kwaliteit van een programma	11
1.2.1 Correctheid en bruikbaarheid	12
1.2.2 Robuustheid	12
1.2.3 Portabiliteit.....	12
1.2.4 Leesbaarheid en uitbreidbaarheid	12
1.2.5 Modulariteit	13
1.2.6 Efficiëntie	13
1.2.7 Herbruikbare componenten	13
1.3 Objectgeoriënteerd programmeren	13
1.3.1 Een voorbeeld uit het 'echte' leven	13
1.3.2 Klasse (Class).....	13
1.3.3 Object (Object).....	14
1.3.4 Berichten (Message).....	15
1.3.5 Inkapseling (Encapsulation).....	15
1.3.6 Overerving (Inheritance).....	16
1.3.7 Polymorfisme (Polymorphism)	16
1.4 De visie .NET	16
1.4.1 Platformonafhankelijk	16
1.4.2 Klassen in het .NET Framework.....	17
1.4.3 Programmeren in verschillende toepassingsgebieden	17
1.4.4 Hergebruik van bestaande componenten	17
1.4.5 Webservices	17
1.5 Overzicht van het .NET Framework	17
1.5.1 Versies van het .NET Framework	17
1.5.2 Visual Studio 2013	18
1.5.3 Situering van het .NET Framework	18
1.5.4 Programmeertalen en toepassingen voor .NET	19
1.5.5 .NET Framework Class Library.....	19
1.5.6 Enkele speciale categorieën van klassen.....	20
1.5.7 Common Language Runtime	20
1.6 Namespaces	22
1.7 Assembly	22
1.8 Toepassingen in .NET en Visual Studio 2013	23
1.9 Oefeningen	24
2 EIGEN KLASSEN AANMAKEN	25
2.1 Inleiding	25
2.2 Instellen van opties	25
2.2.1 Algemene opties	25
2.2.2 Option Explicit	26
2.2.3 Option Strict.....	26
2.2.4 Standaardbestandslocatie.....	26
2.3 Aanmaken van een nieuw project	27

2.4	De klasse Gebruiker	28
2.4.1	Probleemdefinitie	28
2.4.2	Probleemanalyse.....	28
2.4.3	Een klasse toevoegen aan een project	29
2.4.4	Programmacode.....	29
2.5	Value types en reference types	31
2.6	Instantievariabelen	33
2.7	Een constructor	33
2.7.1	Constructor zonder parameters.....	33
2.7.2	De constructor wordt opgeroepen bij het creëren van een object ..	34
2.7.3	Constructor met parameters.....	35
2.7.4	Standaardconstructor	36
2.8	Eigenschappen	36
2.8.1	Property.....	36
2.8.2	ReadOnly, WriteOnly	38
2.9	Methoden	38
2.9.1	De methode ToonGegevens	38
2.9.2	De methode Naam	39
2.10	Overloading	40
2.10.1	Overloading van methodes in een klasse van het framework	40
2.10.2	Methode overloading bij een eigen klasse.....	40
2.10.3	Constructor overloading	41
2.11	Een programma om de klasse uit te testen	41
2.11.1	Aanmaken van het formulier FrmTestKlasseGebruiker.....	41
2.11.2	Programmacode.....	42
2.11.3	Uitvoeren van het programma	44
2.11.4	Het project opslaan	46
2.12	Literals	46
2.13	Shared methods	46
2.14	Naamgeving en andere afspraken	47
2.15	Class View	47
2.16	Object browser	48
2.17	Debuggen	49
2.18	Oefeningen	49
3	ARRAYS EN ZOEKMETHODEN	53
3.1	Inleiding	53
3.2	Declaratie van een array	53
3.3	Een voorbeeld ter opfrissing	54
3.3.1	Probleemdefinitie H03VB01	54
3.3.2	Probleemanalyse.....	54
3.3.3	Schema	54
3.3.4	Bronprogramma en korte bespreking.....	55
3.3.5	Aanmaken van het project	55
3.3.6	Argumenten van event handlers	55
3.4	Een keuzelijst	56
3.5	Arrays en procedures	57
3.5.1	Probleemdefinitie H03VB02	57
3.5.2	Probleemanalyse en schema.....	57
3.5.3	Bronprogramma en korte bespreking.....	58
3.5.4	Aanmaken van het project	60
3.6	For Each/Next	60
3.7	De methode ToString	61

3.8	Inlezen van een array m.b.v. een invoervenster	61
3.8.1	Probleemdefinitie (H03VB03)	61
3.8.2	Bronprogramma	61
3.8.3	Aanmaken van het project	62
3.9	Zoeken in een (niet gesorteerde) array	63
3.9.1	Probleemdefinitie (H03VB04)	63
3.9.2	Analyse	63
3.9.3	Schema	63
3.9.4	Bronprogramma	64
3.9.5	Ontwerp van het formulier	66
3.10	Willekeurige getallen	66
3.11	Sorteren van een array	67
3.11.1	Probleemdefinitie (H03VB05)	67
3.11.2	Bronprogramma	67
3.11.3	Een lijst sorteren	68
3.11.4	Ontwerp van het formulier	68
3.12	Binaire zoekmethode in een gesorteerde array	68
3.12.1	Techniek en methode in het .NET Framework	68
3.12.2	Probleemdefinitie (H03VB06)	68
3.12.3	Probleemanalyse	69
3.12.4	Schema	70
3.12.5	Bronprogramma	70
3.12.6	Ontwerp van het formulier	72
3.13	Oefeningen	72
4	SORTEERALGORITMEN	75
4.1	Inleiding	75
4.2	Omwisselen van de waarden van twee variabelen	75
4.2.1	Probleemdefinitie (H04VB01)	75
4.2.2	Probleemanalyse	75
4.2.3	Schema	76
4.2.4	Bronprogramma	76
4.3	Sorteren van een array met de selectiemethode	77
4.3.1	Sorteren van een array	77
4.3.2	Probleemdefinitie (H04VB02)	77
4.3.3	Probleemanalyse	77
4.3.4	Schema	78
4.3.5	Bronprogramma	78
4.3.6	Ontwerp van het formulier	79
4.4	Sorteren met het algoritme Bubble Sort	79
4.4.1	Bubble Sort	79
4.4.2	Probleemdefinitie (H04VB03)	80
4.4.3	Bronprogramma	81
4.4.4	Ontwerp van het formulier	82
4.5	Recursie	82
4.5.1	Recursie	82
4.5.2	Probleemdefinitie (H04VB04)	83
4.5.3	Uitwerking	83
4.5.4	Ontwerp van het formulier	83
4.6	Quicksort	84
4.6.1	Probleemdefinitie (H04VB05)	84
4.6.2	Probleemanalyse	84
4.6.3	Ontwerp van het formulier	86
4.6.4	Bronprogramma	86
4.7	Oefeningen	88

5	GEGEVENSSTRUCTUREN IN .NET	91
5.1	Inleiding	91
5.2	Een enumerated type	91
5.2.1	Probleemdefinitie (H05VB01)	92
5.2.2	Probleemanalyse	92
5.2.3	Gebruikersinterface	92
5.2.4	Bronprogramma	93
5.3	Een tweede voorbeeld van een enumeratietype	94
5.3.1	Probleemdefinitie (H05VB02)	94
5.3.2	Probleemanalyse	94
5.3.3	Bronprogramma	95
5.3.4	Afdrukken stringwaarden van enumeratielijst	96
5.4	Een structuur	97
5.4.1	Wat is een structuur?	97
5.4.2	Aanmaken van een structuur (H05VB03)	98
5.4.3	Voorbeeld van een structuur	98
5.4.4	Constructor (H05VB04)	99
5.5	Collecties	100
5.5.1	Wat is een collectie?	100
5.5.2	Typed en untyped collections	100
5.6	List()	101
5.6.1	Probleemdefinitie (H05VB05)	101
5.6.2	Bronprogramma	101
5.6.3	Probleemdefinitie (H05VB06)	104
5.6.4	Bronprogramma	104
5.6.5	Eigenschappen en methoden van de klasse List()	105
5.7	SortedList()	106
5.7.1	Wat is een SortedList()	106
5.7.2	Probleemdefinitie (H05VB07)	106
5.7.3	Bronprogramma	107
5.7.4	Gebruikersinterface	108
5.7.5	Eigenschappen en methoden van de klasse SortedList()	108
5.8	Queue()	109
5.9	Stack()	109
5.10	ArrayList	110
5.10.1	Wat is een ArrayList?	110
5.10.2	Probleemdefinitie (H05VB08)	110
5.10.3	Programmacode	111
5.10.4	Boxing en unboxing	113
5.11	De collectie Controls	113
5.11.1	Probleemdefinitie (H05VB09)	113
5.11.2	Bronprogramma	114
5.11.3	Het type van het besturingselement	115
5.12	Oefeningen	115
6	MEER OVER FORMULIEREN	117
6.1	Inleiding	117
6.2	Delen van informatie tussen formulieren	117
6.2.1	Probleemdefinitie (H06VB01)	117
6.2.2	Ontwerpen van het formulier FrmGebruiker	118
6.2.3	Ontwerpen van het formulier FrmContactinformatie	118
6.2.4	De klasse Gebruiker	119
6.2.5	De structuur Contactinformatie	119
6.2.6	Uitwerking van het formulier FrmGebruiker	120
6.2.7	Uitwerking van het formulier FrmContactinformatie	121

6.3	Het besturingselement TabControl.....	121
6.3.1	Probleemdefinitie (H06VB02)	121
6.3.2	Ontwerp van het formulier.....	122
6.3.3	Programmacode.....	123
6.4	Een dialoogvenster	124
6.4.1	Dialoogvenster.....	124
6.4.2	Probleemdefinitie (H06VB03)	124
6.4.3	Ontwerp van het project en het bijkomend formulier	124
6.4.4	Programmacode.....	125
6.5	Gebeurtenissen bij formulieren.....	127
6.5.1	Load	127
6.5.2	Activate en Deactivate	127
6.5.3	FormClosing	127
6.5.4	Resize.....	128
6.6	Eigenschappen en methoden bij formulieren	129
6.7	SDI- en MDI-interfaces	129
6.7.1	Single document interface (SDI)	129
6.7.2	Multiple document interface (MDI).....	129
6.7.3	Probleemdefinitie (H06VB05)	130
6.7.4	Ontwerpen van het parent-formulier.....	131
6.7.5	Ontwerpen van het child-formulier	132
6.7.6	De klasse Gebruiker	132
6.7.7	Programmacode in het parent-formulier.....	132
6.7.8	De programmacode in het formulier FrmGebruiker	134
6.7.9	Eigenschappen van een MDI Child en van een MDI Parent	135
6.8	Oefeningen	136
7	OVERERVING	141
7.1	Inleiding	141
7.2	Definitie van een afgeleide klasse.....	142
7.3	Een voorbeeld van overerving.....	142
7.3.1	Probleemdefinitie (H07VB01)	142
7.3.2	De klasse Punt	142
7.3.3	Testprogramma	143
7.3.4	De klasse Cirkel	144
7.3.5	Testprogramma	145
7.4	Member access modifiers.....	146
7.5	Overridable, Overrides en MyBase	147
7.6	NotOverridable , NonInheritable.....	147
7.7	De klasse System.Object.....	148
7.8	Visuele overerving	148
7.8.1	Probleemdefinitie (H07VB02)	149
7.8.2	Uitwerking.....	149
7.9	Oefeningen	152
8	POLYMORFISME, ABSTRACTE KLASSEN, INTERFACES	155
8.1	Inleiding	155
8.2	Polymorfisme.....	155
8.2.1	Interface-polymorfisme.....	155
8.2.2	Methode overloading is een vorm van polymorfisme	155
8.2.3	Inheritance-polymorfisme	156
8.3	Polymorfisme, een voorbeeld.....	156
8.3.1	Probleemdefinitie (H08VB01)	156
8.3.2	Uitwerking van het voorbeeld	156
8.3.3	TypeOf, TryCast	160
8.4	Abstracte klassen.....	160

8.5	Voorbeeld m.b.t. abstracte klassen	161
8.5.1	Probleemdefinitie (H08VB02).....	161
8.5.2	Uitwerking.....	161
8.6	Interfaces	167
8.6.1	Een interface.....	167
8.6.2	Probleemdefinitie (H08VB03).....	168
8.6.3	Uitwerking.....	169
8.6.4	Vergelijking tussen een abstracte klasse en een interface.....	171
8.6.5	De methode ToString in een interface.....	171
8.7	Oefeningen	171
9	INTERACTIE MET ANDERE APPLICATIES	173
9.1	Inleiding	173
9.2	Automation	173
9.3	De object browser	173
9.4	Interactie met Excel	175
9.4.1	Probleemdefinitie (H09VB01).....	175
9.4.2	Uitwerking van het programma.....	175
9.4.3	De functie Pmt in Visual Basic.....	177
9.5	Gegevens lezen uit Excel	177
9.5.1	Probleemdefinitie (H09VB02).....	177
9.5.2	Bronprogramma.....	178
9.6	Interactie met Word	180
9.6.1	Probleemdefinitie (H09VB03).....	180
9.6.2	Het Word-sjabloon voorbereiden.....	180
9.6.3	Uitwerking van het project.....	181
9.7	Windows-toepassingen starten	182
9.7.1	Probleemdefinitie (H09VB04).....	182
9.7.2	Bronprogramma.....	183
9.8	Oefeningen	184
10	WERKEN MET MAPPEN EN BESTANDEN	185
10.1	Inleiding	185
10.2	De namespace System.IO	185
10.3	De klassen Directory, DirectoryInfo, File en FileInfo	186
10.4	De klasse DirectoryInfo	187
10.4.1	Methoden en eigenschappen van de klasse DirectoryInfo.....	187
10.4.2	Probleemdefinitie H10VB01.....	187
10.4.3	Uitwerking formulier.....	188
10.4.4	Programmacode bij <i>Huidige map</i>	188
10.4.5	Programmacode bij Eigenschappen van een map.....	189
10.4.6	Programmacode bij Creëren van een map.....	190
10.4.7	Programmacode bij Verwijderen van een map.....	190
10.5	De klasse FileInfo	190
10.5.1	Methoden en eigenschappen van de klasse FileInfo.....	190
10.5.2	Probleemdefinitie H10VB02.....	191
10.5.3	Uitwerking formulier.....	192
10.5.4	Programmacode bij Aanmaken nieuw bestand.....	192
10.5.5	Programmacode bij <i>Enkele eigenschappen</i>	192
10.6	De klasse DriveInfo	193
10.6.1	Probleemdefinitie H10VB03.....	193
10.6.2	Uitwerking programma.....	193
10.6.3	Programmacode.....	194
10.7	Streams	194

10.8	StreamReader en StreamWriter	195
10.8.1	Probleemdefinitie (H10VB04)	196
10.8.2	Ontwerp van het formulier.....	197
10.8.3	Programmacode.....	197
10.9	Lezen en schrijven van een sequentieel bestand	198
10.9.1	Probleemdefinitie (H10VB05)	198
10.9.2	Bronprogramma.....	199
10.10	Alternatieve methode.....	202
10.10.1	Probleemdefinitie (H10VB06)	202
10.10.2	Bronprogramma en korte bespreking	202
10.10.3	Bestand bestaat niet.....	203
10.10.4	Illustratie van de eigenschap Anchor.....	203
10.11	Oefeningen	204
11	DATABASES.....	209
11.1	Inleiding	209
11.2	Een database en databasesysteem	209
11.3	De voorbeelddatabase TijdArtGeg	210
11.3.1	De tabel tblTijdschriften	210
11.3.2	De tabel tblTijdschriftartikelen	210
11.3.3	De tabel tblTrefwoorden	211
11.3.4	De tabel tblTijdschriftartikelenTrefwoorden	212
11.4	SQL	212
11.4.1	Eenvoudige SELECT-instructie.....	212
11.4.2	SELECT met WHERE-component	213
11.4.3	SELECT met ORDER BY-component	213
11.4.4	INNER JOIN	214
11.4.5	Het INSERT-statement.....	215
11.4.6	Het UPDATE-statement	215
11.4.7	Het DELETE-statement.....	215
11.5	ADO.NET	215
11.5.1	ADO.NET is een interface voor gegevenstoegang	215
11.5.2	Data providers in het .NET Framework	216
11.5.3	Gegevenstoegang met een DataAdapter-object.....	216
11.5.4	Gegevenstoegang zonder DataAdapter-object.....	217
11.6	Een formulier met gegevenstoegang maken	218
11.6.1	Probleemdefinitie (H11VB01)	218
11.6.2	Een gegevensbron toevoegen in het venster <i>Data Sources</i>	218
11.6.3	Aanmaken van het formulier.....	224
11.6.4	Uittesten van het formulier.....	225
11.6.5	Probleem bij het testen van de verbinding.....	226
11.7	Componenten in het componentenvak	226
11.7.1	DataSet	226
11.7.2	TableAdapter	227
11.7.3	BindingSource.....	227
11.7.4	TableAdapterManager	227
11.7.5	BindingNavigator.....	227
11.7.6	Programmacode die gegenereerd wordt	228
11.8	Hoofd- en subformulier	228
11.8.1	Probleemdefinitie (H11VB02)	228
11.8.2	Een nieuw project aanmaken en de verbinding met de database maken	229
11.8.3	Toevoegen van de gegevens van het hoofdformulier.....	230
11.8.4	Toevoegen van de gegevens van het subformulier.....	231
11.8.5	Uittesten van het formulier.....	232

11.9 Eigenschappen van de DataGridView	232
11.9.1 DataSource	232
11.9.2 Columns	232
11.10 Een keuzelijst om een filter in te stellen	233
11.10.1 Probleemdefinitie	233
11.10.2 Toevoegen van het tekstvak.....	233
11.10.3 Toevoegen van de methoden om een record te selecteren.....	234
11.10.4 Toevoegen van de programmacode	240
11.11 Oefeningen	241
12 EEN PROGRAMMA VERDELEN.....	243
12.1 Inleiding	243
12.2 Installeren van InstallShield Limited Edition	243
12.2.1 Windows Installer.....	243
12.2.2 InstallShield installeren.....	243
12.3 Creëren van een InstallShield Limited Edition Project	248
12.3.1 Een InstallShield-project toevoegen.....	248
12.3.2 Application Information.....	251
12.3.3 Installation Requirements.....	251
12.3.4 Installation Architecture	252
12.3.5 Application Files	252
12.3.6 Application Shortcuts.....	253
12.3.7 Application Registry	255
12.3.8 Installation Interview.....	255
12.3.9 De oplossing bouwen (Build)	256
12.4 Installeren van het programma	257
12.4.1 Installeren van het programma.....	257
12.4.2 Problemen?	259
12.4.3 Starten van het programma	259
12.5 De-installeren van een programma.....	259
12.6 ClickOnce	260
12.6.1 ClickOnce	260
12.6.2 Publiceren van een toepassing	260
12.6.3 Installeren van een toepassing	263
12.6.4 Updates van de toepassing.....	264
12.7 Oefeningen	266
TREFWOORDENREGISTER	267