

Module
Data Structures and Algorithms Using
C#

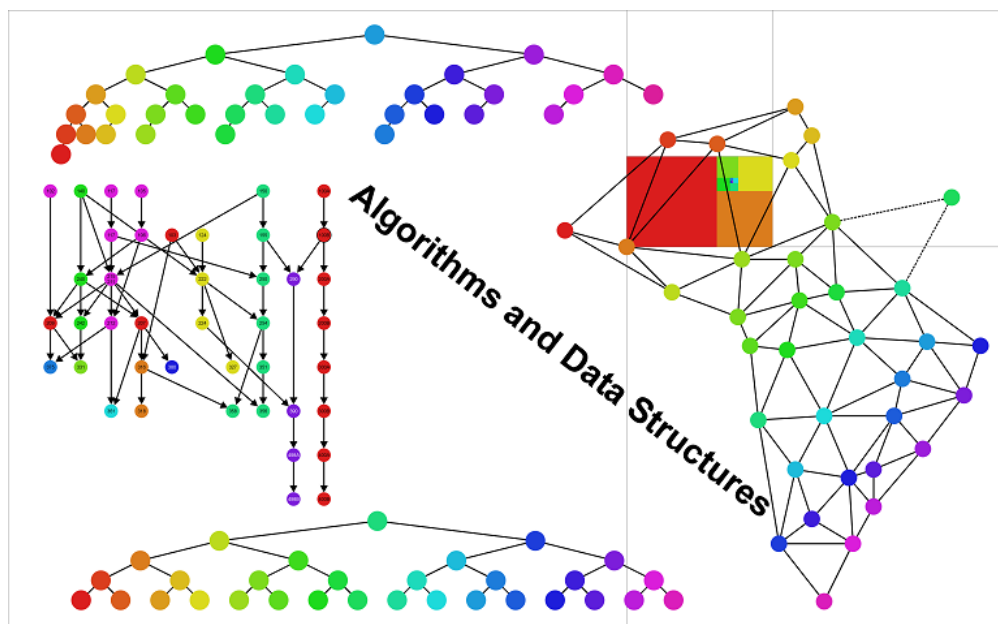
Studiejaar 2
Code AD

Stenden Hogeschool
Van Schaikweg 94
Po box 2080
7801 CB Emmen
tel: 0591 853100
www.stenden.com

© Stenden Hogeschool

Module Data Structures and Algorithms Using C#

Version 3.0: 4 februari 2017



Studiejaar 2016-2017

Module coördinator:	Auteur:
Naam: Jeroen Pijpker	Naam: Jeroen Pijpker
e-mail: jeroen.pijpker@stenden.com	e-mail: jeroen.pijpker@stenden.com

Voorwoord

In de modules Java en C# heb je de beginselen van het programmeren geleerd. Je hebt toen kennis gemaakt met diverse datatypen (simple types en objecten). In de module die nu voor je ligt, leer je hoe je deze datatypen kunt combineren tot zogenaamde Data-structures. Verder ga je je bezig houden met algoritmen om deze data-structures te bewerken. We gaan gebruik maken van de programmeertaal C#.

Voor het ontwikkelen van applicaties voor deze module maken we gebruik van de programmeertaal C# in de ontwikkelomgeving Visual Studio 2015.

Aan het eind van deze module leveren de studenten een API op met daarin een aantal sorteeralgoritmen en zoekalgoritmen.

De studielast voor deze module bedraagt 84 uur (3 EC).

Door evaluaties van studenten is het programma zodanig aangepast dat er meer tijd beschikbaar is voor de module Data Structures and Algorithms Using C#.

Jeroen Pijpker

Emmen, 12 mei 2016

Inhoud

1	INTRODUCTIE	7
1.1	MODULE RATIONALE	7
1.2	COMPETENTIES	8
1.3	MODULE THEMA	9
1.4	MODULE DOELSTELLINGEN	9
1.5	VOORKENNIS	9
1.6	CONVENTIES	9
2	TOETSING	10
2.1	ALGEMEEN	10
2.2	BEOORDELING GROEPSOPDRACHT	10
2.3	ACTIEVE PARTICIPATIE	11
2.4	BEOORDELING WERKCOLLEGES	11
2.5	MODULE HERKANSING	11
3	PROGRAMMA.....	12
3.1	INTRODUCTIE	12
3.1.1	<i>Hoorcolleges</i>	12
3.1.2	<i>Werkcolleges</i>	12
3.2	PROGRAMMAOVERZICHT.....	13
3.3	WEEKPROGRAMMA'S.....	14
3.3.1	<i>Introductiecollege</i>	14
3.3.2	<i>Hoorcollege 1</i>	14
3.3.3	<i>Werkcollege 1</i>	14
3.3.4	<i>Hoorcollege 2</i>	15
3.3.5	<i>Werkcollege 2</i>	15
3.3.6	<i>Hoorcollege 3</i>	16
3.3.7	<i>Werkcollege 3</i>	16
3.3.8	<i>Hoorcollege 4</i>	17
3.3.9	<i>Werkcollege 4</i>	17
3.3.10	<i>Hoorcollege 5</i>	18
3.3.11	<i>Werkcollege 5</i>	18
3.3.12	<i>Hoorcollege 6</i>	19
3.3.13	<i>Werkcollege 6</i>	19
3.3.12	<i>Hoorcollege 7</i>	20
3.3.13	<i>Werkcollege 7</i>	20
3.3.14	<i>Hoorcollege 8</i>	21
3.3.15	<i>Werkcollege 8</i>	21
4	STRUCTUUR & ORGANISATIE.....	22
4.1	MODULE CONTACT UREN	22
5	LITERATUUR/PROGRAMMATUUR	23
5.1	VERPLICHTE LEERSTOF	23
5.2	NASLAGWERK.....	23

5.3	PROGRAMMATUUR	23
6	MODULE EVALUATIE	24
7	BIJLAGEN.....	25
	BIJLAGE 1: CODE CONVENTIES	26
	BIJLAGE 2: EINDOPDRACHT	28
	BIJLAGE 3: SCORING RUBICS	30

1 Introductie

1.1 Module rationale

Voordat een programmeur een computerprogramma kan schrijven om een probleem op te lossen, moet een aanpak (een reeks stappen) bedacht worden die het programma kan volgen om het probleem op te lossen. Zo een reeks stappen wordt een algoritme genoemd. In deze module worden algoritmen besproken voor het sorteren en het zoeken in een verzameling gegevens.

1.2 Competenties

Binnen deze module wordt als beginnend beroepsprofessional gewerkt aan een drietal competenties die betrekking hebben op het analyseren, ontwerpen en realiseren van applicaties op niveau 2.

	Beheren	Analyseren	Adviseren	Ontwerpen	Realiseren
Gebruikers interactie					
Bedrijfsprocessen					
Infrastructuur					
Software	<ul style="list-style-type: none"> Inrichten en gebruikmaken van beheersysteem ter ondersteuning van softwareontwikkeling in teamverband. (niveau 1)	<ul style="list-style-type: none"> Verzamelen en valideren van functionele eisen voor een softwaresysteem met één belanghebbende en volgens een standaardmethode. Acceptatiecriteria definiëren voor bovengenoemde functionele eisen. Een requirementanalyse uitvoeren voor een softwaresysteem met verschillende belanghebbenden, rekening houdend met de kwaliteitseigenschappen. Een analyse uitvoeren om functionaliteit, ontwerp, interfaces e.d. van een bestaand systeem of bestaande component te formuleren en te valideren. Een acceptatietest opstellen aan de hand van kwaliteitseigenschappen. (niveau 2)		<ul style="list-style-type: none"> Opstellen van een ontwerp voor een softwaresysteem, rekening houdend met het gebruik van bestaande componenten en libraries; gebruik maken van ontwerpqualiteitscriteria. Vaststellen van de kwaliteit van het ontwerp, bijvoorbeeld door toetsing of prototyping, rekening houdend met de geformuleerde kwaliteitseigenschappen. Testontwerpen opstellen volgens een gegeven teststrategie. (niveau 2)	<ul style="list-style-type: none"> Bouwen en beschikbaar stellen van een softwaresysteem dat bestaat uit meerdere subsystemen, hierbij gebruik makend van bestaande componenten. Integreren van softwarecomponenten in een bestaand systeem, waarbij o.a. de integriteit en systeemprestaties bewaakt worden. Uitvoeren van regressietesten. - Uitvoeren van en rapporteren over unit-, integratie- en systeemtesten. (niveau 2)
Hardware interfacing					

1.3 Module thema

1.4 Module doelstellingen

Na afloop van deze module kan de student:

1. Een gegeven werkberoepssituatie analyseren.
2. Een programma realiseren waarin de volgende onderwerpen worden gebruikt: (Linked)List, Stack, Queue, String, Tree, Graph en Hash table.
3. Gebruik maken van algoritmen om genoemde data-structures te onderhouden en te benaderen.
4. Een programma realiseren die gebruik maakt van verschillende sorteeralgoritmen.
5. Een traject opzetten voor het ontwikkelen van een applicatie die gebruik maakt van sorteer en zoek algoritmen.
6. Een API ontwikkelen die gebruik maakt van sorteer en zoek algoritmen.

1.5 Voorkennis

De student dient de module Java Finals te hebben afgerond om deel te nemen aan de module Data Structures and Algorithms Using C#.

1.6 Conventies

Een paar conventies met betrekking tot lettertypen zijn opgesteld om onderscheid te kunnen maken tussen tekst en programmeercode.

Normale tekst	Verdana. (tekengrootte 10)
Programma code	Courier new. (tekengrootte 10)
Programma output	Courier new. (tekengrootte 10)

Afspraken met betrekking tot naamgeving en indeling van programmacode kun je vinden in bijlage 1.

2 Toetsing

2.1 Algemeen

Om te beoordelen of je de doelstellingen van deze module hebt behaald, wordt er een groepsopdracht (praktijkttoets) gehanteerd.

Tabel 2.1 geeft een overzicht van de eindtoets met relevante informatie over de normering, punten en credits.

De beoordeling van de groepsopdracht zal plaatsvinden door middel van één beoordelingsformulier. De groepsopdracht moet worden ingeleverd op de vrijdag voor de toetsweek.

Deze module wordt met een voldoende beoordeeld als de norm is gehaald, welke wordt bepaald door middel van de scoring rubrics uit bijlage 3.

Tabel 2.1 Overzicht toetsing

Test method	Max. points	Norm %	Norm in points	Credits	Deadline	Resit
Groepsopdracht	100	55%	55	3 EC		
Total	100	55%	55	3 EC		

2.2 Beoordeling groepsopdracht

Alle stappen (onderdelen) van de groepsopdracht dienen in groepen van 4 personen (bij oneven aantal studenten zal er 1 vijftal gevormd worden) uitgevoerd en besproken te worden met de docent. **Als tijdens de bespreking blijkt dat studenten de opdracht niet zelf hebben uitgevoerd, wordt de groepsopdracht met een onvoldoende beoordeeld.**

De groepseindopdracht bestaat uit een drietal stappen (zie ook bijlage 3):

1. DLL met de verschillende algoritmen en datastructuren,
2. Klassendiagram,
3. Uitprogrammering van de DLL.

Voor een herkansing van de groepseindopdracht wordt eenmalig een vervangende opdracht verstrekt waar dezelfde eisen voor gelden.

De beoordeling van de praktijkttoets zal plaatsvinden aan de hand van het beoordelingsformulier (Scoring Rubrics) welke te vinden is in bijlage 2.

De module Data Structures and Algorithms Using C# wordt met een voldoende beoordeeld als voor de groepseindopdracht de norm is gehaald.

2.3 Actieve participatie

De student dient alle opdrachten voorafgaand aan de praktijktoets af te ronden in de daarvoor betreffende weken. De mogelijkheid voor het aftekenen van deze opdrachten heeft de student tijdens de ingeroosterde begeleide werkcolleges.

2.4 Beoordeling werkcolleges

Tijdens de werkcolleges zullen de volgende onderdelen beoordeeld worden:

1. De aanwezigheid van de student;
De student moet minimaal bij 75% van de werkcolleges de volledige tijd aanwezig zijn geweest.
2. De te maken opgaven uit het boek;
De student moet minimaal 75% scoren (v.b.: dus 3 uit 4 vragen) uit de door de docent geselecteerde vragen.

Al deze 2 onderdelen moeten met een voldoende zijn beoordeeld voor men kan deelnemen aan de afsluitende praktijktoets.

2.5 Module herkansing

Studenten die niet slagen voor de groepseindopdracht kunnen een herkansing doen.

3 Programma

3.1 Introductie

In deze module leer je hoe je deze datatypen kunt combineren tot zogenaamde Data-structures. Verder ga je je bezighouden met algoritmen om deze data-structures te bewerken. We gaan gebruik maken van de programmeertaal C#.

Deze module bevat een aantal (wekelijkse) hoor- en werkcolleges. De student dient hoofdstukken zelfstandig door te werken. Dit dient te gebeuren door middel van zelfstudie in combinatie met het hoorcollege en werkcollege.

Tijdens de hoorcolleges zal de docent vaak een demonstratie geven en heeft de student de mogelijkheid om problemen aan te stippen. Na het hoorcollege kan de student zelfstandig aan de slag met de opdrachten.

Tijdens de werkcolleges heeft de student de mogelijkheid om problemen met de opdrachten te bespreken met de docenten en tevens kunnen dan opdrachten afgetekend worden.

Het geheel zal afgesloten worden met een praktijktoets.

3.1.1 Hoorcolleges

Het is de bedoeling dat de student door middel van zelfstudie de hoorcolleges voorbereid en daar eventuele vragen kan stellen.

Tijdens de hoorcolleges, welke een interactieve vorm hebben, komen de verschillende facetten van Data Structures and Algorithms Using C# aan de orde. Dat kan bijvoorbeeld een introductie van het geheel zijn of de uitleg van algoritmen.

Regelmatig zal echter een specifiek (programmeer)probleem aan de orde komen en zal de docent de oplossing hiervan stapsgewijs voordoen/uitleggen.

3.1.2 Werkcolleges

Het is de bedoeling dat de student door middel van zelfstudie de werkcolleges voorbereid en daar eventuele vragen kan stellen. Ook worden werkcolleges gebruikt om de voortgang van de student te toetsen.

Tijdens de werkcolleges heeft de student de mogelijkheid om de onderdelen van de bijbehorende week af te laten tekenen. Het kan vanzelfsprekend eens voorkomen dat een student het werk van een week niet heeft af gekregen en dit kan dan in de week daaropvolgend worden afgetekend. De mogelijkheid bestaat echter niet dat een student aan het eind van de periode nog een groot gedeelte moet afronden.

Voor de werkcolleges geldt een aanwezigheidsplicht. De docent is voor delen van het werkcollege beschikbaar.

3.2 Programmaoverzicht

In onderstaand overzicht is een overzicht te vinden van de wekelijkse activiteiten.

Wk	Taak nr.	Studie activiteit
1	3.3.1	Introductiecollege
	3.3.2	Hoorcollege 1 (Boek C#).
	3.3.3	Werkcollege 1 (Boek C#).
2	3.3.4	Hoorcollege 2 (Boek Data Structures H1,2,3,4)
	3.3.5	Werkcollege 2 (Opdrachten boek Data Structures H1,2,3,4)
3	3.3.6	Hoorcollege 3 (Boek Data Structures H5,6,7)
	3.3.7	Werkcollege 3 (Opdrachten boek Data Structures H5,6,7)
4	3.3.8	Hoorcollege 4 (Boek Data Structures H8,9,10)
	3.3.9	Werkcollege 4 (Opdrachten boek Data Structures H8,9,10)
5	3.3.10	Hoorcollege 5 (Boek Data Structures H11,12,13)
	3.3.11	Werkcollege 5 (Opdrachten boek Data Structures H11,12,13)
6	3.3.12	Hoorcollege 6 (Boek Data Structures H14,15,16)
	3.3.13	Werkcollege 6 (Opdrachten boek Data Structures H14,15,16)
7	3.3.14	Hoorcollege 7 (Boek Data Structures H17)
	3.3.15	Werkcollege 7 (Opdrachten boek Data Structures H17)
8	3.3.16	Hoorcollege 8 (Opbouw DLL)
	3.3.17	Werkcollege 8 (Programmeren DLL)
9	3.3.18	Herkansing

3.3 Weekprogramma's

3.3.1 Introductiecollege	
Week:	1
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	<ul style="list-style-type: none"> De student verkrijgt een overzicht van de inhoud van de module Data Structures and Algorithms Using C#. De student doorziet wat de werkwijze en de beoordeling binnen de module Data Structures and Algorithms Using C# is.
Inhoud:	Tijdens het introductiecollege krijg je instructie over onder andere de werkwijze/werkvormen, beoordeling, materiaal en inhoud van de module Data Structures and Algorithms Using C#.
Vorbereiden:	
Individuele opdrachten:	

3.3.2 Hoorcollege 1	
Week:	1
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	<ul style="list-style-type: none"> Herhaling C#
Inhoud:	Herhaling C#
Vorbereiden:	
Individuele opdrachten:	

3.3.3 Werkcollege 1	
Week:	1
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	<ul style="list-style-type: none"> Herhaling C#
Inhoud:	Tijdens dit werkcollege kan de student aan de slag met C#.
Vorbereiden:	Bestuderen/herhalen lesstof C#.
Individuele opdrachten:	

3.3.4 Hoorcollege 2	
Week:	2
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	<ul style="list-style-type: none">• Arrays and ArrayLists• Basic Sorting Algorithms (Bubble Sort, Selection Sort, Insertion Sort)• Basic Searching Algorithms (Sequential Searching, Binary Search)
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 1 t/m 4 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 1 t/m 4 boek Data Structures

3.3.5 Werkcollege 2	
Week:	2
Werkvorm:	Werkcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 1 t/m 4 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 1 t/m 4 boek Data Structures

3.3.6 Hoorcollege 3	
Week:	3
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 5 t/m 7 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 5 t/m 7 boek Data Structures

3.3.7 Werkcollege 3	
Week:	3
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 5 t/m 7 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 5 t/m 7 boek Data Structures

3.3.8 Hoorcollege 4	
Week:	4
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 8 t/m 10 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 8 t/m 10 boek Data Structures

3.3.9 Werkcollege 4	
Week:	4
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 8 t/m 10 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 8 t/m 10 boek Data Structures

3.3.10 Hoorcollege 5	
Week:	5
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 11 t/m 13 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 11 t/m 13 boek Data Structures

3.3.11 Werkcollege 5	
Week:	5
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 11 t/m 13 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 11 t/m 13 boek Data Structures

3.3.12 Hoorcollege 6	
Week:	6
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 14 t/m 16 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 14 t/m 16 boek Data Structures

3.3.13 Werkcollege 6	
Week:	6
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 14 t/m 16 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 14 t/m 16 boek Data Structures

3.3.12 Hoorcollege 7	
Week:	7
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 17 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 17 boek Data Structures

3.3.13 Werkcollege 7	
Week:	7
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Bestudeer hoofdstuk 17 boek Data Structures
Individuele opdrachten:	Opdrachten hoofdstuk 17 boek Data Structures

3.3.14 Hoorcollege 8

Week:	7
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	
Inhoud:	
Vorbereiden:	Opbouw DLL
Individuele opdrachten:	Opbouw DLL Zie bijlage 3

3.3.15 Werkcollege 8

Week:	7
Werkvorm:	Werkcollege
Duur:	2
Lesdoelen:	
Inhoud:	
Vorbereiden:	Programmeren DLL
Individuele opdrachten:	<p>Programmeren DLL met daarin:</p> <ul style="list-style-type: none"> • Tijdmetering • Gebruik Generics • Arraylist • BubbleSort (H3) • InsertionSort (H3) • SmartBubbleSort (H3) • BinarySearch (H4) • MinMax (H4) • SequentialSearch (H4) • Queue (H5) • PriorityQueue (H5) • Stack (H5) • BucketHash (H10) • LinearHash (H10) • QuadraticHash (H10) • Circularlist (H11) • DoublyLinkedList (H11) • DoublyNode (H11) • Iterator (H11) • LinkedList (H11) • BinarySearchTree (H12) <p>Met code om de verschillende onderdelen te testen en te beoordelen op het presteren.</p> <p>Zie bijlage 3</p>

4 Structuur & Organisatie

4.1 Module contact uren

Het onderstaande schema geeft een overzicht van alle contacturen in deze module.

Daarnaast wordt van studenten verwacht dat zij hun eigen (project)bijeenkomsten plannen waar zij kunnen werken aan de opdrachten. Dit geldt tevens voor de tijd die de student nodig heeft om (individuele) opdrachten voor te bereiden en te maken. Tevens geeft dit schema een goed overzicht van de verwachte studiebelasting per student.

Module Data Structures and Algorithms Using C#: Student Contacturen en Student Belastinguren																					
Werkvorm	Aantal groepen	Week 1		Week 2		Week 3		Week 4		Week 5		Week 6		Week 7		Week 8		Week 9		Totaal	
		SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU
HC	4	1	3	1	3	1	3	1	3	0	0	0	0	0	0	0	0	0	0	4.00	12.00
WC	2	0	0	0	0	0	0	0	0	1	3	1	3	1	3	0	0	0	0	3.00	9.00
ET	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1.00	3.00
ZS	1	0	6	0	7	0	7	0	7	0	7	0	7	0	7	0	0	0	0	-	48.00
		1	9.00	1	10.00	1	10.00	1	10.00	1	10.00	1	10.00	1	10.00	-	-	1	3.00	8.00	72.00

Table 4.1: Student contact uren (SCU) per week:

HC = Hoorcollege
 WC = Werkcollege
 ET = Eind toets
 ZS = Zelfstudie uren

SCU = Student Contact Uren (45 minuten)
 SBU = Studie Belasting Uren (60 minuten)

5 Literatuur/programmatuur

5.1 Verplichte leerstof

- Deze module
- Boek: Data Structures and Algorithms Using C#
Schrijver: Michael McMillan
Uitgever: Cambridge University Press
ISBN-10: 0-52-167015-2
ISBN-13: 978-0-521670-15-9

5.2 Naslagwerk

- C# Programming Guide
<http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>

5.3 Programmatuur

- Visual Studio 2015

6 Module evaluatie

De module zal worden geëvalueerd door middel van een vragenlijst aan het eind van de module. Deze vragenlijst bevat alle onderdelen van de module inclusief organisatorische aspecten, inhoud, kwaliteit van onderwijzend personeel, etc.

Wij willen je vriendelijk verzoeken deel te nemen aan deze evaluatie. De resultaten van deze evaluatie worden gebruikt om de volgende versie van deze module te verbeteren.

7 Bijlagen

Bijlage 1: Code conventies

De volgende code conventies worden binnen deze module gehanteerd.

1. Naming

1.1 Use meaningful names.

Use descriptive names for all identifiers (names of classes, variables and methods. Avoid ambiguity. Avoid abbreviations.
Simple mutator methods should be named `setSomething(...)`.
Simple accessor methods should be named `getSomething(...)`.
Accessor methods with Boolean return values are often called `isSomething(...)`, for example `isEmpty()`.

1.2 Class names start with a capital letter.

1.3 Class names are singular nouns.

1.4 Methods and variable names start with lowercase letters.

All the – class, method and variable names – use capital letters in the middle to increase readability of compound identifiers, e.g. `numberOfItems`.

1.5 Constants are written in UPPERCASE.

Constants occasionally use underscores to indicate compound identifiers:
`MAXIMUM_SIZE`

2. Layout

2.1 One level of indentation is four spaces.

2.2 All statements within a block are intended one level.

2.3 Braces for classes and methods are alone on one line.

The braces for class and method blocks are on separate lines and are at the same indentation level, for example:

```
public int getAge()
{
    statements
}
```

2.4 Also for all other blocks, are alone on one line.

The braces for all other blocks are on separate lines and are at the same indentation level, for example:

```
while(condition)
{
    statements
}
```

```
if(condition)
{
    statements
}
else
{
    statements
}
```

2.5 Always use braces in control structures

Braces are used in if-statements and loops even if the body is only a single statement.

2.6 Use a space before the opening brace of a control structure's block.

2.7 Use a space around operators.

2.8 Use a blank line between methods (and constructors).

Use blank lines to separate logical blocks of code. This means at least between methods, but also between logical parts within a method.

Bijlage 2: Eindopdracht

Werkwijze

De eindbeoordeling is een eindopdracht waar de student in een groep van maximaal 4 personen een probleem moet oplossen. Dit probleem moet in week 6 worden aangedragen door de studenten en zal minimaal de volgende onderdelen **moeten** bevatten:

- C# .net 4.5 of hoger
- DLL
- Tijdmetering
- Gebruik Generics
- Arraylist
- BubbleSort (H3)
- InsertionSort (H3)
- SmartBubbleSort (H3)
- BinarySearch (H4)
- MinMax (H4)
- SequentialSearch (H4)
- Queue (H5)
- PriorityQueue (H5)
- Stack (H5)
- BucketHash (H10)
- LinearHash (H10)
- QuadraticHash (H10)
- Circularlist (H11)
- DoublyLinkedList (H11)
- DoublyNode (H11)
- Iterator (H11)
- LinkedList (H11)
- BinarySearchTree (H12)
- Een net verzorgde GUI waarmee het programma te bedienen is.

Dit probleem moet ter goedkeuring worden aangeboden aan de docent die er eventueel een wijziging in aan zal brengen. **Na goedkeuring door de docent** mag er gewerkt worden aan de oplossing. Het kan zijn dat de docent bij meerdere opdrachten de gegeven opdrachten gaat herverdelen. De opdracht wordt vervolgens verstrekt aan de studenten en zij dienen deze zelfstandig uit te werken. Deze tijd kunnen de studenten zelf plannen.

De docenten verwachten een ontwerp en de volledige realisatie van het probleem.

Met de realisatie mag de groep beginnen nadat het ontwerp door de docent is goedgekeurd.

De beoordeling van de eindopdrachten zal plaatsvinden aan de hand van het beoordelingsformulier (Scoring Rubrics) welke te vinden is in bijlage 3. De eindopdracht moet tevens worden ingeleverd bij de docent.

De eindopdracht moet worden afgesloten met een presentatie.

De deadline voor het inleveren van de opdracht is de vrijdag van moduleweek 8 voor 16:30.

Bijlage 3: Scoring rubrics

Beoordelingsformulier: Data Structures and Algorithms Using C#			
Docenten:			
Datum:			
Onderdeel	Beschrijving	Weging	Cijfer
Kwaliteit code:	<ul style="list-style-type: none">• De kwaliteit van de code en commentaar.• Gebruik gemaakt van versiebeheer.	20%	
Software:	<ul style="list-style-type: none">• Inrichten en gebruikmaken van beheersysteem ter ondersteuning van softwareontwikkeling in teamverband.• Verzamelen en valideren van functionele eisen voor een softwaresysteem met één belanghebbende en volgens een standaardmethode.• Een ontwerp maken voor een softwaresysteem met modelleertechnieken volgens een standaardmethode.• Een eenvoudig softwaresysteem bouwen, testen en beschikbaar stellen.	30%	
Beroepsproduct:	<ul style="list-style-type: none">• Werkend eindproduct	50%	
Eindcijfer:			
Opmerkingen:			

Onderdeel	<4	4 - 5,4	5,5 - 6,9	7 - 8,4	>8,5
<ul style="list-style-type: none"> • Kwaliteit code / kwaliteit commentaar 	<ul style="list-style-type: none"> • Geen code • Geen commentaar 	<ul style="list-style-type: none"> • Code • Geen commentaar 	<ul style="list-style-type: none"> • Code • Nauwelijks commentaar 	<ul style="list-style-type: none"> • Code logisch • Commentaar 	<ul style="list-style-type: none"> • Code logisch en verantwoording okay • Commentaar
<ul style="list-style-type: none"> • Software: 	<ul style="list-style-type: none"> • De projectgroep heeft ter ondersteuning van de software-ontwikkeling in teamverband geen gebruik gemaakt van een ingericht beheerssysteem. • Functionele eisen zijn door de projectgroep verzameld en gevalideerd volgens een standaardmethode • De projectgroep heeft een geen ontwerp gemaakt voor een softwaresysteem met modelleer-technieken volgens een standaard-methode. • De projectgroep heeft een eenvoudig softwaresysteem gebouwd, getest en beschikbaar gesteld. 	<ul style="list-style-type: none"> • De projectgroep heeft ter ondersteuning van de software-ontwikkeling in teamverband onvoldoende gebruik gemaakt van een ingericht beheerssysteem. • Functionele eisen zijn door de projectgroep verzameld en gevalideerd volgens een standaardmethode • De projectgroep heeft een onvoldoende voor een softwaresysteem met modelleertechnieken volgens een standaardmethode. • De projectgroep heeft een eenvoudig softwaresysteem gebouwd, getest en beschikbaar gesteld. 	<ul style="list-style-type: none"> • De projectgroep heeft ter ondersteuning van de software-ontwikkeling in teamverband voldoende gebruik gemaakt van een ingericht beheerssysteem. • Functionele eisen zijn door de projectgroep verzameld en gevalideerd volgens een standaardmethode • De projectgroep heeft een voldoende ontwerp gemaakt voor een softwaresysteem met modelleer-technieken volgens een standaard-methode. • De projectgroep heeft een eenvoudig softwaresysteem gebouwd, getest en beschikbaar gesteld. 	<ul style="list-style-type: none"> • De projectgroep heeft ter ondersteuning van de software-ontwikkeling in teamverband goed gebruik gemaakt van een ingericht beheerssysteem. • Functionele eisen zijn door de projectgroep verzameld en gevalideerd volgens een standaardmethode • De projectgroep heeft een goed ontwerp gemaakt voor een softwaresysteem met modelleer-technieken volgens een standaard-methode. • De projectgroep heeft een eenvoudig softwaresysteem gebouwd, getest en beschikbaar gesteld. 	<ul style="list-style-type: none"> • De projectgroep heeft ter ondersteuning van de software-ontwikkeling in teamverband uitstekend gebruik gemaakt van een ingericht beheerssysteem. • Functionele eisen zijn door de projectgroep verzameld en gevalideerd volgens een standaardmethode • De projectgroep heeft een uitstekend ontwerp gemaakt voor een softwaresysteem met modelleer-technieken volgens een standaard-methode. • De projectgroep heeft een eenvoudig softwaresysteem gebouwd, getest en beschikbaar gesteld.
<ul style="list-style-type: none"> • Beroepsproduct 	<ul style="list-style-type: none"> • Geen eindresultaat 	<ul style="list-style-type: none"> • Resultaat maar werking niet in orde van meer dan 75% van de onderdelen 	<ul style="list-style-type: none"> • Resultaat maar werking niet in orde van meer dan 50% van de onderdelen 	<ul style="list-style-type: none"> • Resultaat maar werking niet in orde van minder dan 25% van de onderdelen 	<ul style="list-style-type: none"> • Resultaat maar werking niet in orde van minder dan 15% van de onderdelen