HO
GENT

Delegates en Lambdas - Oefeningen

Table of Contents

1. Straatnamen

1.1. Gegeven

Download “StraatnamenInfo.zip” van Chamilo, dit is een zipbestand met daarin de volgende tekstbestanden:

- Straatnamen.csv
- Gemeentenaam.csv
- StraatnaamID_gemeenteID.csv
- ProvincieInfo.csv
- ProvincieIDsVlaanderen.csv

1.1.1. Straatnamen.csv

Dit bestand bevat een lijst met straatnamen in Vlaanderen, Brussel en een deel van Wallonië en een unieke id.

```
EXN;LOS
-9;NULL
1;Acacialaan
2;Adriaan Sanderslei
3;Ahornelaan
4;Antoon van Brabantstraat
5;Antwerpsesteenweg
6;Arkel
7;August Vermeylenlaan
8;Azalealaan
9;Baron van Ertbornstraat
```

1.1.2. Gemeentenaam.csv

Dit bestand bevat de gemeentenamen en een gemeenteId. **Het veld gemeenteNaamId gebruiken we niet.**

Het veld `taalCodeGemeenteNaam` gebruiken we om te filteren want we gaan enkel de Nederlandse naam gebruiken (code : nl).

```
gemeenteNaamId;gemeenteId;taalCodeGemeenteNaam;gemeenteNaam
1;1;nl;Aartselaar
2;2;nl;Antwerpen
3;2;fr;Anvers
4;2;de;Antwerpen
5;3;nl;Boechout
6;4;nl;Boom
```

```
7;5;nl;Borsbeek
```

1.1.3. StraatnaamID_gemeenteID.csv

In dit bestand is de link gelegd tussen de straatnaamId en de gemeenteId.

Om uit te zoeken tot welke gemeente een straatnaam behoort kan je deze lijst gebruiken.

```
straatNaamId;gemeenteId
1;1
2;1
3;1
4;1
5;1
6;1
7;1
```

1.1.4. ProvincieInfo.csv

Informatie over de provincie waartoe een gemeente behoort is in dit bestand opgeslagen. Je vindt er voor elke gemeente (gemeenteId) tot welke provincie deze behoort (provincieId en ProvincieNaam). Ook hier gaan we enkel de Nederlandse namen van de provincies gebruiken, dus filteren op code ‘nl’.

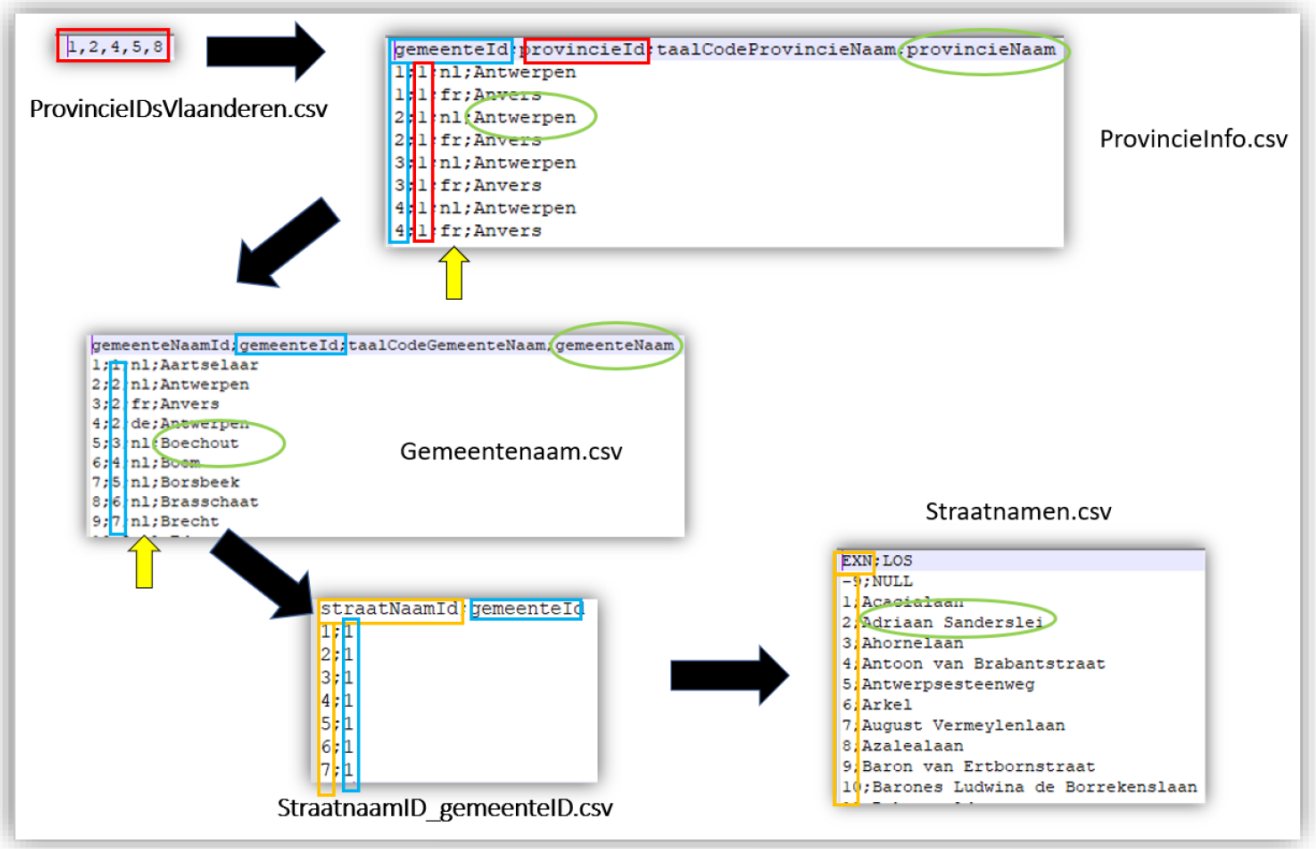
```
gemeenteId;provincieId;taalCodeProvincieNaam;provincieNaam
1;1;nl;Antwerpen
1;1;fr;Anvers
2;1;nl;Antwerpen
2;1;fr;Anvers
3;1;nl;Antwerpen
3;1;fr;Anvers
4;1;nl;Antwerpen
4;1;fr;Anvers
5;1;nl;Antwerpen
```

1.1.5. ProvincieIDsVlaanderen.csv

De provincies die we verwerken kan je terugvinden in dit bestand, het bevat een lijstje met de provincieIds.

```
1,2,4,5,8
```

1.2. Overzicht



1.3. Opgave

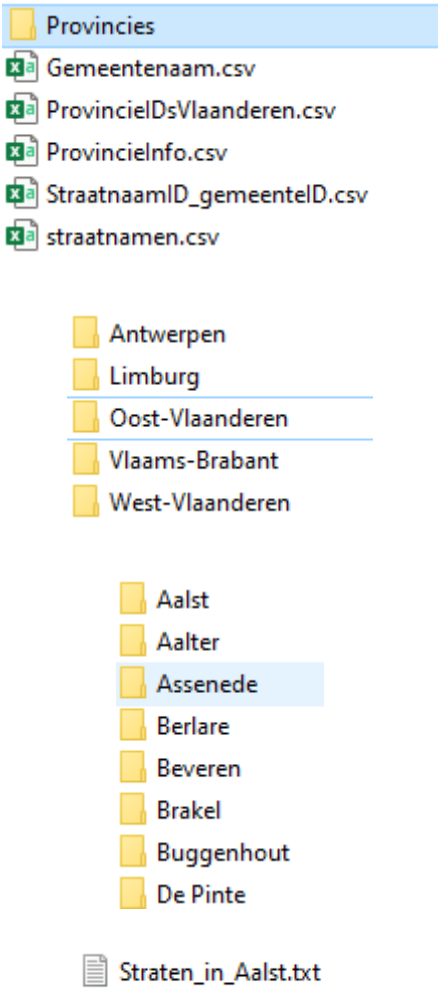
Dit is een oefening op LINQ, een 3-lagen applicatie is niet perse nodig.

Schrijf een methode die als parameters meekrijgt:

- De locatie van de bronbestanden.
- De naam van een subfolder waar de resultaten worden weggeschreven. Deze subfolder komt naast de bronbestanden te staan.

Verwachte functionaliteit:

1. Het programma moet de data uit de bronbestanden verwerken en de volgende uitvoer bekomen:
   a. Voor elke provincie wordt een folder aangemaakt met daarin subfolders voor elke gemeente.
   b. In de folders voor elke gemeente staat een tekstbestand (`Straten_in_xxxx.txt`) met de straatnamen van de gemeente in alfabetische volgorde.




```
't Klein Eeckhout
't Maegelijnplein
't Spieken
't Vestjen
1 Meistraat
Aartstraat
Abbeelstraat
Abdijstraat
Abrahamsweg
Acaciastraat
Achterbremt
Achtermaal
Achterstraat
Achterweg
Achtzaligheden
Affligemdreef
Afspanningsstraat
Ajuinenstraat
```

1.4. Stappenplan

1.4.1. Stap 1

Defineer klassen voor: `Straatnaam`, `Gemeente`, `StraatnaamIdToGemeenteId` en `GemeenteProvincieInfo`.

Lees de bronbestanden in en verwerk de data naar objecten die je bijhoudt in collecties.

Vergeet niet rekening te houden met de te verwerken provincie ids.

1.4.2. Stap 2 (Dit is de moeilijkste stap)

Het is de bedoeling om na het inlezen van de bronbestanden een datastructuur op te stellen die voor ons de nodige info beschikbaar maakt.

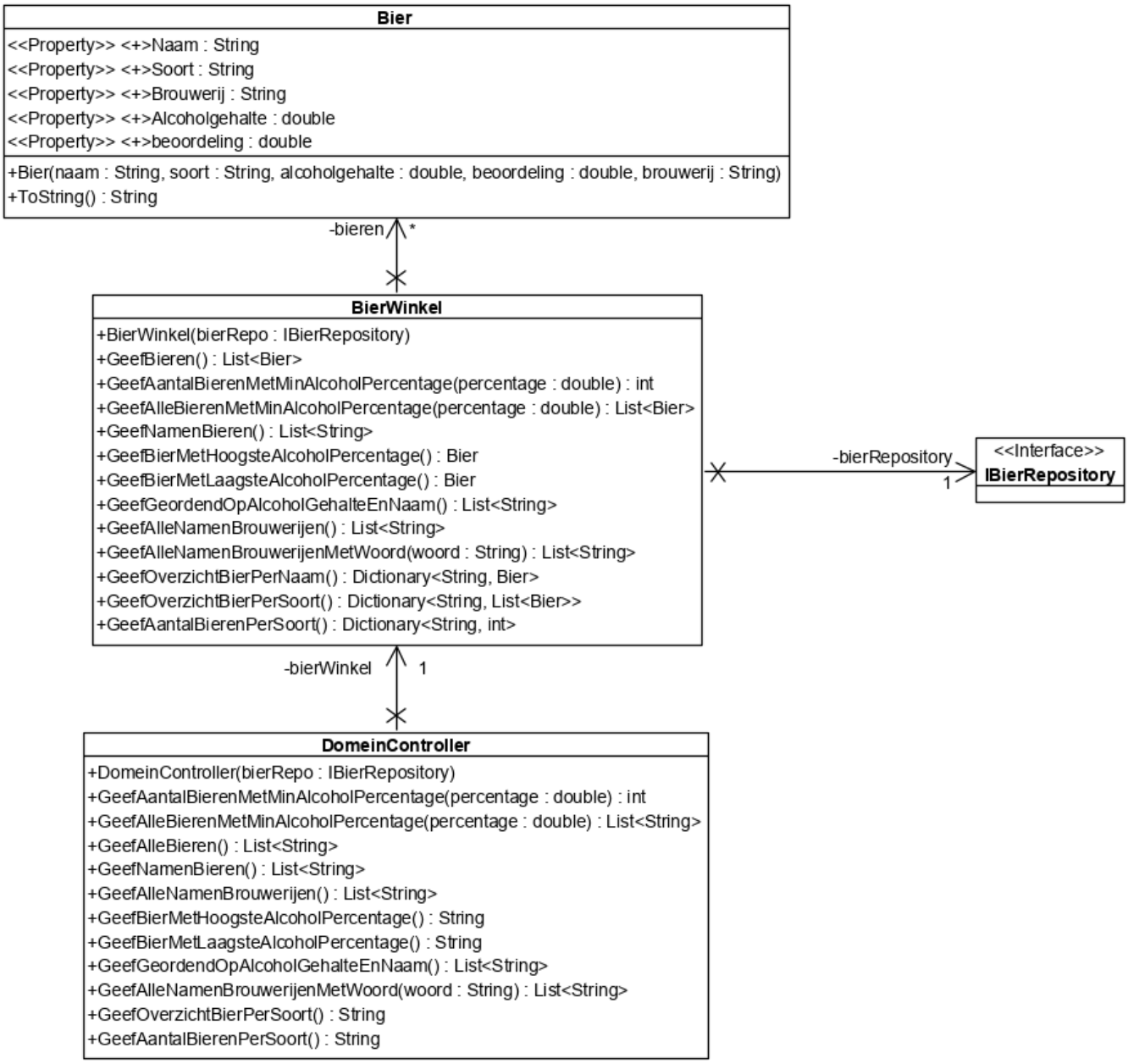
Denk eerst even na over deze datastructuur:

- Gemeenten horen toe aan provincies: hoe kan je gemeenten linken aan een provincie in een collectie.
- Straatnamen horen toe aan gemeenten: hoe kan je straatnamen linken aan een gemeente in een collectie.
- Hoe kom je nu tot een samengestelde collectie die provincies linkt aan gemeenten en die gemeenten linkt aan straatnamen.

1.4.3. Stap 3

Uiteindelijk kom je tot een samengestelde collectie die straatnamen linkt aan gemeenten en die gemeenten linkt aan provincies. Door deze collectie te overlopen kan de gevraagde folderstructuur en tekstbestanden eenvoudig aangemaakt worden.

2. Oefening Bier



Opdracht: Vul de ontbrekende code aan zodat de applicatie correct kan worden uitgevoerd.

Respecteer bovenstaande uml.

Vervolledig de implementatie van de methodes in de klassen BierWinkel en DomeinController.

2.1. Uitvoer Stap 1

```
=====================================================
Bereken het aantal bieren die minstens 8 graden hebben
Aantal bieren van minstens 8 graden:7
```

2.2. Uitvoer Stap 2

```
=====================================================
```

```
Maak een lijst met alle bieren van minstens 8 graden
naam = Tripel_Kanunnik, soort = tripel, brouwerij = Wilderen, alcoholgehalte = 8,20,
beoordeling = 9,1
naam = Black_Albert, soort = tripel, brouwerij = De Struise Brouwers, alcoholgehalte =
13,00, beoordeling = 9,0
naam = Rochefort_10, soort = donker, brouwerij = Brasserie de l'Abbaye de Saint-Rémy,
alcoholgehalte = 11,00, beoordeling = 9,1
naam = Alpaïde, soort = donker, brouwerij = Nieuwhuys Hoegaarden, alcoholgehalte = 9
,50, beoordeling = 9,0
naam = Moinette_Blonde, soort = blond, brouwerij = Dupont, alcoholgehalte = 8,50,
beoordeling = 8,5
naam = Tripel_Karmeliet, soort = tripel, brouwerij = Bosteels, alcoholgehalte = 8,40,
beoordeling = 8,3
naam = Westmalle_Tripel, soort = tripel, brouwerij = Abdij der trappisten van
Westmalle, alcoholgehalte = 9,50, beoordeling = 8,2
```

2.3. Uitvoer Stap 3

```
=======================================================
Enkel namen van bieren laten zien
WestVleteren_Blond
Tripel_Kanunnik
Black_Albert
Rochefort_10
Alpaïde
Cantillon_Geuze
Moinette_Blonde
Wilderen_Goud
Tripel_Karmeliet
Westmalle_Tripel
```

2.4. Uitvoer Stap 4

```
=======================================================
Bier met hoogste aantal graden
Bier met hoogste aantal graden: naam = Black_Albert, soort = tripel, brouwerij = De
Struise Brouwers, alcoholgehalte = 13,00, beoordeling = 9,0
```

2.5. Uitvoer Stap 5

```
=======================================================
Bier met laagst aantal graden
Bier met laagste aantal graden: naam = WestVleteren_Blond, soort = blond, brouwerij =
Sint-Sixtusabdij van Westvleteren, alcoholgehalte = 5,00, beoordeling = 9,3
```

2.6. Uitvoer Stap 6

```
=====================================================
resultaat op alcoholgehalte van hoog naar laag, dan op naam
naam = Black_Albert, soort = tripel, brouwerij = De Struise Brouwers, alcoholgehalte =
13,00, beoordeling = 9,0
naam = Rochefort_10, soort = donker, brouwerij = Brasserie de l'Abbaye de Saint-Rémy,
alcoholgehalte = 11,00, beoordeling = 9,1
naam = Alpaïde, soort = donker, brouwerij = Nieuwhuys Hoegaarden, alcoholgehalte = 9
,50, beoordeling = 9,0
naam = Westmalle_Tripel, soort = tripel, brouwerij = Abdij der trappisten van
Westmalle, alcoholgehalte = 9,50, beoordeling = 8,2
naam = Moinette_Blonde, soort = blond, brouwerij = Dupont, alcoholgehalte = 8,50,
beoordeling = 8,5
naam = Tripel_Karmeliet, soort = tripel, brouwerij = Bosteels, alcoholgehalte = 8,40,
beoordeling = 8,3
naam = Tripel_Kanunnik, soort = tripel, brouwerij = Wilderen, alcoholgehalte = 8,20,
beoordeling = 9,1
naam = Wilderen_Goud, soort = blond, brouwerij = Wilderen, alcoholgehalte = 6,00,
beoordeling = 8,4
naam = Cantillon_Geuze, soort = geuze, brouwerij = Cantillon, alcoholgehalte = 5,00,
beoordeling = 8,5
naam = WestVleteren_Blond, soort = blond, brouwerij = Sint-Sixtusabdij van
Westvleteren, alcoholgehalte = 5,00, beoordeling = 9,3
```

2.7. Uitvoer Stap 7

```
=====================================================
Alle brouwerijen
Sint-Sixtusabdij van Westvleteren
Wilderen
De Struise Brouwers
Brasserie de l'Abbaye de Saint-Rémy
Nieuwhuys Hoegaarden
Cantillon
Dupont
Bosteels
Abdij der trappisten van Westmalle
```

2.8. Uitvoer Stap 8

```
=====================================================
Alle brouwerijen die het woord "van" bevatten
Sint-Sixtusabdij van Westvleteren
Abdij der trappisten van Westmalle
```

2.9. Uitvoer Stap 9

```
=====================================================
Alle bieren per soort
blond = [naam = WestVleteren_Blond, soort = blond, brouwerij = Sint-Sixtusabdij van
Westvleteren, alcoholgehalte = 5,00, beoordeling = 9,3; naam = Moinette_Blonde, soort
= blond, brouwerij = Dupont, alcoholgehalte = 8,50, beoordeling = 8,5; naam =
Wilderen_Goud, soort = blond, brouwerij = Wilderen, alcoholgehalte = 6,00, beoordeling
= 8,4]
tripel = [naam = Tripel_Kanunnik, soort = tripel, brouwerij = Wilderen, alcoholgehalte
= 8,20, beoordeling = 9,1; naam = Black_Albert, soort = tripel, brouwerij = De Struise
Brouwers, alcoholgehalte = 13,00, beoordeling = 9,0; naam = Tripel_Karmeliet, soort =
tripel, brouwerij = Bosteels, alcoholgehalte = 8,40, beoordeling = 8,3; naam =
Westmalle_Tripel, soort = tripel, brouwerij = Abdij der trappisten van Westmalle,
alcoholgehalte = 9,50, beoordeling = 8,2]
donker = [naam = Rochefort_10, soort = donker, brouwerij = Brasserie de l'Abbaye de
Saint-Rémy, alcoholgehalte = 11,00, beoordeling = 9,1; naam = Alpaïde, soort = donker,
brouwerij = Nieuwhuys Hoegaarden, alcoholgehalte = 9,50, beoordeling = 9,0]
geuze = [naam = Cantillon_Geuze, soort = geuze, brouwerij = Cantillon, alcoholgehalte
= 5,00, beoordeling = 8,5]
```

2.10. Uitvoer Stap 10

```
=====================================================
Aantal bieren per soort
blond = 3
tripel = 4
donker = 2
geuze = 1
```

3. Adressenlijst

Gegeven is een bestand met info over adressen (terug te vinden op Chamilo AdressenInfo.zip).

```
West-Vlaanderen,Kortrijk,Watermolenpad
West-Vlaanderen,Kortrijk,Watermolenstraat
West-Vlaanderen,Kortrijk,Watermolenwal
West-Vlaanderen,Kortrijk,Waterpoort
West-Vlaanderen,Kortrijk,Waterputweg
West-Vlaanderen,Kortrijk,Watertorenhoek
West-Vlaanderen,Kortrijk,Watertorenpad
West-Vlaanderen,Kortrijk,Watervalstraat
West-Vlaanderen,Kortrijk,Waterven
West-Vlaanderen,Kortrijk,Wedeplein
West-Vlaanderen,Kortrijk,Weggevoerdenlaan
West-Vlaanderen,Kortrijk,Weimeerslaan
```

```
West-Vlaanderen,Kortrijk,Weimeerspad
West-Vlaanderen,Kortrijk,Weisterstraat
```

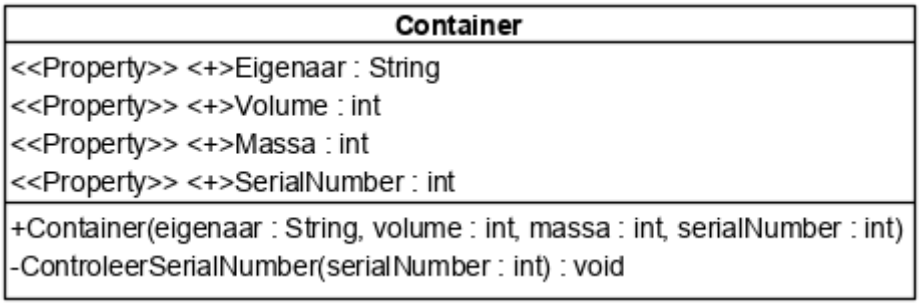
Gevraagde functionaliteit:

- Geef lijst met de provincienamen, alfabetisch gesorteerd.
- Geef lijst van straatnamen voor opgegeven gemeente.
- Selecteer de straatnaam die het meest keren voorkomt en druk voor elk voorkomen de provincienaam, gemeentenaam en straatnaam af. Sortering op basis van provincie en gemeente.
- Voorzie een analoge functie die de meest voorkomende straatnamen weergeeft met een parameter die aangeeft hoeveel straatnamen. Output analoog aan voorgaande functie.
- Voorzie een functie die voor 2 opgegeven gemeenten de gemeenschappelijke lijst van straatnamen weergeeft.
- Voorzie een functie die de straatnamen weergeeft die enkel voorkomen in de opgegeven gemeente, maar die niet voorkomen in een lijst van andere gemeenten.
- Maak een functie die de gemeente weergeeft met het hoogste aantal straatnamen.
- Geef de langste straatnaam weer.
- Geeft naast de langste straatnaam ook de gemeente en provincie weer.
- Geef een lijst met straatnamen die uniek zijn (en toon ook gemeente en provincie).

4. Oefening - Container

Werk verder op de oefening Container uit vorig hoofdstuk (Polymorfisme en Interfaces).

4.1. Gegeven



4.2. Opdracht

In de vorige oefening werden twee klassen aangemaakt die elk de `IComparer` interface implementeren: één om containers op `massa` te sorteren, één om containers op `eigenaar` te sorteren.

- Stap 1:

- Pas de code aan zodat de klasse die sorteren op `massa` mogelijk maakt wordt vervangen door een lambda expressie in de applicatie.

```
Containers bij sorteren op massa:
90kg - Calais - 80m²
100kg - Brugge - 70m²
110kg - Rotterdam - 70m²
150kg - Antwerpen - 60m²
```

- Stap 2:
  - Pas de code aan zodat de klasse die sorteren op `eigenaar` mogelijk maakt wordt vervangen door een lambda expressie in de applicatie.

```
Containers bij sorteren op eigenaar:
Antwerpen - 60m² - 150kg
Brugge - 70m² - 100kg
Calais - 80m² - 90kg
Rotterdam - 70m² - 110kg
```

- Stap 3:
  - Breid de code uit zodat de applicatie ook sorteert op serienummer. Implementeer dit met een Named Function.

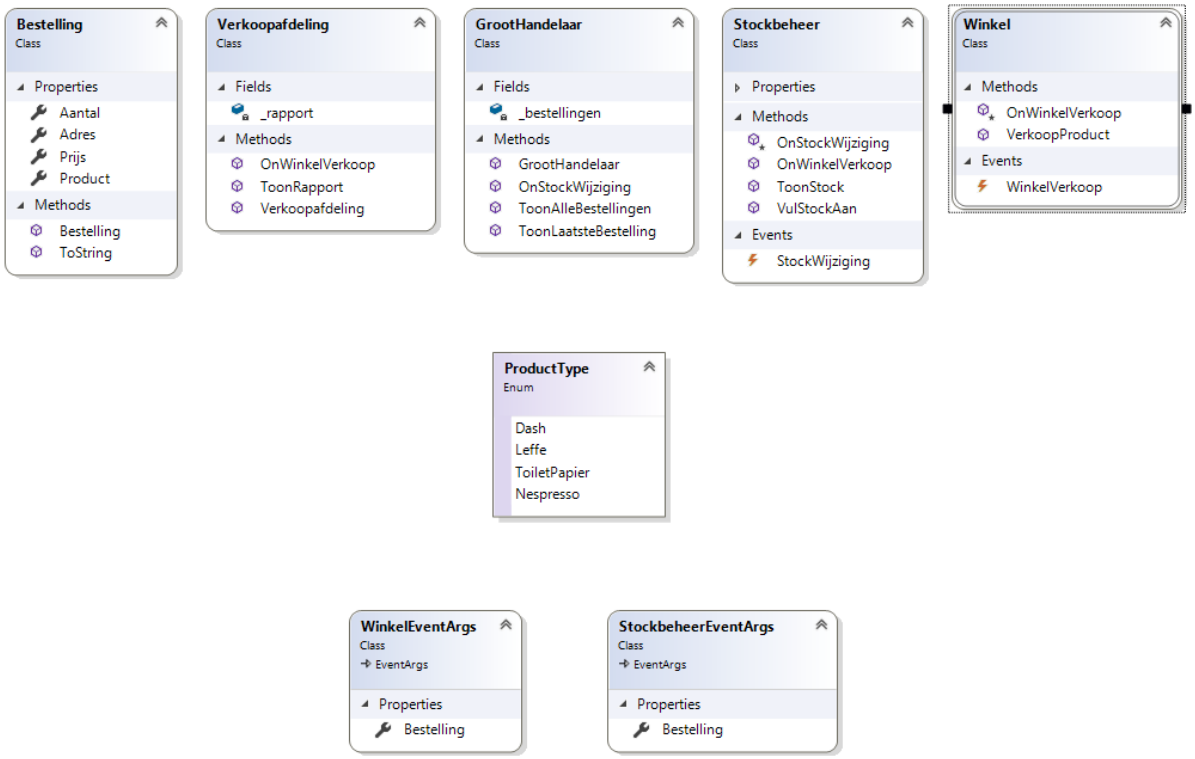
```
Containers bij sorteren op serienummer:
1234 - Antwerpen - 60m² - 150kg
2568 - Rotterdam - 70m² - 110kg
8564 - Brugge - 70m² - 100kg
8569 - Calais - 80m² - 90kg
```

5. Winkelmanagement met events

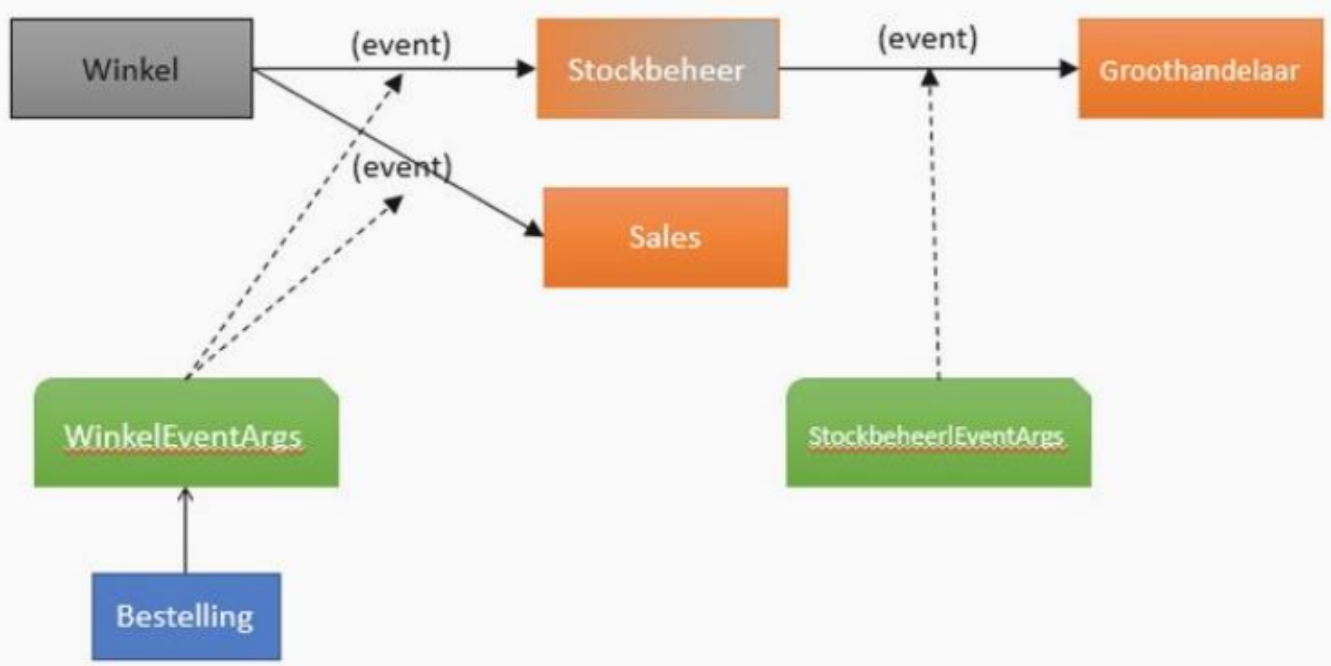
5.1. Inleiding

Een klant vraagt je om een applicatie te bouwen waarbij de verkoop in een winkel wordt opgevolgd door de verkoopafdeling en het magazijn.

5.2. Ontwerp: class diagram



5.3. Gebruik van events



5.4. Detailinformatie

De producten die verkocht worden, worden beschreven door middel van een enumeratie:

```
public enum ProductType { Dash, Leffe, ToiletPapier, Nespresso };
```

Wanneer een product verkocht wordt, noemen we dat een **Bestelling** en deze bevat volgende

informatie (we tonen de signatuur van de constructor):

```
public Bestelling(ProductType product, decimal prijs, int aantal, string adres)
```

Maak een klasse **Winkel** die enkel een methode heeft om producten te verkopen:

```
w.VerkoopProduct(new Bestelling(ProductType.Leffe, 50, 25, "Moerbeekstraat 25 -
Geraadsbergen"));
w.VerkoopProduct(new Bestelling(ProductType.Nespresso, 50, 25, "Moerbeekstraat 25 -
Geraadsbergen"));
w.VerkoopProduct(new Bestelling(ProductType.ToiletPapier, 100, 50, "Stationsstraat 10
- Zottegem"));
w.VerkoopProduct(new Bestelling(ProductType.Nespresso, 10, 95, "Moerbeekstraat 25 -
Geraadsbergen"));
```

Maak een klasse **Verkoopafdeling**. Deze beheert een inventaris van alle geplaatste bestellingen en biedt een methode aan om deze te rapporteren (via **Console.WriteLine()**). Bestellingen worden bijgehouden per klant: de klant wordt geïdentificeerd aan de hand van zijn/haar adres.

Maak een klasse **Stockbeheer** die voor elk product bijhoudt hoeveel er in stock is: we initialiseren de stock van elk product op 100. Telkens wanneer een bestelling aangemaakt wordt, moet de stock aangepast worden. Wanneer de stock van een bepaald product onder een minimumgrens valt, in dit geval 25, moet de groothandelaar geïnformeerd worden zodat een bestelling geplaatst wordt om de stock aan te vullen. De klasse Stockbeheer voorziet een methode om de stock aan te vullen en een methode om de stock te rapporteren (via **Console.WriteLine()**).

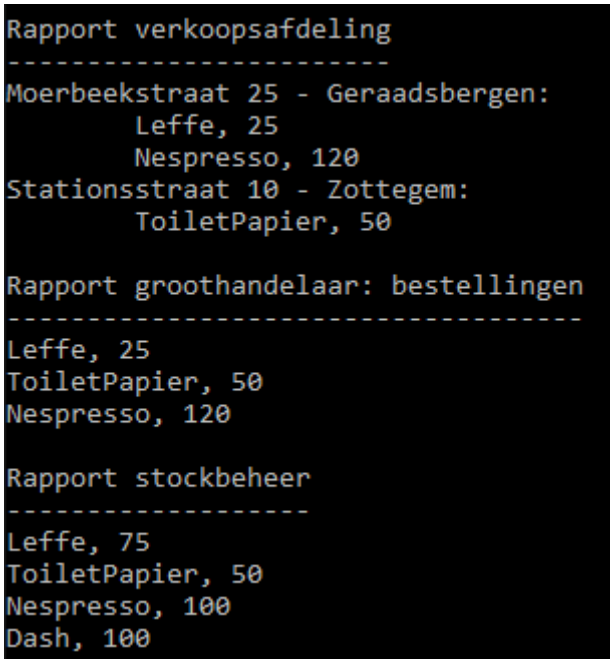
Maak een klasse **Groothandelaar**: deze houdt een lijst bij van alle bestellingen die geplaatst worden en biedt een methode aan om de laatste bestelling op te vragen en een methode om alle bestellingen te rapporteren (via Console.WriteLine()).

Stockbeheer en Verkoopafdeling schrijven zich in op het event WinkelEventArgs; dit event wordt opgeroepen bij het plaatsen van een bestelling. Groothandelaar schrijft zich in op het event StockbeheerEventArgs dat aangeboden wordt door Stockbeheer.

Maak een bibliotheek voor je business code en een aparte Console App.

5.5. Voorbeeld

Bij elke verkoop worden drie rapporten afgedrukt: dat van de verkoopafdeling, groothandelaar en stockbeheer. Na de laatste verkoop zien deze rapporten er als volgt uit:



6. Linq - Bushaltes

Haal eerst **bushaltes.zip** af op Chamillo. Dit is een drie lagen project waar de opzet al is gebeurd. De applicatie leest al een lijst met bushaltes in uit een json bestand dat ook in het project zit.

We gaan op deze collectie in de repository een aantal LINQ operaties uitvoeren. Je steekt best elk van deze operaties in een andere methode in de repository. Laat de gebruiker een nummer uit onderstaande lijst kiezen en print uit wat er in het corresponderende nummer staat in de lijst.

1. Print uit hoeveel haltes er zijn
2. Print alle halte namen uit
3. Geef lijst met de gemeentenamen, alfabetisch gesorteerd, zonder duplicaten.
4. Geef lijst van haltenamen voor een gemeente die de gebruiker kan kiezen.
5. Geef een lijst met haltes waar geen HalteToegankelijkheden zijn.
6. Print de gemeente met het meeste aantal haltes uit met het aantal.
7. Print voor elke gemeente uit hoeveel haltes ze heeft.
8. Voorzie een functie die voor 2 opgegeven gemeenten de gemeenschappelijke lijst van haltenamen weergeeft.
9. Voorzie een functie die de haltes weergeeft die enkel voorkomen in de door de gebruiker opgegeven gemeente, maar die niet voorkomen in bij de andere gemeenten.
10. Geef de langste haltenaam weer.
11. Geef een lijst met haltes die in meerdere gemeenten voorkomen (en print ook af in welke gemeenten ze voorkomen).

7. Linq - Bushaltes Toegankelijkheden

Als je gaat kijken naar het project van vorige oefening zie je dat de Bushalte klasse een property heeft genaamd HalteToegankelijkheden. Deze kan 0, 1 of meerdere toegankelijkheid opties aanduiden voor een halte.

Deze toegankelijkheden worden (als er zijn) gescheiden van elkaar met een puntkomma.

Lijst eerst alle mogelijke values op via een LINQ query. Nu je deze hebt uitgeprint kan je ze in een enum steken.

Voeg een property met enkel een getter toe die de HalteToegankelijkheden omzet in een lijst met deze enum values. Vervolgens kan je nog enkele operaties doen op de dataset

1. Print voor elke gemeente uit hoeveel haltes toegankelijkheden hebben en hoeveel niet.
2. Print uit welke gemeente procentueel de meeste haltes heeft met toegankelijkheden.
3. Print voor elke voorziening uit hoeveel haltes ze hebben