

Tentamenvragen **Herkansing** Programmeren - Flex Software Engineering 1 – T5 2020/2021

Datum: juli 2021 (T5)

Toets-id: T5-0063

Toetscode: ICTB20D1SOEN1-A

Toets: IC_HBO_ICT_HK_1_Programmeren C#(flex)

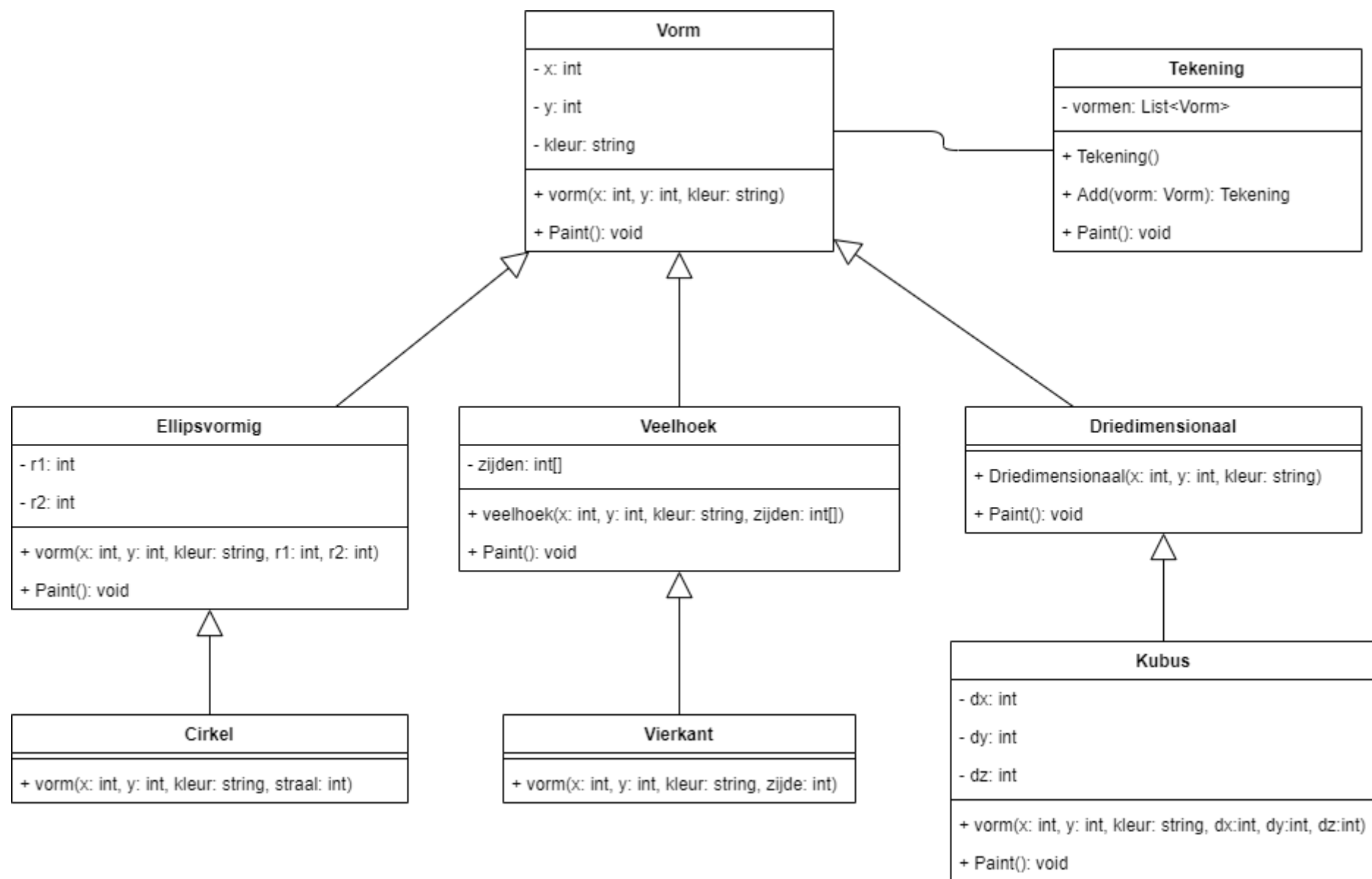
Opleiding	HBO ICT - Flex	Toetsvorm	Tentamen
Titel Toetseenheid	Programmeren	Datum	Juli 2021
Titel Toets	Programmeren C#	Examinator	Martin Molema
Module	Software Engineering 1	Tijdsduur	2,5 uur
	S2 2020/2021	Cijfer	(90pt + 10pt) / 10
Maximale score	90 punten	Pagina's	10

Toegestane hulpmiddelen:

- Toegang tot internet
- Geïntegreerde ontwikkelomgevingen zoals (maar niet uitsluitend)
 - JetBrains Rider
 - Microsoft Visual Studio
 - Visual Studio Code

Vraag 1 (30 punten)

Zie onderstaande klassendiagram



In bovenstaande klassendiagram worden een aantal klassen beschreven.

Vraag 1a (theorie) (5+5 punten)

De antwoorden op onderstaande vragen mag je opnemen in de Visual Studio Solution (TentamenFlexHerkansing.sln), project '**Vormen**', program.cs.

1. Wat is de term die de relatie tussen klasse 'Vorm' en 'Veelhoek' beschrijft?
2. In de klasse 'Tekening' is een variabele 'vormen' van het type `List<Vorm>`. Wat is de gebruikelijke plaats om deze variabele te initialiseren?

Vraag 1b (20 punten)

Het bovenstaande klassediagram laat een verzameling van verschillende soorten (klassen) vormen zien. Bouw deze klassenstructuur met onderstaande informatie.

1. Een **vorm** wordt beschreven met een coördinaat (x,y) en een kleur
2. De klasse '**Ellipsvormig**' gebruikt (naast coördinaat en kleur) twee parameters: straal 1 en straal 2 .
3. Een **Cirkel** is een **Ellipsvormig**-object waarbij de waarden voor straal 1 en straal 2 gelijk zijn.
4. Een **Veelhoek** heeft een array met zijden die de lengtes van die zijden bevatten
5. Een **Vierkant** is een veelhoek waarbij de vier zijden dezelfde waarde hebben
6. Een **Kubus** is een **driedimensionale vorm**, waarbij de lengtes van de drie ribben opgegeven moet worden (d1, d2 en d3)¹

De functie 'Paint' wordt later toegelicht.

De klasse tekening bevat een lijst met vormen. Je kunt vormen toevoegen met de functie

```
public Tekening Add(Vorm vorm)
```

Let op het juiste return type (Tekening) voor method-chaining.

Vraag 2 (20 punten)

In deze vraag gaan we met de gemaakte klassen aan de slag.

2a: Vormen aanmaken (5 punten)

Nu je het klassendiagram hebt geïmplementeerd ga je vormen maken. Maak de volgende vormen aan in *Program.cs*:

- 2 cirkels
- 2 kubussen
- 2 vierkanten

¹ Letter D voor 'Distance'

2b: De tekening maken (5 punten)

Maak een instantie van de klasse `Tekening` en voeg de vormen toe door middel van de methode `'Add'`. Eventueel gebruik je method-chaining om je code te verkorten.

2c: Tekening tekenen (10 punten)

In deze opdracht ga je zorgen dat de tekening op het scherm komt. Misschien niet heel mooi, want het gaat alleen om het afdrukken van de informatie van de tekening.

Implementeer de functie `'public void Paint()'` in de klasse `'Tekening'` zodat deze informatie over alle vormen uit de lijst met vormen (`List<Vorm> vormen`) op het scherm afdrukt. In elk object zit de functie `'Paint'` die weet welke informatie afgedrukt moet worden. Hieronder wordt deze uitgewerkt:

- Alle objecten rapporteren eerst hun coördinaat in de stijl `(x, y)`
- Na het coördinaat volgt `"<spatie><minteken><spatie>"`
- Hierna volgt de kleur gevolgt door `"<spatie>:"`

Vervolgens:

- Driedimensionale objecten rapporteren de tekst `'3d'`.
 - Een kubus rapporteert daarna de tekst `'Kubus met zijden'` gevolgt door de lengte van de ribben tussen haakjes.
- Veelhoeken rapporteren `'Veelhoek met <z> zijden'` gevolgt door de lengte van de zijden tussen haakjes
- Ellipsvormige objecten rapporteren `'Ellipse met straal '` gevolgt door `straal1` en `straal 2` tussen haakjes.

Bijvoorbeeld:

```
(1,1) - blauw :3d:Kubus met zijden (4,5,6)
(4,2) - rood :3d:Kubus met zijden (1,2,3)
(6,3) - geel :3d:Kubus met zijden (3,3,3)
(8,4) - oranje :3d:Kubus met zijden (9,9,9)
(10,10) - rood :Veelhoek met 4 zijden (5,5,5,5)
(20,11) - geel :Veelhoek met 4 zijden (6,6,6,6)
(30,12) - blauw :Veelhoek met 4 zijden (7,7,7,7)
(40,13) - paars :Veelhoek met 4 zijden (8,8,8,8)
(81,19) - blauw :Ellipse met straal (5,5)
(82,19) - groen :Ellipse met straal (6,6)
(83,91) - oranje :Ellipse met straal (7,7)
```

Om een onbekend aantal getallen samen te voegen met een komma als scheidingsteken kun je de functie `String.Join` gebruiken.

Vraag 3 (20 punten)

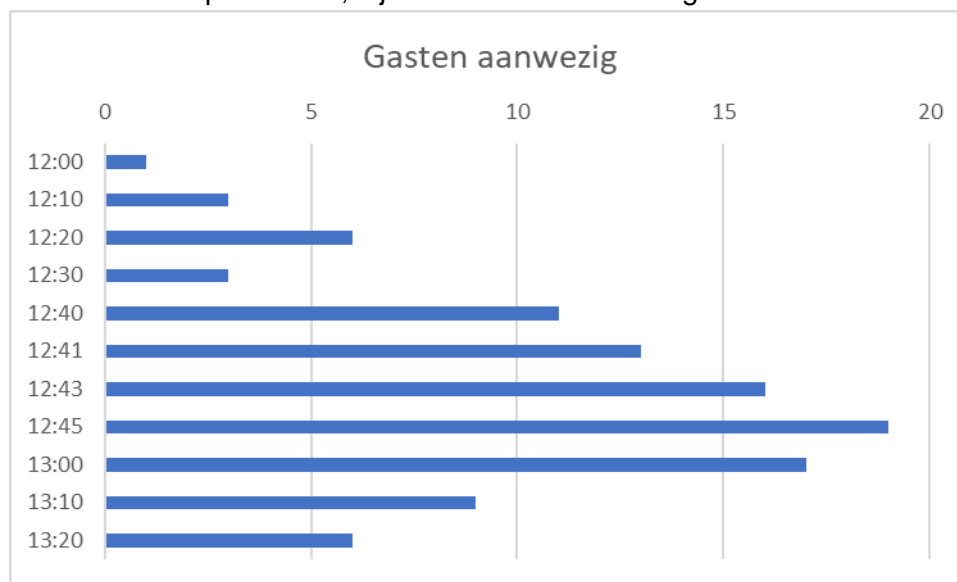
Probleemstelling

Recentelijk is in Leeuwarden een nieuwe pizzeria gestart. In verband met Corona willen zij zicht houden op het totaal aantal gasten dat tegelijk in het restaurant aanwezig is. Bij binnenkomst wordt het tijdstip plus het aantal nieuwe gasten geregistreerd. Als gasten weggaan wordt het tijdstip en het aantal gasten als negatieve waarde genoteerd. Dit kan er als volgt uitzien (het restaurant gaat om 12:00u open):

12:00:00, 1
12:10:00, 2
12:20:00, 3
12:30:00, -3
12:40:00, 8
12:41:00, 2
12:43:00, 3
12:45:00, 3
13:00:00, -2
13:10:00, -8
13:20:00, -3

Op deze manier kan de pizzeria eigenaar goed zicht houden op het verloop. Bij de ingang staat een uitzendkracht die de gasten verwelkomt of uitlaat. Deze uitzendkracht gaat de administratie doen.

Mochten de BOA's komen inspecteren dan kan de uitzendkracht handig het verloop laten zien. Deze methode van noteren maakt het wel wat lastig om te zien hoeveel mensen er nu op een bepaald moment zijn. Daarom heeft de eigenaar een oplossing bedacht. De eigenaar wil dat doen met visuele hulpmiddelen, bijvoorbeeld een staafdiagram zoals hier onder weergegeven.



Deze aantallen kunnen eenvoudig berekend worden door alle getallen uit de lijst op te tellen tot een bepaald moment. Bijvoorbeeld om 12:30u levert dit op (zie voorbeeld tabel hierboven):

$$1 + 2 + 3 - 3 = 3$$

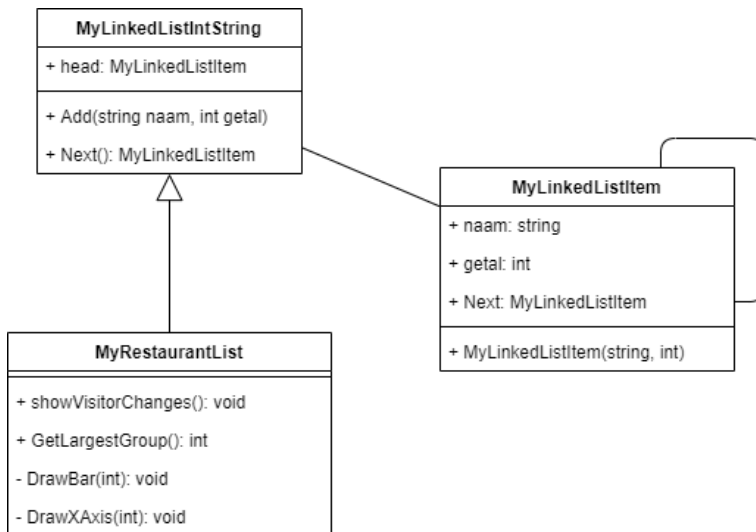
Daarnaast zijn de BOA's benieuwd wat de maximale groepsgrootte is die binnen gelaten is. Dit valt gelukkig eenvoudig af te lezen uit de lijst (in dit geval dus 8 personen).

Technische toelichting

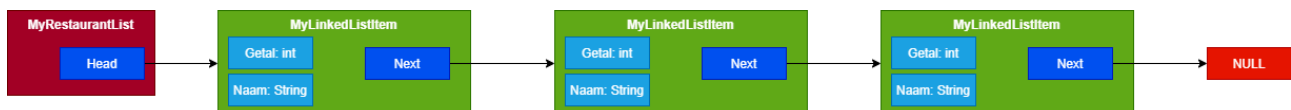
Voor deze vraag wordt een klasse `MyLinkedListIntString` gebruikt die het project **Restaurant** van de Visual Studio Solution zit. De klasse `MyLinkedListIntString` is verborgen maar is wel bruikbaar.

De enkelvoudige gelinkte lijst wordt gebruikt om een lijst van integers & strings op te kunnen slaan. Hiervoor wordt een klasse gebruikt met de naam `MyLinkedListItem`. In onderstaande diagram wordt deze structuur uitgelegd waarbij strings en integers opgeslagen worden.

We gebruiken deze structuur om informatie voor een restaurant op te slaan. De klasse `MyRestaurantList` erft van `MyLinkedListIntString`.



Een lijst van het type `MyLinkedListIntString` bevat een verwijzing naar een **lijst** met gegevens. De start van deze lijst is opgenomen in de variabele `head`. De lijst bestaat uit items van het type `MyLinkedListItem`. In een `MyLinkedListItem` zit een **string**, een **int** en een verwijzing naar het volgende element (`Next`). Als er geen volgend item is, dan heeft `Next` de waarde `NULL`.



Je kunt items toevoegen aan de lijst met de methode `AddItem`. Deze methode heeft twee parameters met de volgende betekenis:

- `Getal` → aantal gasten dat binnenkomt
- `Naam` → Tijdstip dat gasten binnenkomen.

In het bestand `'MyRestaurantList.cs'` zit een klasse met naam `'MyRestaurantList'`. Deze klasse bevat twee publieke methodes en twee private methodes. De publieke methodes zijn:

- `public void showVisitorChanges()`
- `public int GetLargestGroup()`

De twee private methodes mag je gebruiken om het probleem op te lossen waarbij zaken op het scherm getekend moeten worden.

- `private void DrawBar(int nrOfItems)`
- `private void DrawXAxis(int nrOfItems)`

Vraag 3a (8 punten)

Implementeer de methode `public void showVisitorChanges()`. Deze methode maakt een “grafisch” overzicht in de vorm van een soort staaf diagram. Elke gast wordt weergegeven door een '#'-teken. Op elk tijdstip wordt dus het totaal aantal gasten op dat moment weergegeven.

Zorg dat onderstaande output op het scherm komt. De code in 'program.cs' is hier al voor klaargezet.

```
12:00:00|#
12:10:00|###
12:20:00|#####
12:30:00|###
12:40:00|#####
12:41:00|#####
12:43:00|#####
12:45:00|#####
13:00:00|#####
13:10:00|#####
13:20:00|#####
          |-----|-----|
```

Om de #-tekens te tekenen gebruik je de private methode `DrawBar`. Je geeft hier het juiste aantal bezoekers dat op dat moment in het restaurant zit mee als parameter.

Om de X-as te tekenen ('|-----|-----|') gebruik je de private methode `DrawXAxis`. Je geeft hier het **maximaal** aantal gasten mee dat tegelijk in het restaurant aanwezig was. In dit geval moet dat uitkomen op 19 ($1+2+3-3+8+2+3+3 = 19$).

Vraag 3b (12 punten)

Implementeer de methode `public int GetLargestGroup()`. Deze methode bepaalt op een **recursieve** manier de grootste groep die in de lijst genoteerd was. Bij het starten van het programma levert dit onderstaande uitvoer op:

```
De grootste groep was: 8 mensen
```

Vraag 4 (2 x 10 punten)

In een goede programmeertaal is er duidelijk onderscheid tussen variabelen van het type `STRING` en `INTEGER`. Zo ook in C#. Met een integer kun je rekenen en met een string niet. Een string is handig bij het afdrukken op bijv. het scherm.

Toch is het handig dat je deze twee types uit kunt wisselen. Schrijf daarom twee recursieve functies die deze twee types kunnen converteren in de ander:

```
static public string ConvertToString(int number)
```

```
static public int ConvertToInteger(string number)
```

De eerste accepteert dus een integer, en converteert deze naar een string. De tweede accepteert een string en converteert deze naar een integer. Randvoorwaarde is dat we alleen met **positieve** waarden werken.

In het project zijn al veel tests opgenomen die beide functies controleren. Je kunt deze tests uitvoeren via de Test Explorer in Visual Studio. Er wordt gebruik gemaakt van NUnit.

Enkele tips:

- om één karakter (`c`) uit een string om te zetten naar een `int` (value) kun je onderstaande berekening gebruiken:
 - `int value = c - '0';`
- om een waarde (`digitInt`) tussen nul en negen (beide inclusief) om te zetten naar een `char` gebruik je onderstaande functie:
 - `char digitString = (char)('0' + digitInt);`
- maak slim gebruik van delen door 10
- maak slim gebruik van vermenigvuldigen met 10
- gebruik de modulo-operator
- begin '*achteraan*' en werk achterstevoren.

-----EINDE VAN HET TENTAMEN-----

Puntenverdeling en controle

Vraag	Onderdeel	Sub	Punten	Som
1	a	1	5	30
		2	5	
	B		20	
2	a		5	20
	b		5	
	c		10	
3	a		8	20
	b		12	
4	a		20	20
Totaal				90

TOTAAL 90 + 10 start punten = 100