

<b>Opleiding</b>	HBO ICT - Flex	<b>Toetsvorm</b>	Tentamen
<b>Titel Toetseenheid</b>	Programmeren	<b>Datum</b>	-
<b>Titel Toets</b>	Programmeren	<b>Examinator</b>	Martin Molema
<b>Module</b>	Software Engineering 1	<b>Tijdsduur</b>	2,5 uur
	S1 2020/2021	<b>Cijfer</b>	(90pt + 10pt) / 10
<b>Maximale score</b>	90 punten	<b>Pagina's</b>	9

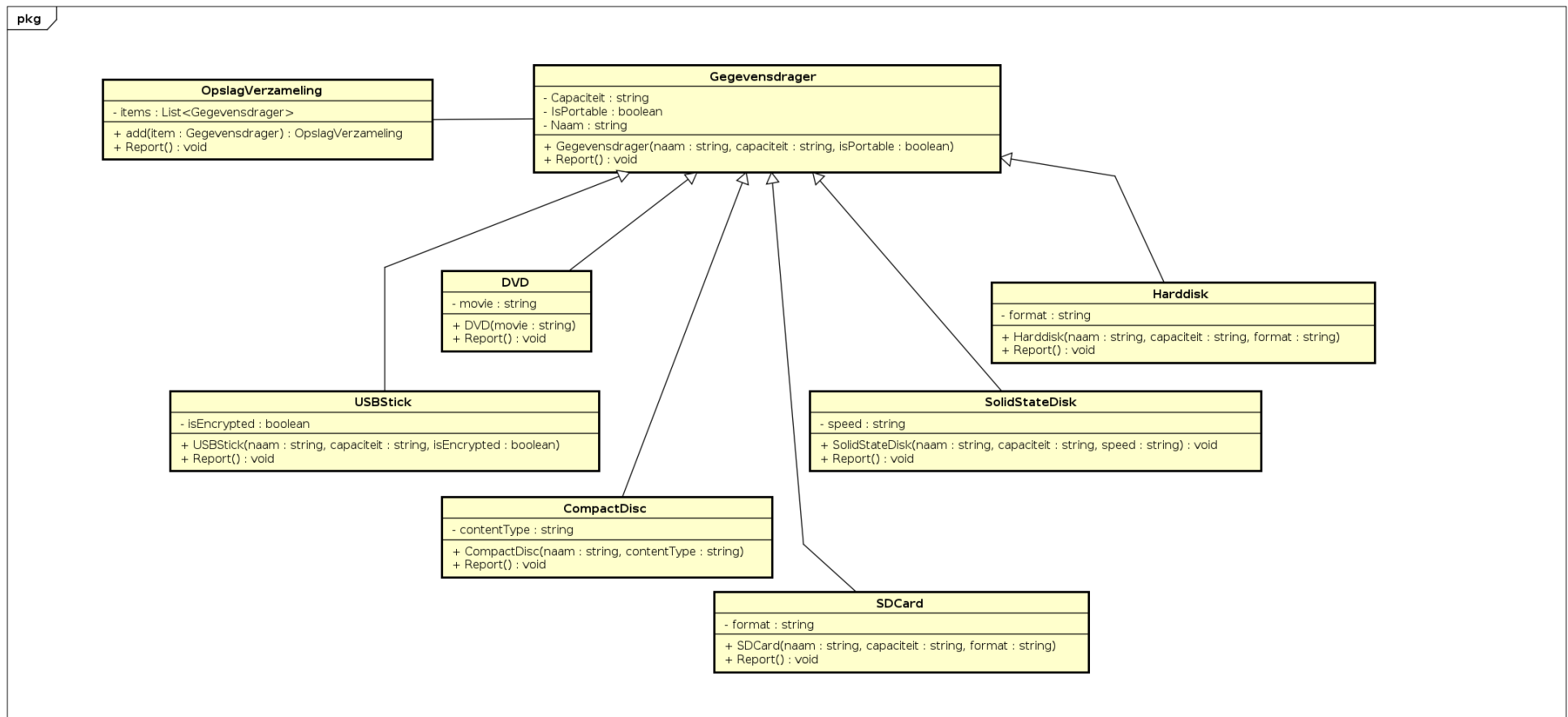
---

<b>Toegestane hulpmiddelen</b>	Internet & Boeken
--------------------------------	-------------------

---

## Vraag 1 (30 punten)

Zie onderstaande klassendiagram



In bovenstaande klassendiagram worden een aantal klassen beschreven.

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

- 1) Bij een aantal klassen staat steeds de methode `'Report ()'`. In de C#-broncode moet een speciaal keyword gebruikt worden bij de declaratie van die methode. Welk keyword moet gebruikt worden in de klasse `Gegevensdrager` en welke in de onderliggende klassen? (5 punten)
- 2) Tussen de verschillende klassen staan pijlen met een open punt. Wat symboliseert zo'n pijl? (5 punten)

## Vraag 1b (20 punten)

Toelichting op het klassendiagram. Het schema is een presentatie van mogelijke gegevensdragers; bijvoorbeeld bij jou in huis. Elke soort gegevensdrager heeft bijzondere eigenschappen. De combinatie van de eigenschappen is meestal uniek. Belangrijke informatie:

- Elke gegevensdrager krijgt een naam. Bijvoorbeeld "Martin's USB Stick".
- Van elke `Gegevensdrager` is minimaal de capaciteit beschikbaar. Deze is uitgedrukt in een getal en een eenheid: "32GB" of "128MB".
- Verder wordt bij een `Gegevensdrager` aangegeven of deze draagbaar is: de boolean `'isPortable'` staat op TRUE indien een gegevensdrager niet ingebouwd is, en dus draagbaar is.
- Op DVD's staan alleen films; er zijn dus geen DVD-roms
- Het format van de `Harddisk` is de omvang; voorbeelden zijn 3,5inch (3,5"), 2,5inch (2,5")
- Het format van de `SDCard` is het type: MicroSD of SD
- `USBSticks` kunnen versleuteld zijn ; de boolean `isEncrypted` staat op TRUE indien de USBStick versleuteld is.
- Bij een SSD (`Solid State Disk`) is de snelheid altijd erg belangrijk. Deze wordt opgenomen in het veld `speed` en bevat de transfer snelheid per seconde. Bijvoorbeeld "500MB/s".

Bouw de basis van de klassen uit het bovenstaande klassendiagram met C# via een *Console Application*. Toelichting

- alle variabelen hebben de beperkte zichtbaarheid **'private'**.

Een aantal klassen hebben vaste eigenschappen. Deze staan hier onder weergegeven. Bouw deze met de juiste taalconstructie in de aangegeven klassen.

- Een CD heeft een vaste capaciteit van "700MB" en is portable
- Een DVD heeft een vaste capaciteit van "4GB" en is portable
- De naam van een DVD staat gelijk aan de naam van de film die er op staat
- De volgende gegevensdragers zijn **NIET** portable
  - `Harddisk`
  - `SolidStateDisk`
- De volgende gegevensdragers zijn **WEL** portable
  - `SDCard`
  - `CompactDisc`
  - `DVD`

- USBStick

Let op het return-type van de methode 'Add' uit de klasse `Opslagverzameling`. Deze methode dient de huidige instantie terug te geven zodat chaining van de methode 'Add' mogelijk is.

De beschrijving van de methode 'Report()' volgt in de volgende vraag (2c).

## Vraag 2 (20 punten)

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

### 2a: Gegevensdragers aanmaken (5 punten)

Nu je het klassendiagram hebt geïmplementeerd ga je gegevensdragers maken. Maak van elke type gegevensdrager minimaal 2 verschillende instanties om een orkest te vormen:

Verzin bij het aanmaken van een instantie van elk instrument zelf een leuke naam voor elk instrument. Bijvoorbeeld "*Martin's USBStick*". Bij de DVD kies je voor de naam van je favoriete films.

### 2b: de gegevensdragers in een verzameling zetten (5 punten)

Maak een instantie van de klasse `Opslagverzameling` en voeg de gegevensdragers toe door middel van de methode 'Add'.

## 2c: Gegevensdragers rapporteren (10 punten)

Implementeer de methode 'Report' in alle klassen. De methode 'Report' in de klasse `Opslagverzameling` zorgt voor het rapporteren van alle gegevensdragers in de lijst met `items`. Dit levert onderstaande output.

```
[USBStick] Kingston = 8GB (portable)
[USBStick] WD = 16GB (portable)
- ! Encrypted
[USBStick] XYZ = 32GB (portable)
[SDCard] Telefoon Martin = 32GB (portable)
- Formaat: microSD
[SDCard] Telefoon Piet = 32GB (portable)
- Formaat: microSD
[SDCard] Telefoon Klaas = 64GB (portable)
- Formaat: full
[Harddisk] sda = 1TB
- 3,5"
[Harddisk] sdb = 1TB
- 2,5"
[SolidStateDisk] sdc = 60GB
- 500MB/s
[SolidStateDisk] sdd = 60GB
- 500MB/s
[DVD] Back to the future = 4GB (portable)
[DVD] Edge of tomorrow = 4GB (portable)
-----
12 items gevonden
```

Uitleg over de output:

- Eerst wordt de naam van het soort Gegevensdrager getoond tussen blokhaken '[...]'. Gebruik hiervoor de functie `'GetType().Name'` op een instantie van een klasse.
- Daarna wordt de naam afgedrukt gevolgd door een '='-teken
- Daarna volgt de opslagcapaciteit
- Als een gegevensdrager portable is, dan wordt dit tussen haakjes afgedrukt.
- Eventueel wordt extra informatie op een volgende regel afgedrukt:
  - Voor SDCard en Harddisk volgt het formaat
  - Voor een USB Stick volgt een indicatie dat deze encrypted is.
  - Voor een SolidStateDisk wordt de snelheid weergegeven.
- Nadat alle items zijn afgedrukt volgt een scheidingslijn
- Als laatste wordt het aantal items afgedrukt.

Gebruik hiervoor op een slimme manier de methodes `'Report()'` uit de klasse `Gegevensdrager` en de onderliggende klassen. Nogmaals: alle klasse variabelen hebben de zichtbaarheid **private**.

## Vraag 3 (20 punten)

Voor deze vraag wordt een klasse 'MyLinkedListString' gebruikt die in de map 'LinkedList' van Visual Studio zit. De klasse 'MyLinkedListString' is verborgen maar is wel bruikbaar.

De eenvoudige gelinkte lijst wordt gebruikt om een lijst van integers, strings of andere typen te kunnen opslaan. Hiervoor wordt een klasse gebruikt met de naam 'MyLinkedListStringItem'. In onderstaande diagram wordt deze structuur uitgelegd waarbij strings opgeslagen worden.



Een lijst van het type 'MyLinkedListString' 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 'MyLinkedListStringItem'. In een item van dat type zit een **string** ('Data') 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'.

In het bestand 'StudentLinkedList.cs' zit een klasse met naam 'StudentLinkedListInt'. Deze klasse bevat twee methodes:

- `public int OnevenSom()`
- `public string Omdraaien()`

Om de opgaven 3a en 3b te testen moet gebruik worden gemaakt van het Nunit Testframework dat in de solution is opgenomen. Hier staan afdoende testcases in opgenomen om je functies te testen.

### Vraag 3a (10 punten)

Implementeer de methode 'public int OnevenSom' zodat deze alle **oneven** getallen in de lijst optelt. Hierbij dient gebruik **alleen** gemaakt te worden van recursie. Het resultaat van deze optelsom wordt teruggegeven als resultaat van de functie.

### Vraag 3b (10 punten)

Implementeer de methode 'omdraaien' zodat deze de lijst met getallen omgekeerd teruggeeft in de vorm van een string. Een voorbeeld:

1,2,3,4,5 → "54321"  
-1,2,-3 → "-32-1"

Zie de lijst met testcases voor meer voorbeelden.

## Vraag 4 (20 punten)

De antwoorden in deze vraag kun je implementeren in de solution binnen het project 'MathProblems'. In dit project zit één C# bestand genaamd 'program.cs' waarin diverse functies zitten die aangevuld moeten worden.

De functies kunnen getest worden met de Nunit testrunner.

### Probleemstelling

De **grootste gemene deler**, afgekort tot ggd, van een aantal gehele getallen (waarvan er ten minste een ongelijk is aan 0) is het grootste positieve gehele getal, waar al deze gehele getallen door gedeeld kunnen worden zonder dat er een rest overblijft.

De grootste gemene deler van de getallen 8 en 12 is bijvoorbeeld 4. 8 en 12 zijn beide deelbaar door de getallen 1, 2 en 4. De grootste van de set {1, 2, 4} is het getal 4, dus de  $GGD(8, 12) = 4$ . Ander voorbeeld: De delers van 7 en 28 zijn {1, 7} dus  $GGD(7, 28) = 7$ .

Het **kleinste gemene veelvoud** (d.w.z.: 'kleinste gemeenschappelijke veelvoud') van twee verschillende gehele getallen, is het kleinste positieve gehele getal dat een veelvoud is van beide getallen, dus het kleinste positieve gehele getal, waarvan beide getallen deler zijn. Het kleinste gemene veelvoud van bijvoorbeeld 3 en 5 is 15. Het kleinste gemene veelvoud van 15 en 27 is gelijk aan 135: 135 is deelbaar door zowel 15 als 27. Het getal 270 is ook een veelvoud van 27 en 135 (270 delen door 27 = 10, 270 delen 15 = 18). Echter 135 is kleiner. Een kleiner getal dan 135 is niet te vinden, en daarom is 135 de KGV.

Dit is te berekenen door de absolute waarde van het product ( $15 \times 27$ ) te delen door de grootste gemene deler  $GGD(15, 27)$ .

Formule voor *Kleinste Gemene Veelvoud*:

$$KGV(a, b) = \frac{|a * b|}{GGD(a, b)}$$

Bijvoorbeeld:

$$KGV(15, 27) = \frac{|15 * 27|}{GGD(15, 27)} = \frac{405}{3} = 135$$

Algoritme voor het bepalen van de **Grootste Gemene Deler**:

$GGD(a, b)$ :

```
Als (a == b) → a
Als (a > b) → GGD(a-b, b)
Als (a < b) → GGD(a, b-a)
```

Een voorbeeld van dit algoritme voor 15 en 27:

$GGD(15, 27)$ :

- $15 < 27$  →  $GGD(15, 27-15)$
- $15 > 12$  →  $GGD(15-12, 12)$
- $3 < 12$  →  $GGD(3, 12-3)$
- $3 < 9$  →  $GGD(3, 9-3)$
- $3 < 6$  →  $GGD(3, 6-3)$

- $3 == 3 \rightarrow 3$

Uitkomst  $GGD(15, 27) = 3$

## Vraag 4a (10 punten)

Implementeer de functie `kgv(int a, int b)` die het **Kleinste Gemeenschappelijke Veelvoud** van de factoren A en B berekent. Maak gebruik van **recursie** voor de functie van de **GGD**.

## Vraag 4b (10 punten)

We willen nu de som berekenen van een aantal  $kgv(a, b)$  functies. Deze som wordt als volgt gedefinieerd:

```
int somkgv(int n)
```

waarbij de som berekend wordt van alle combinaties  $(x, n)$  waarbij  $x$  loopt van  $1$  tot  $n$ . In wiskundige termen:

$$somkgv(n) = \sum_{x=1}^n kgv(x, n)$$

Oftewel:

$$somkgv(5) = kgv(1, 5) + kgv(2, 5) + kgv(3, 5) + kgv(4, 5) + kgv(5, 5) = 5 + 10 + 15 + 20 + 5 = 55$$

Implementeer de functie `somkgv(int n)`. Maak gebruik van recursie; het gebruik van lus-constructies als `FOR`, `DO-WHILE`, `FOREACH` of `WHILE` is niet toegestaan.

-----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		10	20
	b		10	
4	a		10	20
	b		10	
<b>Totaal</b>				<b>90</b>

TOTAAL 90 + 10 start punten = 100