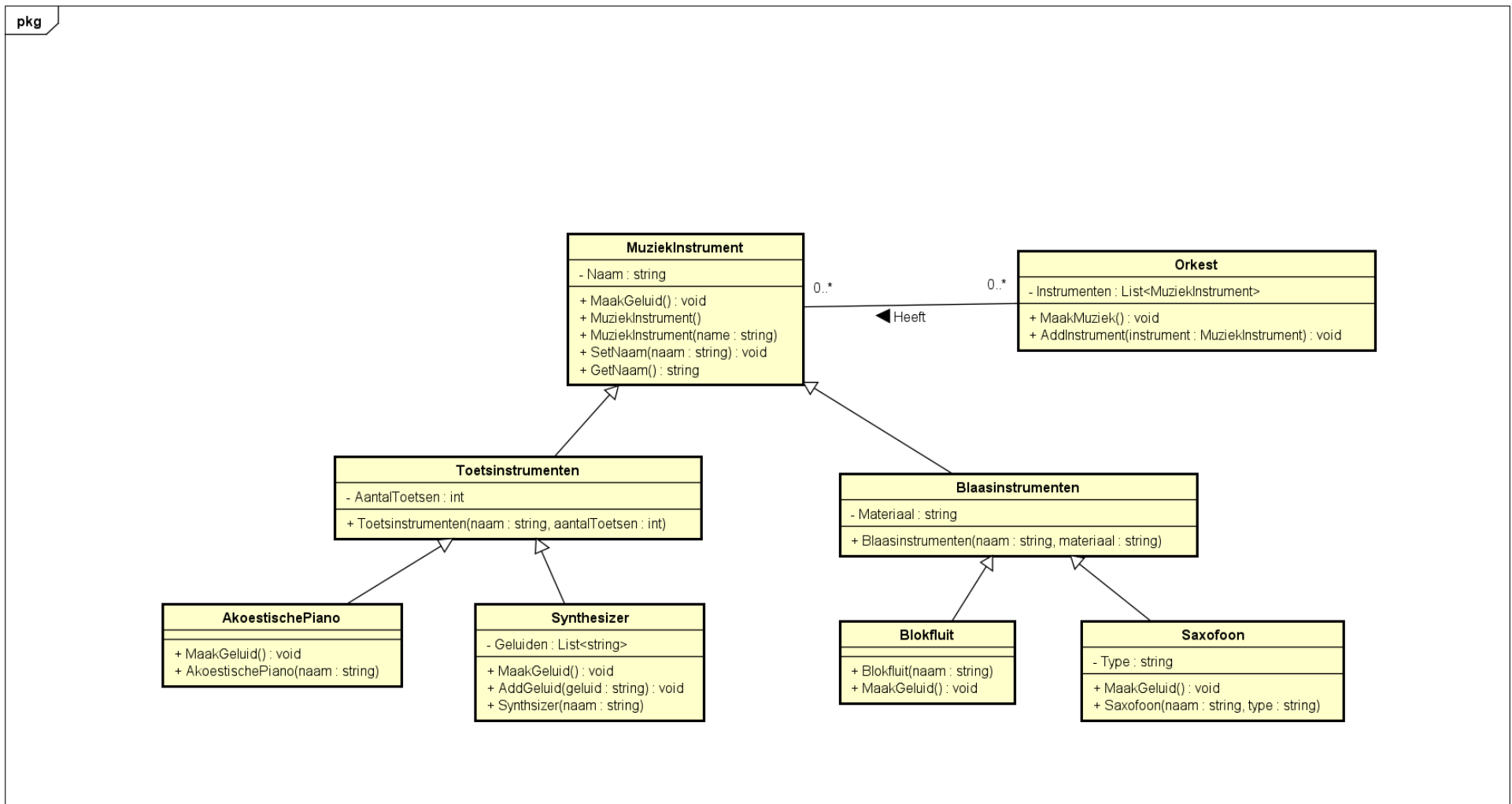


Opleiding	HBO ICT - Flex	Toetsvorm	Tentamen
Titel Toetseenheid	Programmeren	Datum	-
Titel Toets	Programmeren	Examinator	Martin Molema
Module	Software Engineering 1	Tijdsduur	2,5 uur
	S2 2019/2020	Cijfer	(90pt + 10pt) / 10
Maximale score	90 punten	Pagina's	8

Toegestane hulpmiddelen	Internet & Boeken
--------------------------------	-------------------

Vraag 1

Zie onderstaande klassendiagram



In bovenstaande klassendiagram worden een aantal klassen beschreven.

Vraag 1a (theorie) (5+5 = 10 punten)

- 1) Bij de klasse Muziekinstrumenten staan twee methodes met dezelfde naam. Hoe heet zo'n methode en wat is het doel? **(5 punten)**
- 2) Tussen de verschillende klassen staan pijlen met een open punt. Wat symboliseert zo'n pijl? **(5 punten)**

Vraag 1b (20 punten)

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

- alle variabelen hebben de beperkte zichtbaarheid **'private'**.
- In de klasse **"Synthesizer"** staat een lijst met strings. Dit zijn de geluiden die de synthesizer kan maken.

Je krijgt de volgende extra informatie om alle functies te implementeren:

- Klasse `Orkest`
 - Instrumenten kunnen toegevoegd worden aan de lijst ('Instrumenten') met de methode `AddInstrument`.
- Klasse `MuziekInstrument`
 - Met de methode `SetNaam` kan de naam van een instrument aangepast worden
- Klasse `Synthesizer`
 - Met de methode `AddGeluid` kunnen diverse geluiden worden toegevoegd aan de lijst met geluiden ('Geluiden').

Let op: De functie `'MaakGeluid()'` wordt uitgelegd in vraag 2c.

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

- `AkoestischePiano`
 - Naam: **"Grand Piano"**
 - `AantalToetsen`: **88**
- `Synthesizer`
 - Naam: **"Synthesizer"**
 - `AantalToetsen`: **64**
- `Blokfluit`
 - `Materiaal`: **"Hout"**
- `Saxofoon`
 - `Materiaal`: **"Koper"**

Vraag 2 (20 punten)

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

2a: Instrumenten aanmaken (5 punten)

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

- 1 akoestische piano
- 1 synthesizer
- 2 alt saxofoon (type = 'alt')
- 2 tenor saxofoons (type = 'tenor')

De synthesizer vervult diverse instrumenten aan. Voeg de volgende instrumenten toe aan de synthesizer om het orkest compleet te maken:

- Drumstel
- Gitaar
- Contrabas

Verzin bij het aanmaken van een instantie van elk instrument zelf een leuke naam voor elk instrument. Bijvoorbeeld "*Martin's piano*".

2b: het orkest aanmaken (5 punten)

Maak een orkest van de eerder gemaakte instrumenten door deze aan het orkest toe te voegen.

2c: Muziek maken

Nu het orkest compleet is gaan we muziek maken. Dit betekent dat we elk instrument geluid moeten laten maken. Dat doen we door de lijst met instrumenten bij langs te lopen en het bijbehorende geluid te laten maken.

De instrumenten maken het volgende geluid:

- Akoestische Piano: "Pling pling"
- Blokfluit: "tuut"
- Saxofoon: "toet" + een indicatie van het type saxofoon. Dus de alt saxofoon maakt het geluid "toet alt".

De synthesizer heeft een speciale rol: als gevraagd wordt om geluid te maken, worden één voor één alle beschikbare geluiden gemaakt.

Alle geluiden moet op het scherm afgedrukt worden. Op elke regel staat één instrument, gevolgd (op dezelfde regel) door het gemaakte geluid. Bijvoorbeeld:

```
Martin's piano: pling pling
```

Vraag 3 (16 punten)

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

De enkelvoudige 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 getal ('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'.

De koeriersdienst

We gaan deze lijst gebruiken om een koeriersdienst op te zetten. Deze koeriers verzamelen de hele dag orders. De pakketjes van deze orders zetten ze netjes in deze lijst. Als het tijd is om te gaan rijden maken ze een uitdraai van alle pakketjes in de lijst zodat ze weten wat er aan boord is.

Dit gaat als volgt. In de solution 'Koeriersdienst' vind je twee bronbestanden.

Vraag 3a

Maak een lijst met 10 pakketjes. Gebruik hiervoor de klasse 'StudentLinkedListString' en de methode 'AddItem'. Bijvoorbeeld

```
AddItem("Schoenen");
```

Vraag 3b

In de klasse 'StudentLinkedListString' zit een methode 'PakketjesLijst' om de lijst met pakketjes af te drukken. Bouw deze functie met gebruik making van recursie. Deze functie drukt de inhoud van de lijst af op de console. Hieronder de gewenste uitvoer

```
Pakketten
-----
Strijkijzer
Ghettoblaster
RaspberryPi
Smartphone Galaxy S10
Shampoo
WC Papier
-----
6 Pakketten
-----
```

Vraag 3c

Het vervelende van een koeriersdienst is dat je in omgekeerde volgorde moet werken. Dus op basis van de lijst uit vraag 3b is de auto van de koerier ingeladen. Het eerste pakketje ligt dus het verste weg in de auto, en moet dus als laatste bezorgd worden.

Maak de functie 'RouteLijst' af zodat deze de lijst in omgekeerde volgorde afdruckt op het scherm. Daarbij moet gebruik worden gemaakt van een nummering. Zie onderstaande uitvoer, gebaseerd op de lijst uit de vorige vraag (3b).

```
Routelijst
6 - WC Papier
5 - Shampoo
4 - Smartphone Galaxy S10
3 - RaspberryPi
2 - Ghettoblaster
1 - Strijkijzer
```

Het pakket dat als laatste is toegevoegd staat dus bovenaan deze lijst, voorafgegaan door het nummer uit de lijst. Het pakket dat als eerste is toegevoegd aan de lijst krijgt nummer 1.

Het pakket dat als laatste is toegevoegd krijgt het hoogste nummer.

Vraag 4 (19 punten)

Gegeven een getal n met een waarde van 1 tot 100 (inclusief). Deel dit getal door twee net zo lang tot je uitkomt bij het getal 1. Als n oneven is moet n eerst met 1 verhoogd worden.

Opdracht Schrijf een functie die telt hoe vaak je een oneven getal tegenkomt in dit algoritme. Maak de functie 'CalcSteps' af. Deze functie dient recursief te zijn.

Je kunt de functie 'CalcSteps' testen via het NUnit testframework (reeds in de solution opgenomen). De functie 'TestCalculations' voert 100 tests uit.

De uitvoer is als volgt:

```
n = resultaat
```

bijvoorbeeld:

```
1 = 1
2 = 1
3 = 2
4 = 1
5 = 3
6 = 2
7 = 2
8 = 1
9 = 4
10 = 3
11 = 3
12 = 2
13 = 3
```

Enkele voorbeelden van het algoritme en de tussenstappen staan in onderstaande lijst. De uitkomst van de functie staat tussen blokhaken. Dus als $x=11$, dan is de uitkomst 3. De **tussenresultaten** worden als volgt berekend:

$n = 11 : 11 + 1 = 12 \rightarrow 12/2 = 6 \rightarrow 6/2 = 3 \rightarrow 3 + 1 = 4 \rightarrow 4/2 = 2 \rightarrow 2/2 = 1 \rightarrow 1 : [3]$

```

1 => [1]
2 => 1 => [1]
3 => 4 => 2 => 1 => [2]
4 => 2 => 1 => [1]
5 => 6 => 3 => 4 => 2 => 1 => [3]
6 => 3 => 4 => 2 => 1 => [2]
7 => 8 => 4 => 2 => 1 => [2]
8 => 4 => 2 => 1 => [1]
9 => 10 => 5 => 6 => 3 => 4 => 2 => 1 => [4]
10 => 5 => 6 => 3 => 4 => 2 => 1 => [3]
11 => 12 => 6 => 3 => 4 => 2 => 1 => [3]
12 => 6 => 3 => 4 => 2 => 1 => [2]
13 => 14 => 7 => 8 => 4 => 2 => 1 => [3]
14 => 7 => 8 => 4 => 2 => 1 => [2]
15 => 16 => 8 => 4 => 2 => 1 => [2]
16 => 8 => 4 => 2 => 1 => [1]

```

Let op:

- a) als je met een oneven getal start, dan telt dat mee
- b) als je de functie met de waarde 1 aanroept is het resultaat dus ook 1

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

Puntenverdeling en controle

Vraag	Onderdeel	Sub	Punten	Som
1	a	1	5	35
		2	5	
	b	1..5	25	
2	a	5		20
	b	5		
	c	10		
3		1	8	16
		2	8	
4			19	19
Totaal				90

TOTAAL 90 + 10 start punten = 100