

Leren programmeren in C#

Michiel Rotteveel

Leren programmeren in C#

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: M. Rotteveel
Redactie: J.G. Neurink / M.H. van Haren
Opmaak: Henk Pel
Titel: Leren programmeren in C#
Foto omslag: M. Rotteveel
ISBN 978 90 3725 936 0
Eerste druk / eerste oplage
© Boom beroepsonderwijs 2021

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoudsopgave

Deel 1: Introductie

Leeswijzer 2

Inleiding Deel 1: Introductie 3

Programmeren in C# 3

Hoofdstuk 1 De programmeeromgeving (IDE) 5

Wat is een IDE? 5

Opstarten 5

Naam geven 8

Spelregels naamgeving 9

De IDE 10

Je eerste programma 11

De Code Editor 12

Opdracht IDE 1: Disclaimer 14

Wat weet je nu? 14

Hoofdstuk 2 Rekenen 15

Rekenen 15

Opdracht Rekenen 1: Optellen 15

Opdracht Rekenen 2: Meerdere getallen optellen 15

Opdracht Rekenen 3: Aftrekken 16

Opdracht Rekenen 4: Gebroken getallen aftrekken 16

Opdracht Rekenen 5: Meerdere getallen aftrekken 17

Opdracht Rekenen 6: Meer getallen vermenigvuldigen 17

Opdracht Rekenen 7: Negatieve getallen 18

Opdracht Rekenen 8: Delen 18

Opdracht Rekenen 9: Modulo 18

Bibliotheek 19

Haakjes en volgorde 19

Opdracht Rekenen 10: Machtsverheffen 19

Wat moet je nu kunnen en kennen? 19

Hoofdstuk 3 Variabelen 21

Wat is een variabele? 21

De tafel van X 21

Hoe 'denkt' een computer? 21

Werking van een variabele 21

Opdracht Variabelen 1: Vraag? 25

| | |
|---|----|
| Gebruikersinput | 25 |
| <i>Opdracht Variabelen 2: Verklaring</i> | 26 |
| Typen variabelen | 26 |
| Teksten | 27 |
| <i>Opdracht Variabelen 3: Variabele verklaring</i> | 29 |
| Getallen | 29 |
| Omzetten van variabelen | 32 |
| Logische fout | 42 |
| Syntactische fout | 42 |
| <i>Opdracht Variabelen 4: Fahrenheit naar Celsius</i> | 42 |
| Logische variabele | 42 |
| DatumTijd-variabele | 43 |
| Bereik van variabelen | 45 |
| Opdrachten Variabelen | 46 |
| <i>Opdracht Variabelen 5: Bereken oppervlak</i> | 46 |
| <i>Opdracht Variabelen 6: Bereken omtrek</i> | 46 |
| <i>Opdracht Variabelen 7: Hoelang nog?</i> | 47 |
| <i>Opdracht Variabelen 8: Euro naar ponden 1</i> | 47 |
| <i>Opdracht Variabelen 9: Euro naar ponden 2</i> | 47 |
| <i>Opdracht Variabelen 10: Verbruik</i> | 47 |
| <i>Opdracht Variabelen 11: Vermogen elektrische apparaten berekenen *</i> | 47 |
| <i>Opdracht Variabelen 12: Verbruik elektrische apparaten berekenen *</i> | 47 |
| <i>Opdracht Variabelen 13: Golfsnelheid *</i> | 47 |
| <i>Opdracht Variabelen 14: Het gulden getal uitrekenen *</i> | 48 |
| Wat moet je nu kunnen en kennen? | 48 |

Hoofdstuk 4 Selectie 50

| | |
|--|----|
| Wat is selectie? | 50 |
| Hoe maak je een selectie: if | 50 |
| <i>Opdracht Selectie 1: Leeftijdchecker 1</i> | 52 |
| Hoe maak je een selectie: if ... else | 52 |
| <i>Opdracht Selectie 2: Leeftijdchecker 2</i> | 53 |
| <i>Opdracht Selectie 3: Vriezen of dooien</i> | 53 |
| Nesten | 53 |
| <i>Opdracht Selectie 4: Nesten</i> | 57 |
| <i>Opdracht Selectie 5: AddYears</i> | 58 |
| <i>Opdracht Selectie 6: Aanvragen vergunning</i> | 58 |
| <i>Opdracht selectie 7: Reisbeleid</i> | 60 |
| <i>Opdracht Selectie 8: Motorrijbewijs *</i> | 60 |
| <i>Opdracht Selectie 9: Soort motor *</i> | 61 |
| If ... else | 61 |
| <i>Opdracht Selectie 10: Maandnamen met if... else</i> | 62 |
| Switch | 63 |
| <i>Opdracht Selectie 11: Maandnamen met switch</i> | 64 |
| Wat moet je nu kunnen en kennen? | 64 |

Hoofdstuk 5 Herhaling 65

- Wat is herhaling? 65
- Het gebruik van een loop (herhaling) 68
 - Opdracht Herhaling 1: Rij getallen zonder loop* 70
 - Opdracht Herhaling 2: Rij getallen met loop* 70
- Herhaling met behulp van een variabele 70
 - Opdracht Herhaling 3: Rij getallen met variabele loop* 71
 - Opdracht Herhaling 4: De tafel van ...* 72
 - Opdracht Herhaling 5: Meerdere tafels* 72
 - Opdracht Herhaling 6: Machten 1* 72
 - Opdracht Herhaling 7: Machten 2 ** 73
 - Opdracht Herhaling 8: Faculteit uitrekenen ** 73
 - Opdracht Herhaling 9: De rij van Fibonacci ** 73
- Meer loops 74
 - do ... while 74
 - Opdracht Herhaling 10: Do...While loop* 74
 - for 74
 - Opdracht Herhaling 11: For ...* 75
 - foreach 75
 - Wat moet je nu kunnen en kennen? 75

Hoofdstuk 6 Arrays 76

- Wat is een Array? 76
- Hoe maak je een Array? 76
 - Opdracht Array 1: Hele getallen* 77
 - Opdracht Array 2: Decimale getallen* 77
 - Opdracht Array 3: Teksten* 77
- Toegang tot een element van de Array 78
 - Opdracht Array 4: Hele getallen* 78
 - Opdracht Array 5: Teksten* 78
 - Opdracht Array 6: Teksten en nummers* 78
- Bewerkingen met Arrays 79
 - Opdracht Array 7: Een element van een Array bewerken* 79
 - Opdracht Array 8: Letters tellen** 79
 - Opdracht Array 9: De elementen van twee Arrays combineren* 79
- Herhaling en Arrays 79
- Een Array sorteren 80
 - Opdracht Array 10: Getallen sorteren* 81
 - Opdracht Array 11: Teksten sorteren* 81
 - Opdracht Array 12: Bewerken* 81
- Problemen met Arrays 81
 - Opdracht Array 13: Resize* 81
 - Opdracht Array 14: Te groot* 81
- Wat moet je nu kunnen en kennen? 82

Hoofdstuk 7 Forms & Controls 83

- Een Windows-programma 83
 - Properties (Eigenschappen) 86*
- De Code Editor 87
- Controls 89
- Opdrachten Controls 91
 - Opdracht Controls 1: Voorbeeld controls 91*
 - Opdracht Controls 2: Temperatuur 91*
 - Opdracht Controls 3: Aanvraag vergunning 91*
 - Opdracht Controls 4: Rekenmachine 92*
 - Opdracht Controls 5: Datum – tijd * 92*
- Wat moet je nu kunnen en kennen? 93

Hoofdstuk 8 Foutafhandeling 94

- Wat is foutafhandeling? 94
- Foutafhandeling met de MessageBox 94
 - Opdracht Foutmelding 1: Messagebox 98*
- Foutafhandeling met de ErrorProvider 98
 - Opdracht Foutmelding 2: ErrorProvider 100*
- Wat moet je nu kunnen en kennen? 100

Hoofdstuk 9 Methoden 101

- Wat is een methode? 101
- Hoe maak je een methode? 102
 - Opdracht Methode 1: Welkomstboodschap 103*
 - Vorbereidende opdracht Methode 2 en 3 104*
 - Opdracht Methode 2: Oppervlakte 104*
 - Opdracht Methode 3: Omtrek 104*
 - Opdracht Methode 4: Euro naar ponden 104*
 - Opdracht Methode 5: Het gulden getal uitrekenen * 105*
- Wat moet je nu kunnen en kennen? 105

Hoofdstuk 10 Events 106

- Wat is een event? 106
- Het gebruik van events 107
- Toepassing 111
 - Opdracht Event 1: Geheimschrijver 111*
 - Opdracht Event 2: Kleurenwisselaar * 111*
- Wat moet je nu kunnen en kennen? 112

Hoofdstuk 11 Teken en 113

- Waarom tekenen? 113
- Eenvoudig tekenen 113
 - Opdracht Teken en 1: Verticale tralies 114*
 - Opdracht Teken en 2: Horizontale tralies 114*
 - Opdracht Teken en 3: Raster 114*

| | |
|--|------------|
| Interactief tekenen met invoervelden | 115 |
| <i>Opdracht Tekenen 4: Variabel raster</i> | 115 |
| <i>Opdracht Tekenen 5: Scrollbar toevoegen</i> | 116 |
| Interactief tekenen met events | 117 |
| <i>Opdracht Tekenen 6: Een halve lijn</i> | 117 |
| <i>Opdracht Tekenen 7: De hele lijn</i> | 117 |
| Andere objecten tekenen | 118 |
| <i>Opdracht Tekenen 8: Rechthoek</i> | 118 |
| <i>Opdracht Tekenen 9: Rechthoek met diagonaal *</i> | 118 |
| <i>Opdracht Tekenen 10: Ellips *</i> | 119 |
| Tekenen door meerdere punten | 119 |
| <i>Opdracht Tekenen 11: Lijnen</i> | 120 |
| <i>Opdracht Tekenen 12: Golven *</i> | 121 |
| Wat moet je nu kunnen en kennen? | 121 |
| Hoofdstuk 12 Klasse (Class) | 122 |
| Wat is een class? | 122 |
| Interface | 122 |
| Encapsulation | 122 |
| Herbruikbaar | 123 |
| Gebruik van een klasse | 123 |
| Het maken van de klasse (beschrijving) | 123 |
| Vormgeven van de klasse | 124 |
| Public versus Private | 124 |
| Eigenschappen (properties) | 125 |
| <i>Opdracht Klasse 1: Rechthoek aanmaken</i> | 126 |
| Gebruik van een klasse | 126 |
| Constructor | 128 |
| <i>Opdracht Klasse 2: Rechthoek berekeningen</i> | 129 |
| <i>Opdracht Klasse 3: Rechthoek constructor</i> | 129 |
| Het invoerveld | 129 |
| Overloaden | 130 |
| <i>Opdracht Klasse 4: Rechthoek overloaden</i> | 131 |
| Het tekenen | 131 |
| <i>Opdracht Klasse 5: Rechthoek tekenen</i> | 132 |
| Uitbreiden van de klasse | 132 |
| <i>Opdracht Klasse 6: Rechthoek uitbreiden</i> | 134 |
| Wat moet je nu kunnen en kennen? | 135 |
| Nawoord Deel 1 | 136 |

Deel 2: Gegevens

Inleiding Deel 2: Gegevens 139

Hoofdstuk 1 Interne gegevens 141

Array 141

Opdracht Array 1: Telefoonmerkenlijst 157

Value type versus Reference type 158

Opdracht Array 2: Uitgebreide telefoonmerkenlijst 177

List<T> 178

Opdracht List<T> 1: Telefoonmerkenlijst 181

List versus Array 181

Enumerable (opsommingen) 181

Opdracht Enum 1: Telefoonmerkenlijst 187

Eenvoudige objecten (Structs) 207

Opdracht Struct 1: 3D-punt 215

Opdracht Struct 2: Telefoonnummer 215

Opdracht Struct 3: Geografische coördinaten 216

Opdracht Struct 4: Geografische coördinaten aanvulling 216

Hoofdstuk 2 Bestanden (externe gegevens) 218

Tekstbestanden 218

Opdracht Tekstbestanden 1: StringArray wegschrijven 222

Opdracht Tekstbestanden 2: String Array meerdere selecties 224

*Opdracht Tekstbestanden 3: Bewaren gebruikersinstellingen * 225*

.csv-bestanden (Comma Separated Values) 226

Opdracht .csv-bestanden 1: Flexibele opslag 232

Opdracht .csv-bestanden 2: Klassenlijst 233

Opdracht .csv-bestanden 3: Klassenlijst inlezen 234

*Opdracht .csv-bestanden 4: Klassenlijst inlezen Extra * 234*

Afbeeldingen 234

Opdracht Afbeeldingen 1: Liquid layout-form 240

Opdracht Afbeeldingen 2: Instellingen 240

Opdracht Afbeeldingen 3: Crashbestendig maken 240

Opdracht Afbeeldingen 4: Draaien en flippen 241

Opdracht Afbeeldingen 5: Kopiëren 241

Opdracht Afbeeldingen 6: Afbeeldingen en tekenen 241

*Opdracht Afbeeldingen 7: Thumbnails * 242*

Video 242

Opdracht Video 1: Extra knoppen en bestandsinformatie 249

Opdracht Video 2: Extra instellingen toevoegen 250

*Opdracht Video 3: Meerdere bestanden selecteren en afspelen * 250*

Hoofdstuk 3 Systeembestanden (externe gegevens) 251

Windows registry 251

XML-bestanden 255

Opdracht XML 1: ListBox vullen vanuit XML 263*Opdracht XML 2: Extra selectieveld* 263*Opdracht XML 3: DataGridView* 263*Opdracht XML 4: Weerstation ** 264*Opdracht XML 5: Landenlijst* 267**Nawoord Deel 2 268*****Deel 3: Databases*****Inleiding Deel 3: Databases 271**

C# en databases 271

Benodigde voorkennis 271

Hoe werk je met dit boek? 272

Hoofdstuk 1 Introductie databases 273

Wat is een database? 273

Relationele databases 273

De databases in dit boek 275

Hoofdstuk 2 Hoe werkt het? 276

DataSets 276

Opdracht DataSet 1: Garage / tabel Auto's 280*Opdracht DataSet 2: Landen, tabellen en werelddelen* 280

Binding data aan controls 281

Opdracht Binding 1: Koppelen van controls 285*Opdracht Binding 2: Garages* 285

1-op-n master-detail-binding 286

Opdracht 1-op-n-koppeling 1: Facturen 295*Opdracht 1-op-n-koppeling 2: Wielerploeg* 295*Opdracht 1-op-n-koppeling 3: Provincies* 296

Look-up-tabellen in een DataGridView 296

Opdracht look-up DataTable 1: Facturen 304*Opdracht look-up DataTable 2: Auto's* 304

Value & display members 304

Opdracht Value- & DisplayMembers 1: Auto's 307*Opdracht Value- & DisplayMembers 2: Eigenaren* 307

N-op-m (n:m) koppeling tussen DataTables 307

Opdracht n:m-koppeling 1: Auto's en eigenaren 321*Opdracht n:m-koppeling 2: Aanvullen voorbeeld 6 n:m-koppeling* 322*Opdracht n:m-koppeling 3: Klassen en leerlingen ** 322

Wat heb je geleerd? 322

Hoofdstuk 3 Database versus Dataset 324

Inleiding 324

Desktop-database: Microsoft Access 324

*Opdracht Access 1: Rugbyteams 340**Opdracht Access 2: Auto's en eigenaren 340**Opdracht Access 3: Klassen en leerlingen * 340**Opdracht Access 4: Politici en partijen 341***Hoofdstuk 4 Drag & Drop 342**

Inleiding 342

*Opdracht Drag & Drop 1: Vakantiewoningen 347**Opdracht Drag & Drop 2: Kastelen en pachters * 347*

Wat heb je geleerd? 348

Hoofdstuk 5 Multiformulierapplicatie 349

Eén applicatie, meerdere formulieren 349

*Opdracht Multiformulier 1: Rugbyteams 356**Opdracht Multiformulier 2: Auto's en eigenaren 356*

Meerdere formulieren bij start applicatie 356

Opdracht Multiformulier 3: Login 366

Wat heb je geleerd? 366

Hoofdstuk 6 Online databases 367

MySQL 367

*Opdracht MySQL 1: Adreslijst 375**Opdracht MySQL 2: Meerdere adressen 375**Opdracht MySQL 3: Voetbaluitslagen 375**Opdracht MySQL 4: Classic Models compleet * 376*

Microsoft SQL Server Express 376

*Opdracht SQL Server Express 1 * 376*

Wat heb je geleerd? 376

Nawoord Deel 3 378***Deel 4: Objectoriëntatie*****Inleiding Deel 4: Objectoriëntatie 381**

C# OO & UML 381

Benodigde voorkennis 382

Opmerkingen bij deel 4 382

Hoofdstuk 1 Voorbereiding 383

Objectoriëntatie 383

Kenmerken 383

UML 383

| | |
|---|------------|
| Vorbereiding voorbeeldprogramma | 384 |
| Wat heb je geleerd? | 390 |
| Hoofdstuk 2 Klassen (Classes) | 391 |
| Beschrijving | 391 |
| Inhoud (instantie) | 391 |
| UML | 391 |
| Vuistregels | 392 |
| Soorten klassen | 392 |
| Naamgeving | 393 |
| Opdrachten Klassen | 394 |
| <i>Opdracht 2.1: Aanmaken klasse Gras</i> | 394 |
| <i>Opdracht 2.2: Aanmaken klasse Venijnboom</i> | 394 |
| <i>Opdracht 2.3: Aanmaken klasse Vingerhoedskruid</i> | 394 |
| <i>Opdracht 2.4: Aanmaken klasse Koe</i> | 394 |
| <i>Opdracht 2.5: Aanmaken klasse Lynx</i> | 394 |
| <i>Opdracht 2.6: Aanmaken klasse Beer</i> | 394 |
| Wat heb je geleerd? | 394 |
| Hoofdstuk 3 Encapsulation, variabelen en eigenschappen | 395 |
| Encapsulation | 395 |
| Naamgeving variabelen én eigenschappen | 396 |
| Variabelen | 396 |
| UML | 397 |
| Visual Studio klassendiagram | 399 |
| Eigenschappen (properties) | 400 |
| Opdrachten Encapsulation, variabelen en eigenschappen | 404 |
| <i>Opdracht 3.1: Vorbereiding klasse Gras</i> | 404 |
| <i>Opdracht 3.2: De klasse Gras implementeren</i> | 405 |
| Verkorte schrijfwijze eigenschappen | 406 |
| Opdracht Verkorte schrijfwijze | 409 |
| <i>Opdracht 3.2: Verkorte schrijfwijze: de klasse Venijnboom</i> | 409 |
| Wat heb je geleerd? | 410 |
| Hoofdstuk 4 Methoden | 411 |
| Beschrijving | 411 |
| Eisen | 411 |
| Overloaden | 411 |
| UML | 412 |
| Opdrachten Methoden | 418 |
| <i>Opdracht 4.1: Methoden toevoegen aan de klasse Koe</i> | 418 |
| <i>Opdracht 4.2: Methoden toevoegen aan de klassen Gras en Venijnboom</i> | 418 |
| Reflectie | 418 |
| Wat heb je geleerd? | 418 |

Hoofdstuk 5 Constructor 419

Beschrijving 419

Eisen constructor 419

UML 419

Opdrachten Constructor 421

*Opdracht 5.1: Constructors voor Koe 421**Opdracht 5.2: Constructors voor Gras en Venijnboom 421*

Opdrachten Tekenen 423

*Opdracht 5.3: Tekenen van Koe 423**Opdracht 5.4: Tekenen van Gras en Venijnboom 423*

Wat heb je geleerd? 424

Hoofdstuk 6 Structs 425

Beschrijving 425

Stack 425

Heap 425

Eigenschappen struct 426

UML 426

Wat heb je geleerd? 429

Hoofdstuk 7 Compositie 430

Beschrijving 430

Wat heb je geleerd? 434

Hoofdstuk 8 Overerving 435

Beschrijving 435

Voordelen 435

UML 435

NatSimII 437

Erven 438

Afspraken 439

Opdracht Overerving 452

Opdracht 8.1: Klasse Vingerhoedskruid 452

Wat heb je geleerd? 452

Hoofdstuk 9 Abstracte klasse 453

Beschrijving 453

UML 453

Opdrachten Abstracte klassen 454

*Opdracht 9.1: Abstract maken van de klasse Leven 454**Opdracht 9.2: Abstract maken van de klasse Plant 454*

Wat heb je geleerd? 454

Hoofdstuk 10 Oefening, uitbouwen van het project 455

Opdracht Typering object 457

Opdracht 10.1: IsDier() 457

| | |
|--|------------|
| Opdrachten Uitbouwen van het project | 459 |
| <i>Opdracht 10.2: Vleeseter</i> | 459 |
| <i>Opdracht 10.3: Methode Eet() bij Vleeseter</i> | 459 |
| <i>Opdracht 10.4: Alleseter</i> | 459 |
| <i>Opdracht 10.5: Methode Eet() bij Alleseter</i> | 459 |
| Opdracht Vernieuwde klasse | 461 |
| <i>Opdracht 10.6: Nieuwe klasse Koe</i> | 461 |
| Hoofdstuk 11 Interfaces | 463 |
| Beschrijving | 463 |
| UML | 464 |
| Opdracht Interfaces | 468 |
| <i>Opdracht 11.1: Klok toevoegen</i> | 468 |
| UML-klassendiagram Dier | 468 |
| Wat heb je geleerd? | 469 |
| Hoofdstuk 12 Delegates | 470 |
| Voorafje | 470 |
| Beschrijving | 470 |
| UML | 470 |
| Aanmaken delegate | 471 |
| Waarom delegates? | 475 |
| Callback-functies | 475 |
| Opdrachten Delegate | 479 |
| <i>Opdracht 12.1: De ProgressBar</i> | 479 |
| <i>Opdracht 12.2: Spreektijd</i> | 479 |
| <i>Opdracht 12.3: Spreektijd met eindtijd waarschuwing *</i> | 479 |
| Multithreading | 479 |
| Wat heb je geleerd? | 480 |
| Hoofdstuk 13 Events | 481 |
| Beschrijving | 481 |
| UML | 482 |
| Opdrachten Completeren formulier | 491 |
| <i>Opdracht 13.1: Toevoegen code</i> | 491 |
| <i>Opdracht 13.2: Test het programma</i> | 491 |
| NatSimII tot nu toe | 503 |
| Opdrachten Events | 504 |
| <i>Opdracht 13.3: Extra informatie tonen</i> | 504 |
| <i>Opdracht 13.4: Nog meer informatie tonen *</i> | 504 |
| Collision detection | 504 |
| Opdrachten Collision detection | 509 |
| <i>Opdracht 13.5: Red de Venijnboom</i> | 509 |
| <i>Opdracht 13.6: Vleeseter, Jaguar</i> | 509 |
| <i>Opdracht 13.7: Vleeseter, Lynx</i> | 509 |
| <i>Opdracht 13.8: Alleseter, Beer *</i> | 510 |

Opdracht 13.9: Red de Venijnboom opnieuw 511

Opdracht 13.10: Conversie van Leven naar Venijnboom 511

Opdracht 13.11: Conversie van Leven naar Vingerhoedskruid 511

Wat heb je geleerd? 512

Hoofdstuk 14 Oefening baart kunst 513

Opdrachten NatSimII 513

Opdracht 14.1: Venijnboom 513

Opdracht 14.2: Vingerhoedskruid 513

Opdracht 14.3: Zaaïen 513

Opdracht 14.4: Over gras lopen * 513

Opdracht 14.5: Voortplanting 513

Opdracht 14.6: Voortplanten Planten * 514

Opdracht 14.7: Shooting game * 514

Opdracht 14.8: Snelheid wijzigen * 514

Opdracht 14.9: Plaatjes * 514

Opdracht 14.10: Geluid * 514

Opdracht 14.11: Collision detection * 514

Opdracht 14.12: Evenwicht * 515

Nawoord Deel 4 516

Bijlagen

ASCII-tabel deel 1: Besturingscodes 519

ASCII-tabel deel 2: Tekens 520

Begrippen 521

Bronvermelding 523

Cookbook Events 524

Kleurcodering Visual Studio 2019 C# editor 527

Naamgeving klasse 528

Naming conventions C# 529

UML-diagram NatSimII 534

Deel 1: Inleiding

Leeswijzer

| | |
|------------------------------|---|
| <code>do...while</code> | speciale combinaties van commando's, ook wel constructies genoemd |
| <code>Commentaar</code> | commentaar in de code |
| <code>double</code> | gereserveerde woorden C# |
| <code>KleinGetal</code> | de naam / het type van een object |
| <code>Naam</code> | de zelfbedachte naam van een control of project |
| <code>Standaardcode</code> | alle normale code |
| <code>Overbodige code</code> | code die niet uitgevoerd wordt |
| <code>Show()</code> | de naam van een methode |
| <code>"Tekst"</code> | de waarde van een <code>string</code> -variabele |
| <code>Variabele</code> | lokale variabelen en globale variabelen |

Inleiding Deel 1: Introductie

Programmeren in C#

Voor je ligt een boek waarmee je de beginselen leert van het programmeren met behulp van de programmeertaal C# (C sharp). Behalve bij de taal C# wordt ook stilgestaan bij algemene programmeerprincipes. Je hebt dus geen voorkennis nodig om met dit boek te beginnen.

Programmeren is niets anders dan een stel opdrachten aan een computer geven. Gelukkig kun je deze opdrachten in een tekstfile opslaan, zodat je hetzelfde programma meerdere keren kunt laten uitvoeren. Een computer verstaat geen menselijke taal, maar begrijpt alleen enen en nullen. Daarom moet je tekstbestand vertaald worden naar een machinetaal van enen en nullen. In computertermen heet dit compileren. Dat doet het programma waarmee je programmeert (de programmeeromgeving of IDE, zie hoofdstuk 1) voor je. Wij gebruiken daarvoor Visual Studio Community 2019. Dat kun je gratis downloaden van www.visualstudio.com. Dit boek is gemaakt met Windows 10.

Een computerprogramma kent een paar basisprincipes:

- Alle programmeerregels worden een voor een na elkaar uitgevoerd.
- Een programma maakt meestal gebruik van variabelen.
- Je kunt sommige regels laten herhalen in een lus.
- Je kunt een computer laten kiezen.
- Een programmeertaal gebruikt woorden die gereserveerd zijn om opdrachten te geven aan de computer (reserved words). Die kun je niet voor iets anders gebruiken.

Dit boek bevat een eerste kennismaking met het programmeren. Je hebt dus geen voorkennis nodig van wat voor programmeertaal dan ook. Keerzijde is wel dat sommige zaken eenvoudiger worden voorgesteld dan ze in werkelijkheid zijn om je de achterliggende principes te leren. Maar voor een eerste kennismaking met programmeren zul je aan dit boek meer dan voldoende hebben.

Dit boek bestaat niet alleen uit leeswerk, maar ook uit een reeks voorbeelden en opdrachten. Lees de voorbeelden dus niet alleen, maar voer ze regel voor regel in, in een eigen programma. Probeer iedere regel te begrijpen. Maak daarna de opdrachten. Sla iedere opdracht apart op in een bestandje. Soms heb je het voor een latere opdracht nog nodig. De opdrachten met een sterretje (*) zijn bedoeld als extra opdrachten (moeilijker of als verdieping van de lesstof). Ze zijn niet essentieel voor je leerproces, maar wel leerzaam.

Waarom zoveel eenvoudige opdrachtjes? Waarom zelf typen en niet knippen en plakken? Je begrijpt dan beter wat je doet en leert zo al doende. Bovendien... goed programmeren leer je alleen door het heel veel te doen!

Veel succes & plezier.

Hoofdstuk 1

De programmeeromgeving (IDE)

Wat is een IDE?

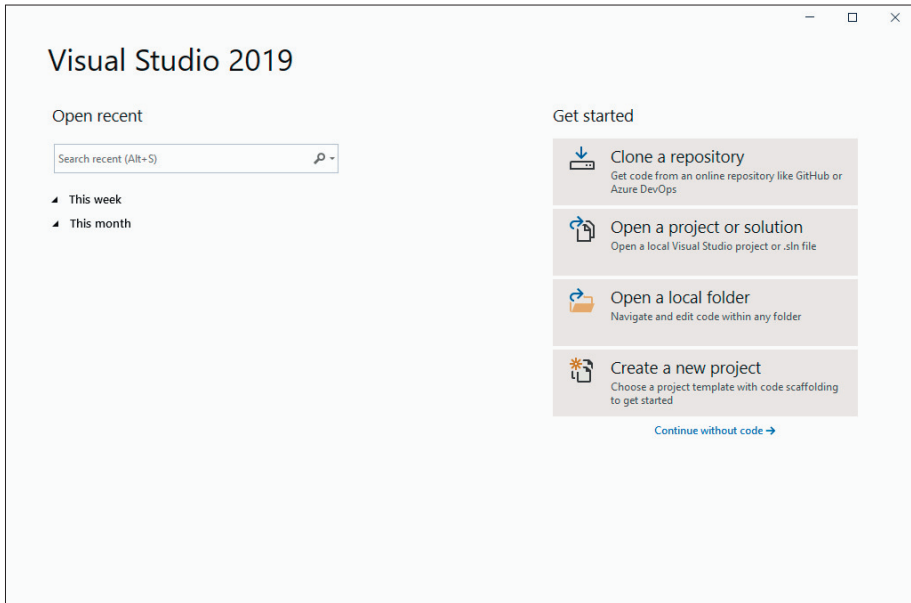
IDE staat voor Integrated Development Environment, of geïntegreerde programmeeromgeving. Dit is een programma om programma's te ontwerpen, te schrijven en te testen. Dit laatste noemen we debuggen. Een programmeertaal is niets anders dan een taal waarin je opdrachten opschrijft die een computer kan uitvoeren.

En net zoals Word een grammatica- en spellingcontrole heeft, heeft een IDE dat ook voor de programmeertalen waar die voor gemaakt is.

Omdat een programmeertaal heel erg precies is, kan de IDE je ook helpen en suggesties geven over wat je allemaal kunt maken. Dat noemen we IntelliSense. De IDE van C# die wij gebruiken heet Microsoft Visual Studio.NET. Visual Studio heet 'visual' omdat je direct kunt zien wat je maakt. Binnen Visual Studio wordt er heel veel gebruikgemaakt van kleur om je te laten zien wat er gebeurt. Met Visual Studio.NET – of zoals de meeste mensen zeggen: .NET (spreek uit als dot net) – kun je, behalve in C#, ook in Visual Basic.NET, C++, F#, Javascript, HTML, Python en nog veel meer programmeertalen programmeren.

Opstarten

Als je .NET 2019 opstart kom je in het volgende scherm terecht. Andere versies kunnen op details afwijken, maar het idee is hetzelfde.

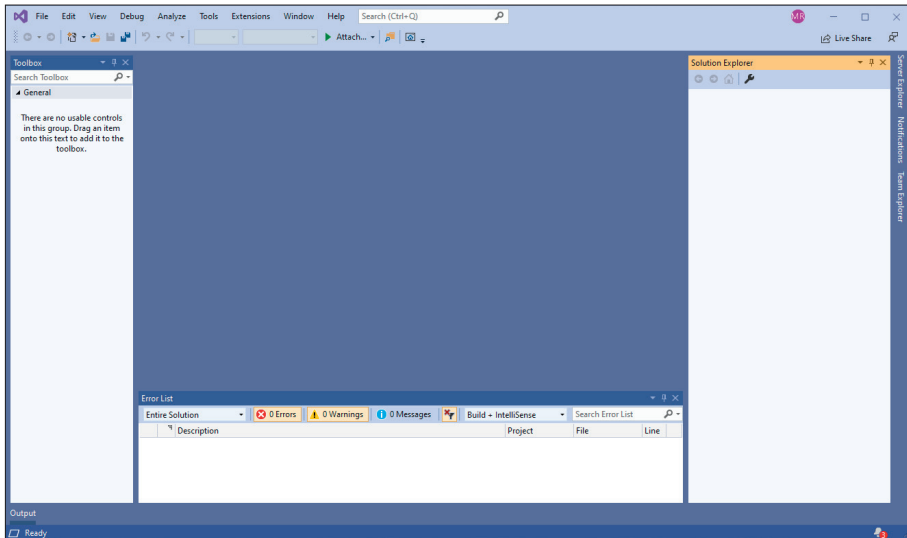


Schermafbeelding 1: Start scherm

Eigenlijk spreekt het startscherm voor zich. Aan de linkerkant zie je de projecten die je recent geopend hebt, rechts heb je opties om projecten op te zoeken, een map te openen en een nieuw project aan te maken.

Omdat we eerst Visual Studio willen leren kennen starten we het programma op zonder project. Dit doen we door op de blauwe tekst te klikken [Continue without code->](#)

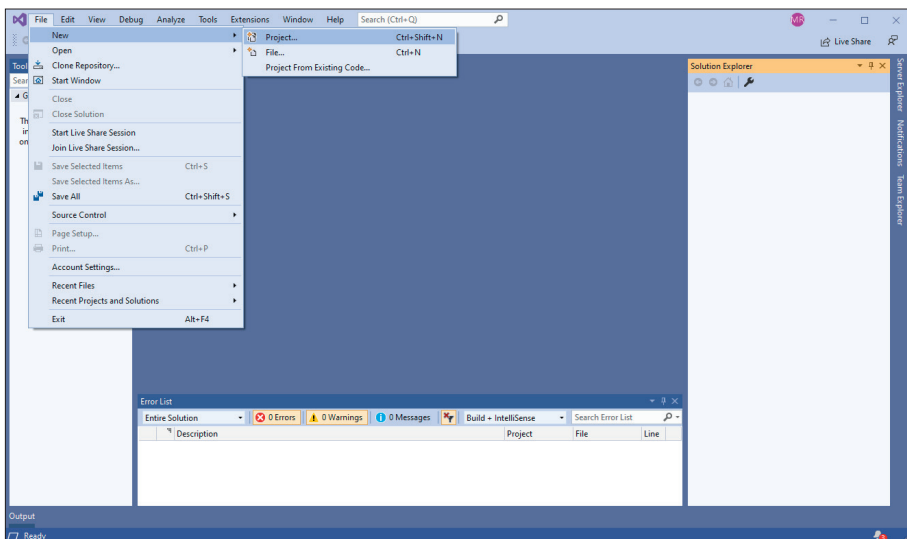
Je komt nu in het volgende scherm:



Schermafbeelding 2: Onderdelen startscherm

Je ziet in Schermafbeelding 2 twee belangrijke gebieden:

- 1 Het Menu en de Toolbar. Deze zijn vergelijkbaar met Word en andere Office-pakketten. Via de Toolbar kun je allerlei instellingen veranderen en functies aanroepen.
- 2 De werkomgeving bestaat uit: een linkerpanel, een middengebied en een rechterpanel.

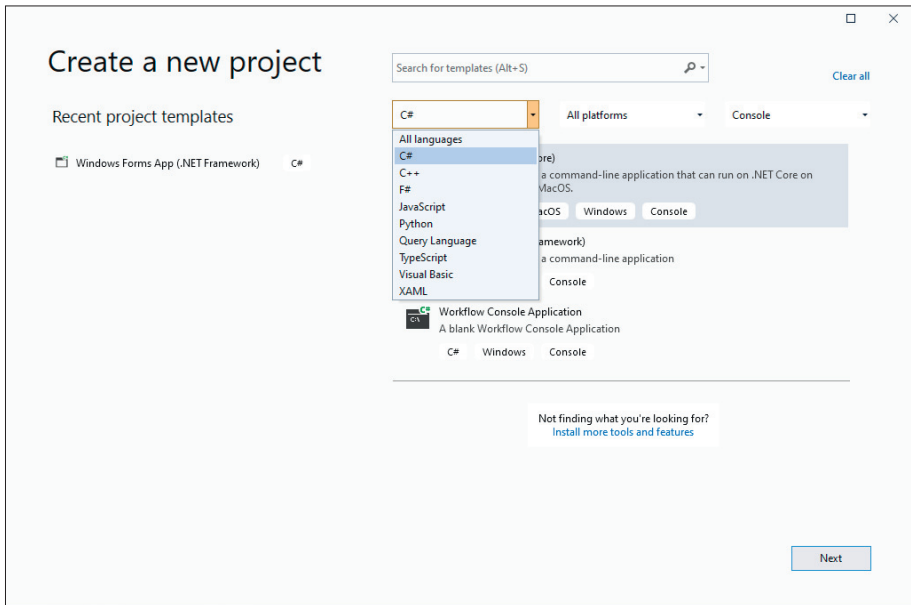


Schermafbeelding 3: Keuze New Project

Naam geven

We gaan nu een nieuw programma maken. Net zoals bij Word klik je dan op *File*, *New*. Anders dan bij Word kies je nu voor *Project*.

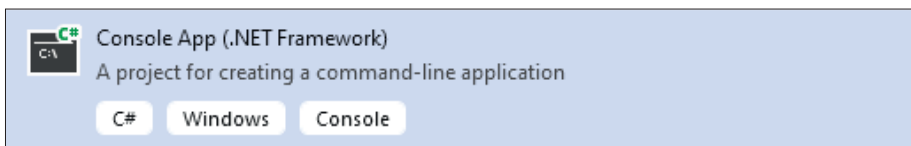
Nu kom je in het scherm *Create a new project*. In de eerste dropdown-list, die in de afbeelding is opengeklapt, kun je een selectie maken voor een van de vele programmeertalen die je kunt gebruiken binnen Visual Studio. Wij kiezen voor C#. Kies je voor een andere taal dan kan het zijn dat de omgeving en de code heel veel op C# lijken. Alleen zal het niet werken omdat de syntax, de regels van de taal, afwijkend zijn. Voor een Amerikaan lijken Duits en Nederlands erg veel op elkaar. Alleen als ik Nederlands tegen een Duitser spreekt verstaat hij me vaak niet. Zo verschillend zijn computertalen minstens ook.



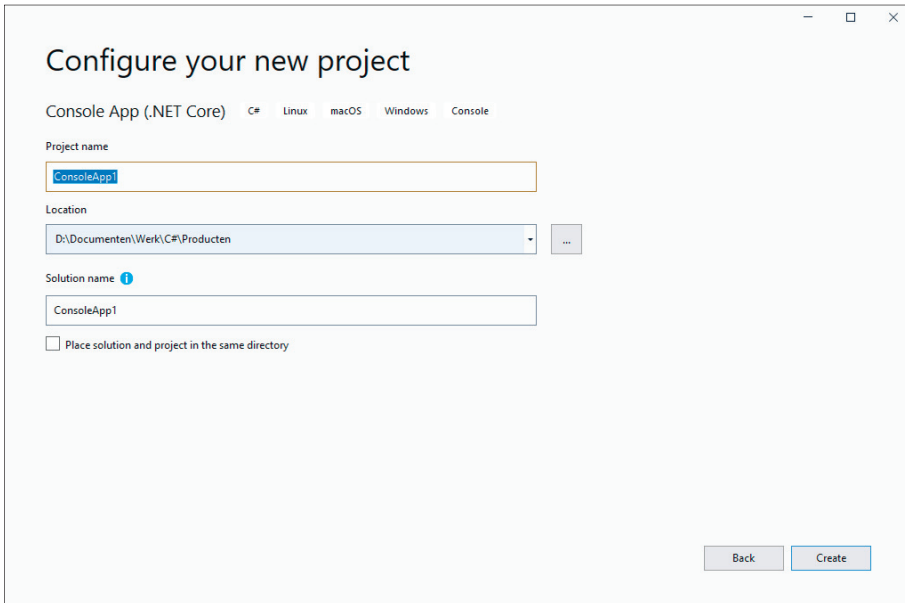
Schermafbeelding 4: Create a new project selectie C#

Let op! Bij sommige andere talen, zoals Visual Basic, staan exact dezelfde keuzemogelijkheden en kom je in hetzelfde scherm terecht, alleen de taal is anders!

Vervolgens dubbelklik je op *Console App (.NET Framework)*.



Schermafbeelding 5: Selectie console-app



Schermafbeelding 6: Configure your new project

Vervolgens geef je een naam aan je project. Standaard staat daar zoiets als `ConsoleApp1`.

Het volgende project dat je maakt krijgt dan de naam `ConsoleApp2`, enzovoort. Niet zo handig, want hoe weet je nu welk programma dat mooie spel is dat je ooit gemaakt hebt?

Er bestaat een mogelijkheid om een programma achteraf een nieuwe naam te geven, alleen gaat dat regelmatig mis, omdat de naam van het programma overall in jouw programma wordt gebruikt, ook op de achtergrond. Zelfs als je alleen de verschillende onderdelen van je programma (in C# heten die onderdelen ‘methoden’) een duidelijke naam geeft, bijvoorbeeld `BerekenDeVersnelling()`, dan heet die methode in werkelijkheid `ConsoleApp1.BerekenDeVersnelling()`.

We beginnen dus met het programma een betekenisvolle naam te geven, een naam waaraan je kunt zien wat het programma doet. Bijvoorbeeld `Mahjong` als je een Mahjongspel wilt maken, `Rekenmachine` als je een rekenmachine gaat maken, enzovoort. Ons programma heet `KennismakenProgrammeerOmgeving`.

Spelregels naamgeving

C# is hoofdlettergevoelig. Dit houdt in dat het programma met de naam `MijnProgramma` voor de computer iets totaal anders is dan het programma met de naam `mijnprogramma`!

Aan de naamgeving van verschillende onderdelen in een programma zijn spelregels verbonden voor de schrijfwijze. In de bijlage *Naming conventions C#* zijn de belangrijkste afspraken gebundeld.

Algemeen:

- Gebruik geen spaties in een naam.
- Gebruik geen liggend streepje (-) in een naam. Een laag streepje, of underscore (_), mag wel, maar wordt sterk afgeraden.
- Gebruik de PascalCase-notering voor variabelen. Dit houdt in dat je alle woorden aan elkaar schrijft en dat ieder nieuw woord met een hoofdletter begint:

`KennismakenProgrammeerOmgeving` of `BerekenVersnelling`.

De naam van het programma (`KennismakenProgrammeerOmgeving`) voer je in bij *Name*. Bij *Location* geef je de directory aan waar je je programma op wilt slaan. Standaard is dat de project-directory die `.NET` bij installatie aanmaakt. Deze kun je vinden in `c:\users\%gebruikersnaam%\source\repos`.

De *Solution Name* wordt automatisch voor je ingevuld als je de naam van het programma invoert. Voorlopig laten we dat zo. Een programma bestaat meestal uit een aantal kleinere programma's. Samen heten deze programmaatjes in de Visual-Studio-taal een *Solution* (oplossing).

De IDE

Je komt nu in de IDE zelf terecht, waarbij het scherm er ongeveer als Schermafbeelding 7 uit zal zien. Wijk het af, zijn sommige vensters niet actief, dan kun je ze aanzetten door in de knoppenbalk te kiezen voor *View* en daarna de naam van het ontbrekende venster op te zoeken.

Let op! Zie je niet direct de juiste naam? Er is ook een optie *Other Windows* met nog meer mogelijkheden.

Het scherm bevat de volgende belangrijke onderdelen (zie Schermafbeelding 7):

- 1 De knoppenbalk bovenaan het scherm. Hiermee bestuur je de IDE.
- 2 Direct daaronder een balk met veelvoorkomende functies waaronder de Debug-functie. Hier komen we zo op terug.
- 3 Links de Toolbox met de grafische bouwstenen voor je programma.
- 4 Rechtsboven de Solution Explorer waarin een overzicht wordt getoond van alle onderdelen en bestanden van je programma.
- 5 Rechtsonder het Properties-scherm. Hier stel je de eigenschappen van je programma in zoals achtergrondkleur, gebruikt lettertype, maar ook hoe je programma moet reageren op een muisklik.
- 6 Het Design-scherm. Hier ontwerp je je programma.