

Academiejaar 2018-2019		3^{de} examenperiode
Faculteit: FBO Opleiding, afstudeerrichting en jaar: 1TI/1TILE Naam van het opleidingsonderdeel: OOP2 (Eventueel) dOLOD / Deelexamen: Campus: Schoonmeersen - Aalst Lesgever(s): L. Lewyllie, A. Pierreux, H. Roobrouck, L. Stroobant, S. Vandermeersch, L. Van Steenberghe en L. Vuyge	Examendatum: 04 / 09 / 2019 Aanvangsuur examen: 9u15 PC-nummer:	
Naam en voornaam student:		
Studentennummer:		
Lesgever bij wie de student de onderwijsactiviteit volgde:	Lesgroep v/d onderwijsactiviteit:	
Behaald resultaat: _____ op _____		



Tijdens het examen mogen onderstaande hulpmiddelen gebruikt worden:

- Java How To Program (Early objects) – 11e editie (of eerder) - Deitel & Deitel Prentice Hall
- API-documentatie:
<https://docs.oracle.com/javase/10/docs/api>
- Ter beschikking gestelde (test)klassen en tekstbestand

Algemene richtlijnen:

- Vul het bovenstaande kader in.
- Heb je individuele onderwijs- en examenmaatregelen, noteer dan in de rechterbovenhoek van elke pagina IOEM (afkorting voor individuele onderwijs- en examenmaatregel).
- Controleer of deze examenbundel alle pagina's bevat. Indien een pagina ontbreekt, verwittig dan de lesgever of de toezichter zodat je een nieuw exemplaar kan ontvangen.
- Je mag geen enkele vorm van communicatie – noch draadloos noch online – gebruiken tijdens de examens (chatten, mailen, Messenger,...) tenzij anders aangegeven in de exameninstructies. GSM's en dergelijke moeten **UITGESCHAKELD** zijn (niet op stand-by, trillen,...). GSM's, smartphones, smartwatches enz. mogen tijdens de examens ook **NIET** gebruikt worden om de tijd te raadplegen. Het niet volgen van deze gedragscode wordt gesanctioneerd als "onregelmatigheden bij een examen" (artikel 55 van de onderwijs- en examenregeling).

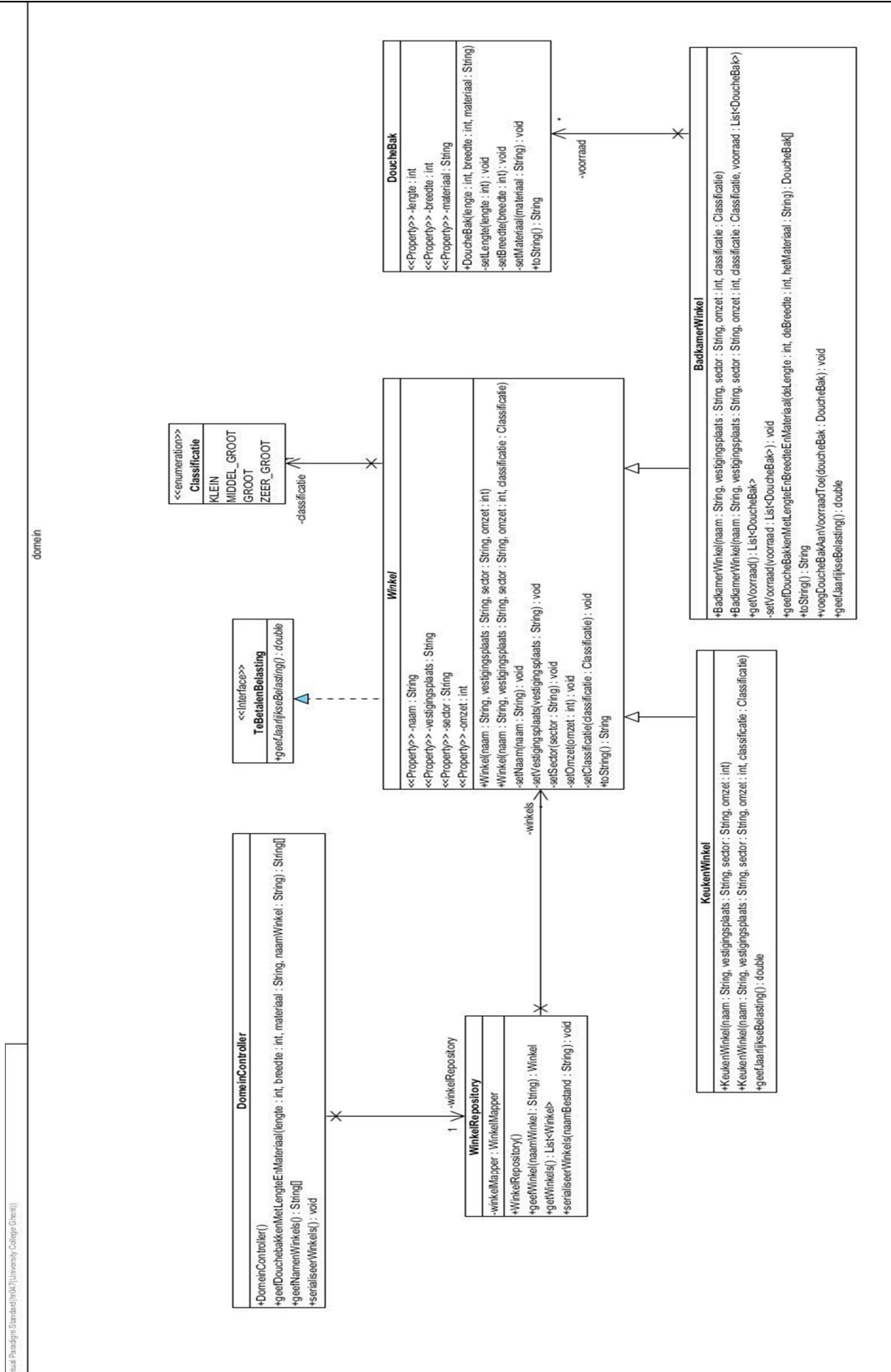
Bijzondere richtlijnen:

- Ga op je bureaublad naar het mapje met als naam jouw PC-nummer. Dit mapje gebruik je als workspace voor het NetBeans-project dat je voor het oplossen van dit examen aanmaakt.
- Noteer het PC-nummer = nummer op geel etiket PC in het kader van je examenopgaveblad.
- Maak alle implementaties in Java conform de gegeven UML en de gegeven uitvoer.
- Werk op de juiste manier: het ophalen van de resultaten van de deelopdrachten gebeurt steeds via de domeinController.

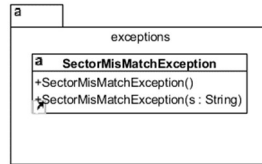
Aanvullen van het antwoordformulier:

- Ga op je bureaublad naar het mapje met als naam jouw PC-nummer. Hernoem het **Word**-document Antwoordformulier (Selecteer het Word-document => rechtermuisknop: Rename.) **NIET** het mapje. De nieuwe naam is **Klasgroep_Naam_Voor naam**, vb: 1.01_Jansens_Jan.
- Open het antwoordformulier en vul je naam, voornaam en PC-nummer in. Kopieer elk antwoord op de juiste plaats (kader) naar dit antwoordformulier. Doe dit na het oplossen van elke vraag. Niet op het einde van het examen. Bewaar je antwoordformulier regelmatig.
- Bij het einde van het examen sluit je NetBeans en Word af. Laat je PC aan staan en open het mapje met jouw antwoordformulier in.
- Het antwoordformulier wordt via NetSupport verzameld door de lectoren. Enkel het antwoordformulier wordt verbeterd.

Deel 1 is volledig gebaseerd op onderstaand UML-diagram (/70)



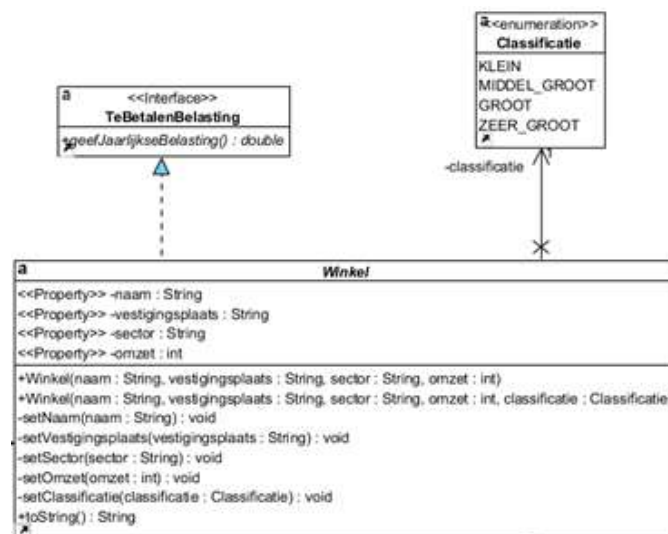
package exceptions



Creëer een eigengemaakte exceptionklasse `SectorMismatchException`. Deze exception zal geworpen worden bij het toekennen van een foutieve sector (zie verder Winkel). Het is een checked exception.

package domein

Werk de domeinklassen uit volgens het bovenstaande UML en meegegeven specificaties.



Interface TeBetalenBelasting

Schrijf de interface `TeBetalenBelasting` met de methode `geefJaarlijkseBelasting` uit.

Abstracte klasse Winkel

De klasse bevat een constructor waarbij alle attributen worden meegegeven als parameter en een 2^{de} constructor waarbij de `Classificatie` als defaultwaarde `KLEIN` krijgt.

De meegegeven argumenten worden eerst grondig gecontroleerd:

- De naam mag niet leeg zijn. Indien dit wel het geval is, wordt een `IllegalArgumentException` met passende foutboodschap gegooid.
- De sector mag niet leeg zijn en bestaat uit 3 hoofdletters gevolgd door exact 4 cijfers OF 5 kleine letters gevolgd door minstens 2 cijfers. Een `SectorMismatchException` met passende foutboodschap, wordt gegooid als het argument niet voldoet aan deze criteria.
- De omzet is een strikt positief getal. Indien het argument niet voldoet, wordt een `IllegalArgumentException` met passende foutboodschap gegooid.
- De classificatie mag niet leeg zijn.

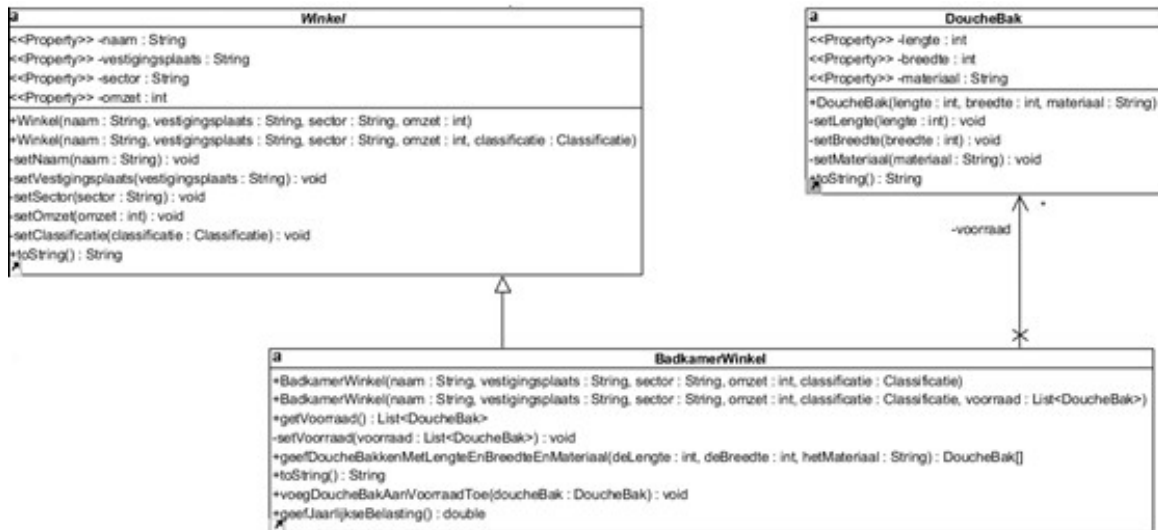
Alle attributen in de klasse hebben een getter en setter.

Voorbeelden tekstuele weergave van een winkel:

BadkamerWinkel Het Douchekot, in Aalst, sector = ABC1234, omzet = 5000

KeukenWinkel 't Keukenkot, in Herzele, sector = pipes123, omzet = 1200

Subklasse BadkamerWinkel



De klasse bevat één constructor waarbij alle attributen worden meegegeven als parameter en een 2^{de} waarbij geen voorraad wordt meegegeven. De defaultwaarde van voorraad is dan een lege lijst. Het attribuut voorraad heeft een bijhorende getter en setter.

Een methode geefDoucheBakkenMetLengteEnBreedteEnMateriaal, die uit de voorraad een array van doucheBakken teruggeeft. Deze doucheBakken hebben **minstens** een lengte deLengte en **minstens** een breedte deBreedte hebben én zijn van materiaal hetMateriaal. Maak een array met de gepaste grootte volgens het aantal doucheBakken die voldoen aan deze criteria.

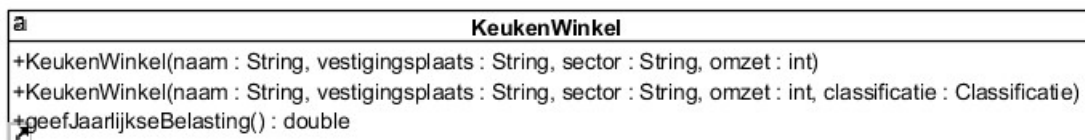
Methode toString geeft de gegevens van de badkamerwinkel terug en de info van douchebakken op de volgende manier:

```
[BadkamerWinkel Het Douchekot, in Aalst, sector= ABC1234, omzet= 5000
DoucheBak 1: lengte= 90 cm, breedte= 90 cm, materiaal= acryl
DoucheBak 2: lengte= 120 cm, breedte= 90 cm, materiaal= acryl
DoucheBak 3: lengte= 120 cm, breedte= 90 cm, materiaal= composiet
DoucheBak 4: lengte= 140 cm, breedte= 90 cm, materiaal= composiet]
```

De methode voegDoucheBakAanVoorraadToe voegt de meegegeven Douchebak toe aan de voorraad.

Een BadkamerWinkel betaalt als jaarlijkse belasting 2.5% van de omzet.

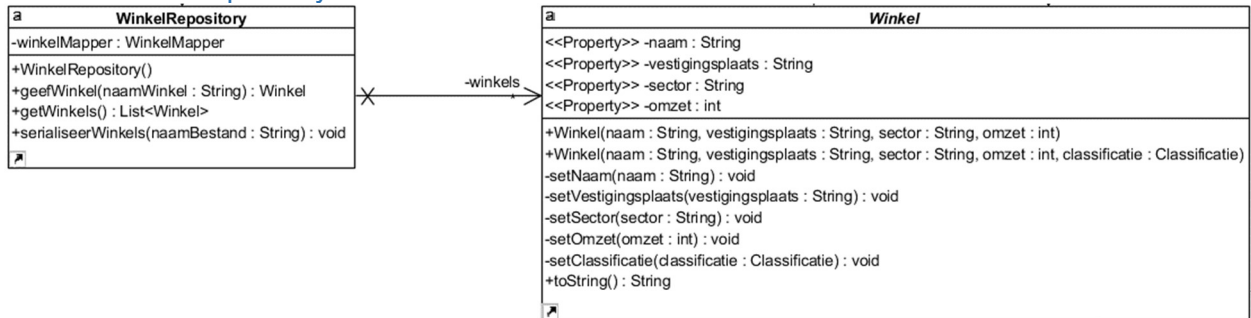
Subklasse KeukenWinkel



De klasse bevat één constructor waarbij alle attributen worden meegegeven als parameter en een 2^{de} constructor waarbij de Classificatie als defaultwaarde KLEIN krijgt.

Een keukenwinkel betaalt als jaarlijkse belasting een vaste bedrag van €2500.

Klasse WinkelRepository

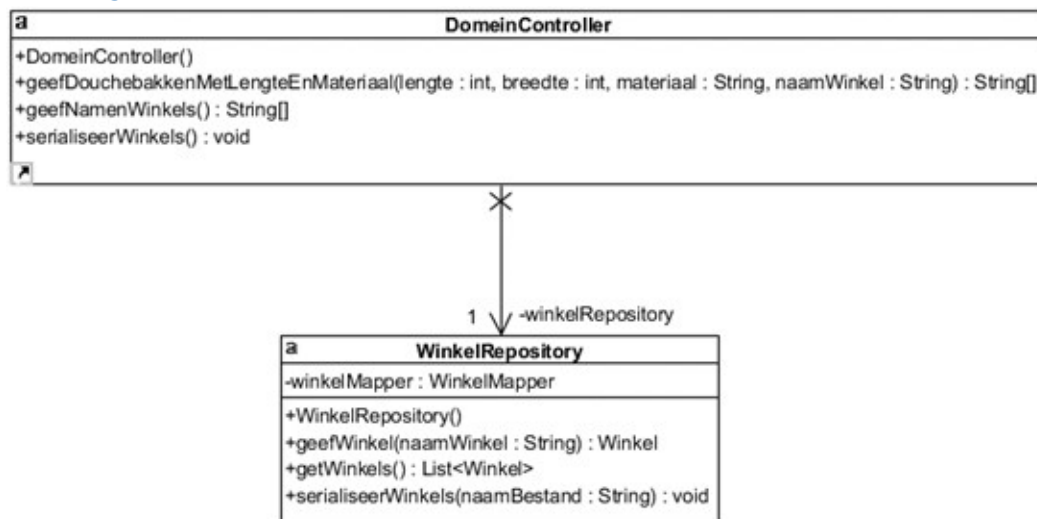


Bij het aanmaken van de WinkelRepository wordt de collectie van Winkel-objekten opgevuld met de data uit het tekstbestand "winkels_in.txt" (zie verder persistentielaag).

De methode geefWinkel geeft het Winkel-object terug met als naam de meegegeven parameter. Indien dergelijke winkel niet bestaat, wordt null teruggeven.

De methode serialiseerWinkels serialiseert de Winkel-objekten naar een bestand met de naam naamBestand (zie verder persistentielaag).

Klasse DomeinController



Voorzie de correcte constructor zonder parameters.

De methode geefDouchebakkenMetLengteEnMateriaal haalt de douchebakken op van de winkel met naam naamWinkel, die voldoen aan de lengte, breedte en materiaal.

De info over de douchebakken wordt als een array van Strings teruggegeven.

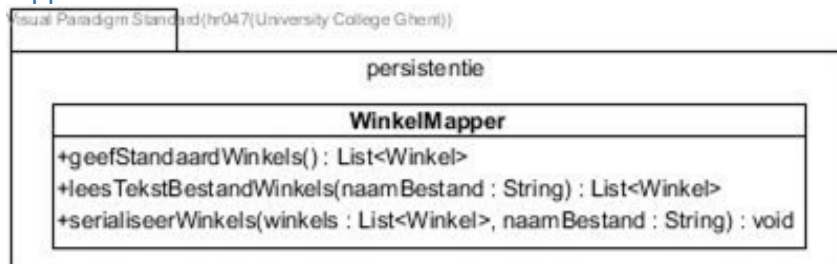
Indien de meegegeven naamWinkel leeg is of geen naam is van een BadkamerWinkel is, wordt een IllegalArgumentException met passende foutboodschap gegooit.

De methode geefNamenWinkels geeft enkel de namen terug van alle winkels.

De methode serialiseerWinkels serialiseert de lijst met winkels naar het bestand winkels_uit.ser.

package persistentie

Klasse WinkelMapper



De methode *leesTekstBestandWinkels* leest een tekstbestand met naam “*naamBestand*” waarin de informatie staat van een reeks winkels. Elke lijn in het bestand is als volgt opgebouwd:

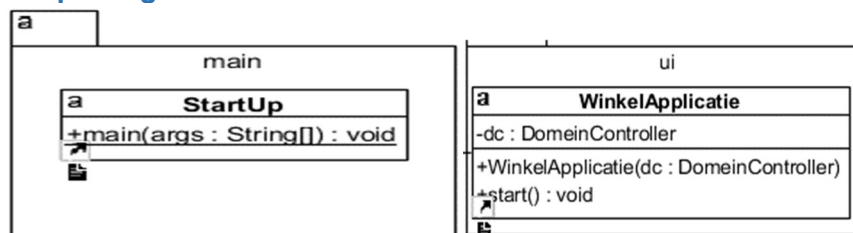
soort-naam-vestigingsplaats-sector-aantal personeelsleden-classificatie

De soort wordt aangegeven door “b” of “k” voor respectievelijk een badkamerwinkel of een keukenwinkel. Afhankelijk van het karakter dat daar dus staat zal je een *BadkamerWinkel*-object of een *KeukenWinkel*-object aan je lijst van winkels moeten toevoegen.

De code moet ook kunnen omgaan met fouten in het invoerbestand. Mocht er een record ongeldige informatie bevatten (een woord op de plaats van het aantal personeelsleden, een veld te weinig,...) moet die regel overgeslagen worden. **De rest van de regels moeten wel nog verwerkt worden. In de standaard errorlog krijgen we een melding te zien die zegt dat een regel overgeslagen werd.** Alle mogelijke exceptions moeten opgevangen worden en van een gepaste melding voorzien zijn.

De lijst van winkels moet ook kunnen weggeschreven worden. Dit gebeurt in de methode *serialiseerWinkels*. Deze krijgt een lijst winkels en de naam van het bestand waarin ze moeten weggeschreven worden. **De winkels worden één per één geserialiseerd, niet de volledige lijst ineens.** Zorg hier ook weer voor de afhandeling van de mogelijke exceptions.

package main en package ui

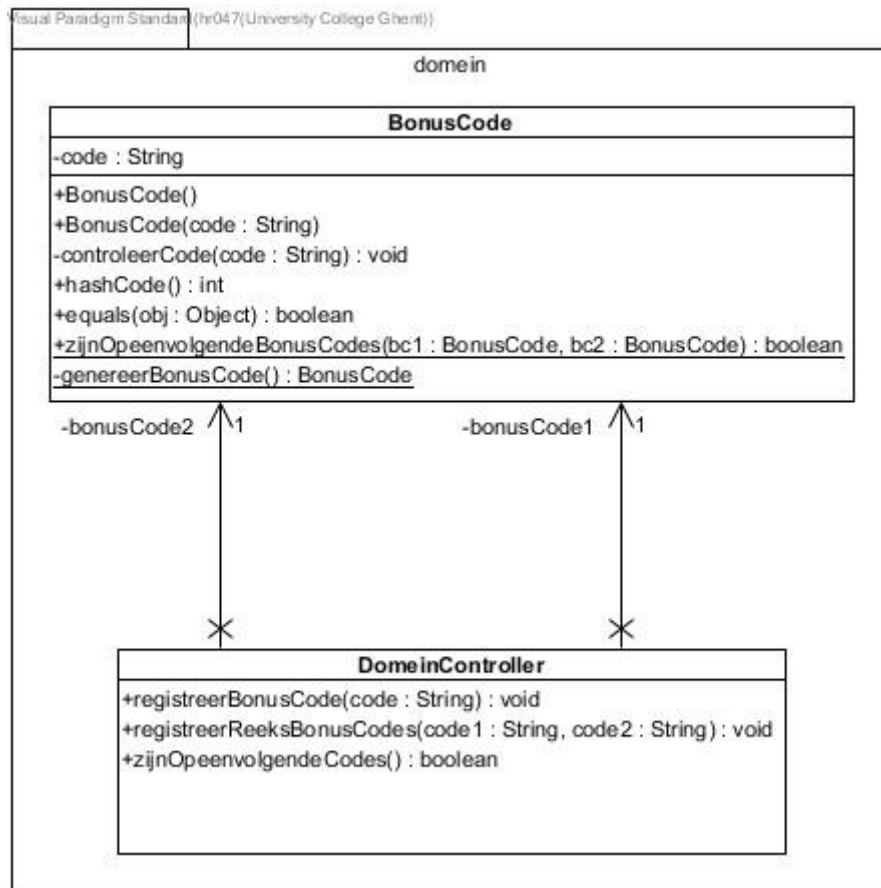


We werken de *WinkelApplicatie* uit, waarbij alle *Winkel*-objecten worden geserialiseerd naar het bestand **winkels_uit.ser**

Resultaat:



Respecteer bovenstaande UML.

Deel 2 is gebaseerd op onderstaand UML-diagram (/30)**package domein****Klasse BonusCode**

De default constructor maakt gebruik van de genereerBonusCode-methode. Deze methode genereert een BonusCode-object met een geldige code. De code begint met een A (unicode = 65), daarna 5 cijfers en dan 3 hoofdletters. Maak hierbij gebruik van StringBuilder

Schrijf de controleerCode-methode. Een geldige code is een code die begint met A, daarna 5 cijfers en 3 hoofdletters, vb.: A12345ABC.


Indien dit niet het geval is, wordt een IllegalArgumentException gegooid met een passende foutboodschap.

BonusCode-objecten zijn dezelfde als hun code identiek is. Vervolledig met de juiste methode(s) die deze informatie bevatten.

2 Bonuscodes zijn een geldige reeks als de letters van hun respectievelijke codes dezelfde zijn **en** het verschil tussen de getallen niet is groter dan 150. Gebruik deze informatie voor het implementeren van de methode zijnOpeenvolgendeBonusCodes.
(2 keer dezelfde BonusCode geeft *niet waar* als resultaat.)

Als we de applicatie starten hebben we volgende uitvoer:

Eerst moet in de comboBox gekozen worden voor ingave van **Een** of een **Reeks** bonusCodes:



BonusCode

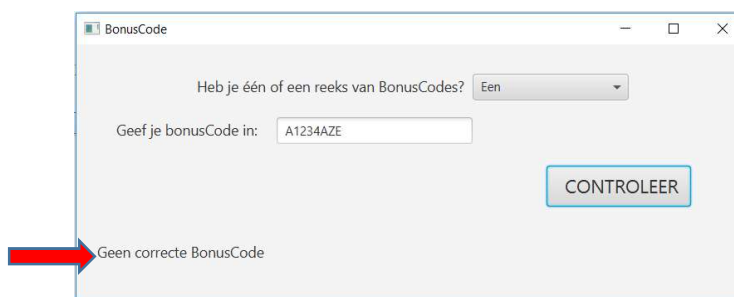
Heb je één of een reeks van BonusCodes? Keuze?

Geef je bonusCode in:

CONTROLEER

Maak eerst een keuze!

Bij een foutieve ingave van een bonusCode krijgen we de foutboodschap 'Geen correcte BonusCode' onderaan:



BonusCode

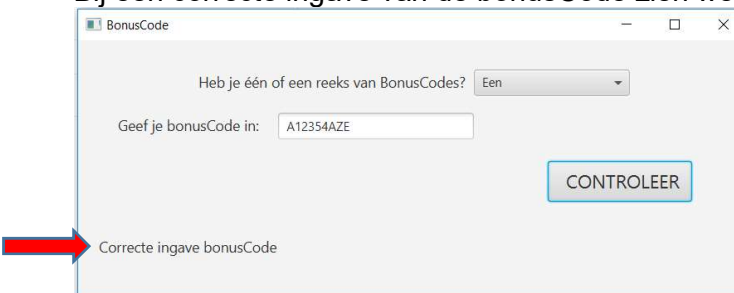
Heb je één of een reeks van BonusCodes? Een

Geef je bonusCode in: A1234AZE

CONTROLEER

Geen correcte BonusCode

Bij een correcte ingave van de bonusCode zien we 'Correcte ingave bonusCode':



BonusCode

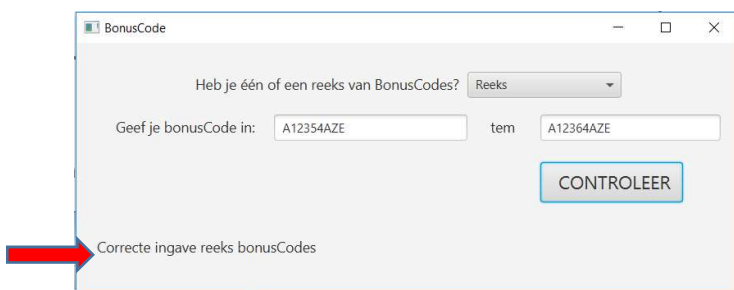
Heb je één of een reeks van BonusCodes? Een

Geef je bonusCode in: A12354AZE

CONTROLEER

Correcte ingave bonusCode

Bij een correcte ingave van een reeks van bonusCodes:



BonusCode

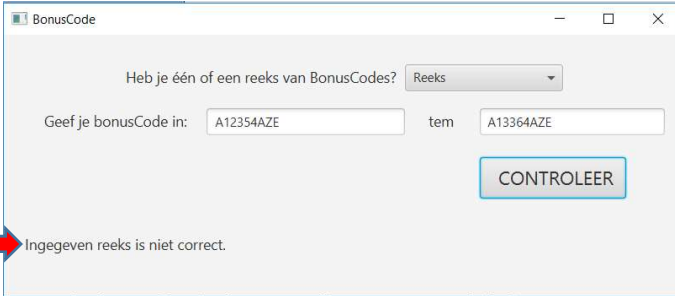
Heb je één of een reeks van BonusCodes? Reeks

Geef je bonusCode in: A12354AZE tem A12364AZE

CONTROLEER

Correcte ingave reeks bonusCodes

Bij een foutieve ingave van een reeks van bonusCodes:



The screenshot shows a web application window titled "BonusCode". Inside the window, there is a form with the following elements:

- A label: "Heb je één of een reeks van BonusCodes?"
- A dropdown menu: "Reeks" (selected)
- A label: "Geef je bonusCode in:"
- Two input fields: The first contains "A12354AZE", the second contains "A13364AZE".
- A label: "tem"
- A button: "CONTROLEER"

Below the input fields, a red arrow points to a message: "Ingegeven reeks is niet correct."

Vul de nodige code aan zodat bij het klikken op de knop Controleer wordt nagegaan of de ingegeven bonusCode of reeks van bonusCodes correct zijn.

Maak hierbij gebruik van de reeds aanwezige methodes in de DomeinController!