

Programmeren volgens het lagenmodel in C# met Visual Studio

HA-2266-04

Nascholingen Katholiek Onderwijs Vlaanderen
Bert Cauwenberg en Lieven Pauwels
Werkgroep Handel
2018



Programmeren volgens het lagenmodel in C# met Visual Studio

HA-2266-04

Nascholingen Katholiek Onderwijs Vlaanderen
Bert Cauwenberg en Lieven Pauwels
Werkgroep Handel
2018

Inhoud

1	Inleiding.....	5
1.1	Wat houdt het lagenmodel in?	5
1.1.1	Algemeen.....	5
1.1.2	Terminologie en afkortingen	7
1.1.2.1	Lagenmodel.....	7
1.1.2.2	Enkele afkortingen	7
1.2	Overzicht van de hoofdstukken	8
1.3	Situering in de leerstof programmeren	9
1.4	Didactische opbouw van de cursus	10
1.4.1	Einddoel van deze cursus.....	10
1.4.2	Didactische opbouw doorheen de hoofdstukken	11
1.4.3	Typografische afspraken en schematisering	12
1.4.4	Gebruik van class libraries uit HA-2265.....	13
2	Business- versus presentatie-laag.....	15
2.1	Aparte business-code	15
2.1.1	Principe.....	15
2.1.2	Voorbeeld 1: TelTekens	18
2.1.3	Voorbeeld 2: IBAN-nummer	20
2.1.4	Samenvatting	23
2.1.5	Oefeningen.....	24
2.2	De business laag volgens OOP-principes	27
2.2.1	Inleiding	27
2.2.2	Voorbeeld 1: Uitbreidingen voor TelTekens	28
2.2.2.1	Uitbreiding 1.....	28
2.2.2.2	Uitbreiding 2: Tussenversie A	29
2.2.2.3	Uitbreiding 2: Tussenversie B.....	30
2.2.2.4	Uitbreiding 2: Tussenversie C	31
2.2.2.5	Uitbreiding 2: Eindversie (D)	33
2.2.2.6	Uitbreiding 2: Samengevat	34
2.2.2.7	Uitbreiding 2: Besluit	36
2.2.2.8	Uitbreiding 3.....	36
2.2.3	Voorbeeld 2: IBAN-nummer	38
2.2.4	Samenvatting	40
2.2.5	Oefeningen.....	41
2.3	Klassen uit de business laag hergebruiken	43
2.3.1	Principe en gebruik van een class library	43
2.3.2	Voorbeeld: TelTekens.....	46
2.3.3	Algemene werkwijze	48
2.3.4	Oefeningen.....	49
2.3.4.1	Reeks 1	49
2.3.4.2	Reeks 2	50
2.4	Klassen uit de business laag afschermen	51
2.4.1	Principe en gebruik van een controller	51
2.4.2	Voorbeeld: MonsterBattle	52
2.4.2.1	Probleemstelling	52
2.4.2.2	Probleemanalyse.....	56
2.4.2.3	Business laag.....	58
2.4.2.4	Presentatie laag.....	59
2.4.2.5	Oefeningen.....	60
2.4.3	Algemene werkwijze	63
2.4.4	Oefeningen.....	65
2.5	Keuzemogelijkheden voor de “presentatie”	66
2.5.1	Beknopt overzicht	66
2.5.2	Enkele voorbeelden	67
2.5.2.1	Windows Forms.....	67
2.5.2.2	Windows Presentation Foundation (WPF)	69
2.5.2.3	Dynamische webpagina (met ASP.NET)	74

3	Nog een uitgewerkt voorbeeld: DiceGame	79
3.1	Probleemstelling	79
3.1.1	Algemene omschrijving	79
3.1.2	Spelregels	79
3.2	Eerste implementatie van het spel	80
3.2.1	Probleemanalyse	80
3.2.2	De klasse Dice	82
3.2.3	Functies en procedures	84
3.2.4	Voorbeeldcode DiceGame (versie 1).....	85
3.3	Tweede implementatie	88
3.3.1	Probleemanalyse	88
3.3.2	Hergebruik van code (m.b.v. class libraries)	88
3.3.3	Uitwerking voor DiceGame	90
3.3.4	Schematisch overzicht (DiceGame versie 2)	92
3.3.5	Schematisch overzicht (algemeen)	92
4	Design patterns, best practices en stappenplannen	93
4.1	Werkwijze in Visual Studio	93
4.1.1	Stappenplan.....	93
4.1.2	Tips i.v.m. naamgevingen	93
4.1.3	Schematische werkwijze (versie 1)	94
4.1.4	Noot	94
4.2	Oefeningen (reeks 1)	95
4.2.1	Opdrachten bij DiceGame	95
4.2.2	Opdrachten met uitgewerkt algoritme	96
4.2.3	Open opdrachten.....	100
4.2.4	Remediëringsopdrachten	103
4.2.5	Opmerkingen en toelichtingen bij enkele oefeningen	106
4.3	Design patterns.....	108
4.3.1	MVC.....	108
4.3.2	MVP	110
4.3.3	MVVM.....	110
4.3.4	Bronnen en afspraken	112
4.3.5	Stappenplan (versie 2 - met controller)	113
4.3.6	Schematische werkwijze (versie 2 - met controller)	114
4.4	Best practices.....	115
4.4.1	Naamgeving	115
4.4.2	Documenteren van zelf gedefinieerde klassen	116
4.4.2.1	Annotatiecommentaar	116
4.4.2.2	XML-gestructureerde commentaar.....	117
4.4.3	Verstandig gebruik van IntelliSense in Visual Studio	118
4.5	Uitwerking voor DiceGame	119
4.5.1	Klassendiagram	119
4.5.2	Voorbeeldcode business laag	120
4.5.3	Voorbeeldcode presentatie laag (m.b.v. een controller).....	121
4.6	Oefeningen (reeks 2)	122
4.6.1	Opdrachten bij “DiceGame”	122
4.6.2	Opdrachten met uitgewerkt algoritme	122
4.6.3	Open opdrachten.....	122
4.6.4	Remediëringsopdrachten	125
5	Business laag volgens OOP-principes	127
5.0	Inleiding	127
5.0.1	Bespreking DiceGame-implementatie.....	127
5.0.2	Andere implementatie voor de DiceGame business code.....	128
5.0.3	OOP-analyse voor het DiceGame-spel.....	129
5.1	Voorbeeld: DiceGame	130
5.1.1	UML-klassendiagram	130
5.1.2	Uitleg bij de klassen.....	131
5.1.3	Voorbeeldcode business laag	133
5.1.4	Oefeningen	135
5.2	Bijkomende voorbeelden	136

5.2.1	BtwApp	136
5.2.2	Verbruik	137
5.2.3	Landmeter	140
5.3	Oefeningen.....	142
5.3.1	Herwerken van voorgaande opgaven.....	142
5.3.2	Nieuwe opgaven	143
5.3.3	Uitbreiden van voorgaande opgaven	146
5.3.4	Addendum: UML-diagrammen	148
5.3.5	Besluit: slotopmerkingen.....	159

