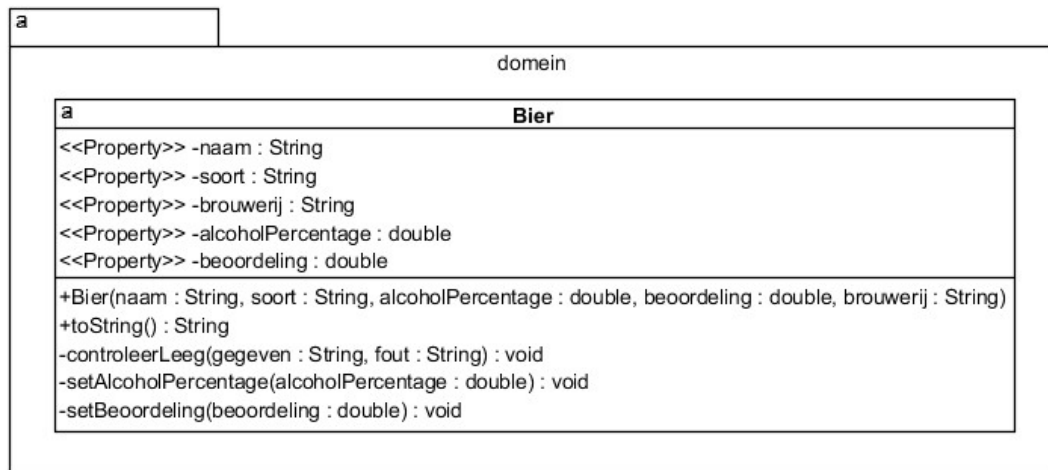
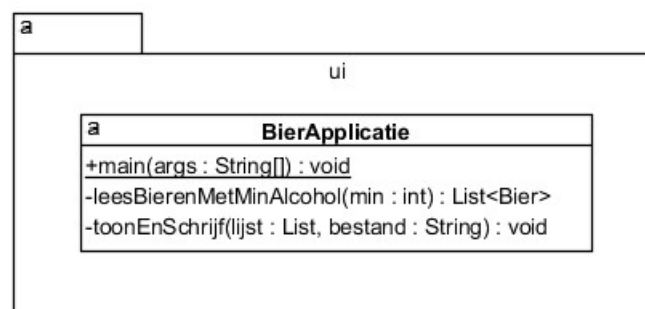


1. Bieren



Package domein:

- **Klasse Bier**
 - 5 attributen: naam (String, final), soort (String, final), brouwerij (String, final), alcoholgehalte (double), beoordeling (double)
 - Constructor met 5 parameters: de controle van de final attributen gebeurt via de methode controleerLeeg (Strings mogen niet leeg zijn)
 - Getters en private setters voor elk niet-final attribuut waarin het volgende wordt gecontroleerd: alcoholpercentage moet tussen 0 en 50 (grenzen inbegrepen) liggen en beoordeling moet een geldige score op 10 zijn.
 - Methode controleerLeeg: throwt een exception wanneer het gegeven leeg is. In de foutmelding wordt gebruik gemaakt van de tekst uit de tweede parameter.
 - Methode toString(): zie output



Package ui:

- **Klasse BierApplicatie**
 - Main-methode:
 - Lees eerst een getal in om het minimumpercentage alcohol te bepalen. Dit getal moet een percentage voorstellen en moet dus tot het interval [0,100] behoren. Enkel gehele getallen zijn toegelaten. Bij foute invoer wordt het getal opnieuw ingelezen.

- Bepaal daarna de lijst met bieren die aan het criterium voldoen met behulp van de methode leesBierenMetMinAlcohol.
- De bestandsnaam waarnaar de bieren weggeschreven moeten worden is **bierenMetXofMeerProcentAlcohol.txt** waarbij X het ingevoerde minimumpercentage voorstelt.
- Geef de lijst met bieren en deze bestandsnaam door aan de methode toonEnSchrijf om de lijst te tonen en weg te schrijven.
- Methode leesBierenMetMinAlcohol: lees uit het bestand “bieren.txt” enkel deze bieren die een alcoholpercentage van minimaal het opgegeven aantal hebben en stockeer alle informatie in een lijst van Bier-objecten die je teruggeeft aan de main-methode
- Methode toonEnSchrijf: laat de namen van alle bieren uit de gegeven lijst op het scherm zien en schrijf deze ook naar het opgegeven bestand

Voorbeeld:

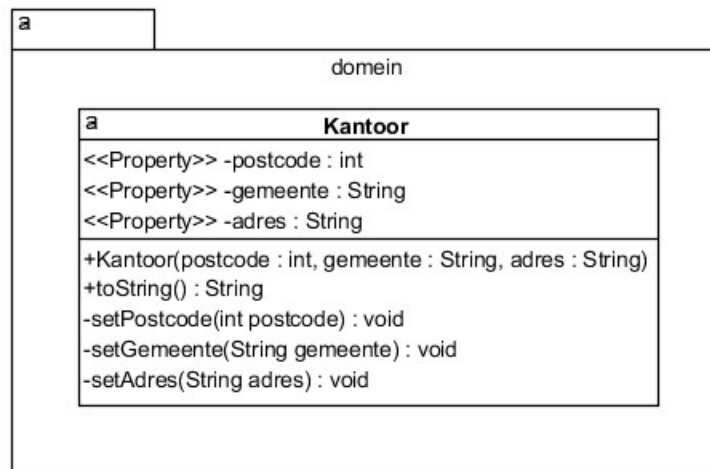
Input: tekstbestand bieren.txt (naam, soort, alcoholgehalte, beoordeling , brouwerij):

WestVleteren_Blond	blond	5.0	9.3	Sint-Sixtusabdij van Westvleteren
Tripel_Kanunnik	tripel	8.2	9.1	Wilderen
Black_Albert	donker	13.0	9.0	De Struise Brouwers
Rochefort_10	onbekend	11.0	9.1	Brasserie de l'Abbaye de Saint-Rémy
Alpaïde	donker	9.5	9.0	Nieuwhuys Hoegaarden
Cantillon_Geuze	geuze	5.0	8.5	Cantillon
Moinette_Blonde	blond	8.5	8.5	Dupont
Wilderen_Goud	blond	6.0	8.4	Wilderen
Tripel_Karmeliet	tripel	8.4	8.3	Bosteels
Westmalle_Tripel	tripel	9.5	8.2	Abdij der trappisten van Westmalle

Output: tekstbestand bierenMet8OfMeerProcentAlcohol.txt (enkel naam):

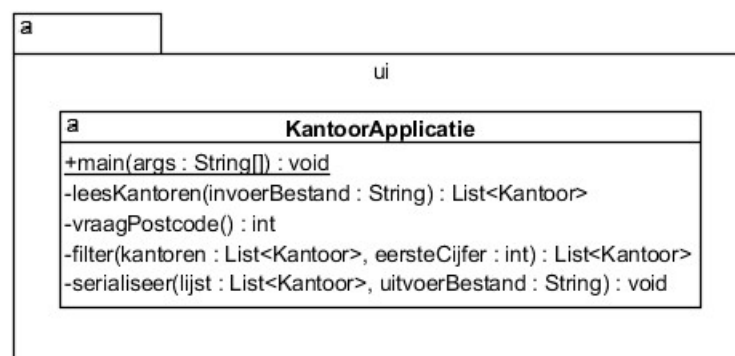
Tripel_Kanunnik
 Black_Albert
 Rochefort_10
 Alpaïde
 Moinette_Blonde
 Tripel_Karmeliet
 Westmalle_Tripel

2. Kantoren



Package domein:

- Kantoor
 - 3 attributen: postcode (int) , gemeente (String) en adres (= straat + huisnummer) (String)
 - Constructor met 3 parameters
 - Getters en setters voor elk attribuut: postcode moet een geldige Belgische postcode zijn, gemeente en adres mogen geen lege Strings zijn
 - toString(): zie output



Package ui:

- KantoorApplicatie
 - Main methode:
 - Lees alle kantoren in vanuit het bestand "adressen.txt"
 - Vraag vervolgens een geldige postcode op en bepaal hiervan het eerste cijfer
 - Zorg voor een lijst die enkel bestaat uit kantoren in een gemeente met een postcode die met hetzelfde eerste cijfer begint als de ingevoerde postcode.
 - Serialiseer deze lijst in een bestand met als naam "kantorenbeginnendmet?.ser", waarbij "?" vervangen wordt door het begincijfer van de postcode.

- Methode leesKantoren: lees het gegeven bestand uit en stockeer alle informatie in een lijst van Kantoor-objecten.
- Methode vraagPostcode: vraag een **geldige** Belgische postcode aan de gebruiker en blijf dit doen zolang de invoer ongeldig (geen geheel getal, te klein, te groot) is.
- Methode filter: maak een nieuwe lijst met alle Kantoor-objecten waarvan de postcode begint met hetzelfde cijfer als het opgegeven cijfer.
- Methode serialiseer: serialiseer de gefilterde lijst (in één keer) naar het opgegeven bestand

Voorbeeld:

Tekstbestand met adressen van klantenkantoren (adressen.txt):

1500 Halle Louis Vanbeverenstraat 27
 1800 Vilvoorde Tuchthuisstraat 2
 9000 Gent Sint-Pietersnieuwstraat 62
 9000 Gent Schoonmeersstraat 52
 9100 Sint-Niklaas Heistraat 88
 9160 Lokeren Oud-Strijderslaan 3
 9300 Aalst Dendermondsesteenweg 75
 9500 Geraardsbergen Kaai 15
 9600 Ronse Zonnestraat 55
 9800 Deinze Gentpoortstraat 20
 9900 Eeklo Molenstraat 58

Invoer:

```
run:
Geef je postcode: rt
Geef een getal in!
Geef je postcode: 10
Geef je postcode: 9000
BUILD SUCCESSFUL (total time: 7 seconds)
|
```

Uitvoer is een geserialiseerd bestand met adressen van kantoren met postcode beginnend met 9 (kantorenbeginnendmet9.ser)

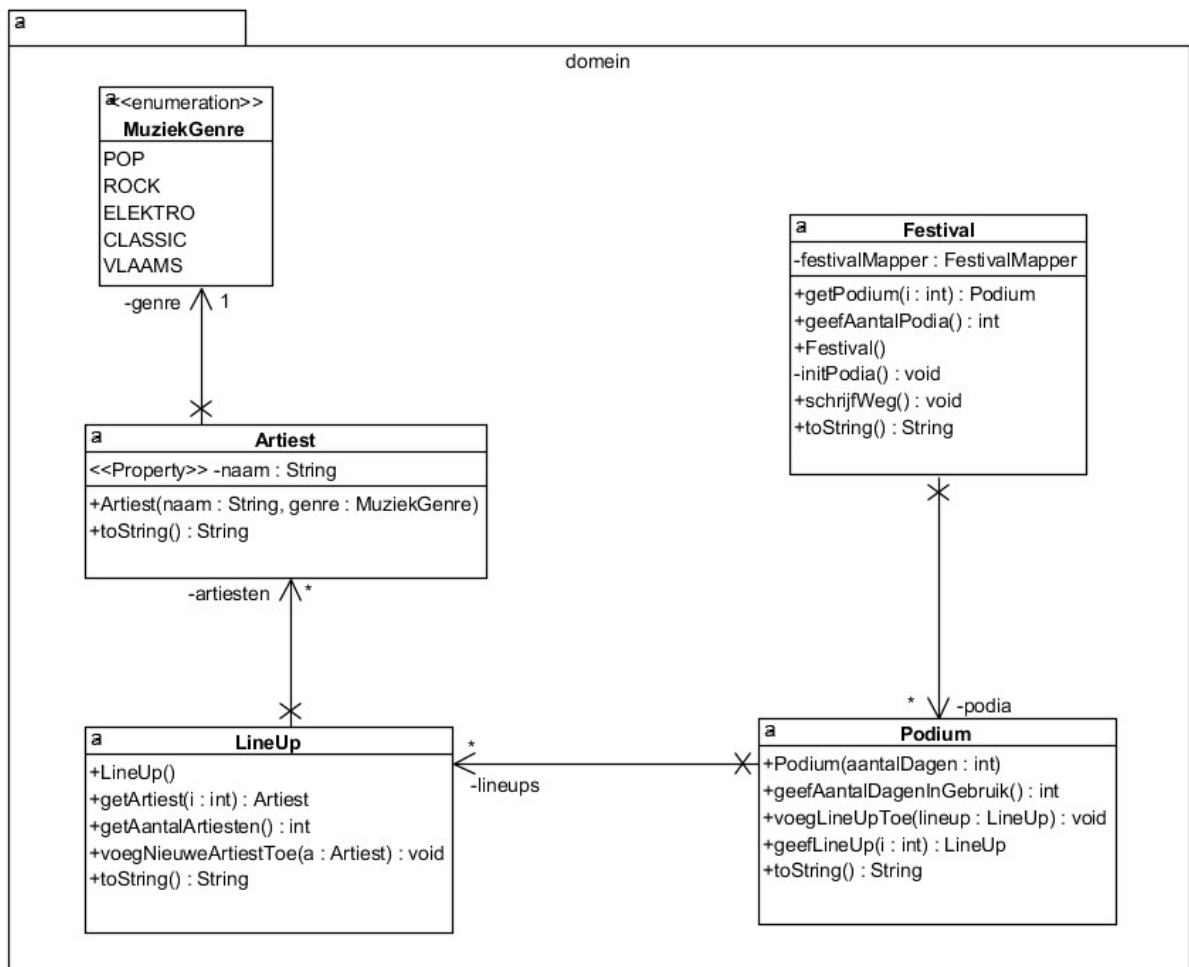
3. Festival

In een tekstbestand (“festival.txt”) krijgen we de gegevens van de artiesten van een festival. Een tweede tekstbestand (“podia.txt”) bevat de gegevens van elk podium op datzelfde festival.

Bedoeling is dat je aan de hand van deze invoerbestanden het volledige verloop van het festival (line-up per podium per dag) kunt uitschrijven op het scherm en serialiseren.

Realiseer hiervoor volgende klassen:

Package domein:



- **Enumeratie MuziekGenre:**
 - Bevat de volgende waarden: POP, ROCK, ELEKTRO, CLASSIC, VLAAMS
- **Klasse Artiest:**
 - Een artiest heeft als attributen zijn naam (final) en zijn genre. Gebruik voor dit laatste de bovenstaande enumeratie. Controleer of de naam wel degelijk inhoud bevat.
 - Getters voor beide attributen

- Maak een methode `toString` die alle gegevens van de `Artiest` in Stringvorm teruggeeft (zie uitvoer)
- **Klasse `LineUp`:**
 - Een `LineUp`-object bevat een lijst van artiesten die op een bepaalde dag in die volgorde optreden.
 - Maak een constructor aan, die de lijst initialiseert.
 - Maak een methode `voegNieuweArtiestToe` aan die een artiest toevoegt aan de arraylist.
 - Maak een methode `geefArtiest` die de artiest op plaats `i` ophaalt.
 - Maak een methode `geefAantalArtiesten` die het aantal artiesten in de lijst teruggeeft.
 - Maak een methode `toString` die alle gegevens van de `LineUp` in Stringvorm teruggeeft (zie uitvoer)
- **Klasse `Podium`:**
 - Deze klasse houdt de gegevens betreffende de optredens op een bepaald podium van het festival bij. Dit gebeurt als volgt: elk podium-object bevat een array van `LineUp`-objecten. Deze array is even groot als het aantal dagen dat er op dit podium muziek wordt gespeeld.
 - Maak een constructor die een array aanmaakt die even groot is als het opgegeven aantal dagen en deze initialiseert.
 - Maak een methode `geefAantalDagenInGebruik` die het aantal dagen dat het podium wordt gebruikt, teruggeeft.
 - Maak een methode `voegLineUpToe` die een `LineUp` achteraan de array toevoegt.
 - Maak een methode `geefLineUp` die een `LineUp` teruggeeft op een gegeven plaats van de array.
 - Maak een methode `toString` die alle gegevens van het `Podium` in Stringvorm teruggeeft (zie uitvoer)
- **Klasse `Festival`:**
 - Deze klasse houdt een lijst van podia bij.
 - Maak een constructor die de hulpmethode `initPodia` aanroept.
 - Maak een methode `initPodia` die via de persistentielaag eerst alle artiesten uitleest en dan de diverse podia van het festival aanmaakt.
 - Maak een methode `schrijfWeg` die ervoor zorgt dat via de persistentielaag het volledige huidige festivalobject weggeschreven kan worden.
 - Maak een methode `toString` die alle gegevens van het volledige `Festival` in Stringvorm teruggeeft (zie uitvoer)

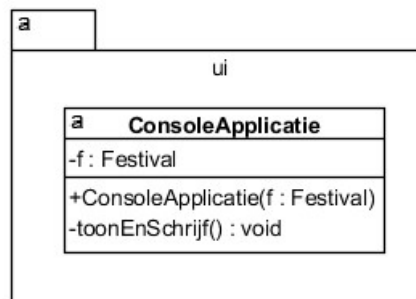
Package persistentie:



- **Klasse FestivalMapper**

- De methode leesArtiestenUit leest het tekstbestand met de meegegeven naam uit, en geeft alle informatie uit het bestand terug onder de vorm van een lijst van Artiest-objecten.
- In de methode maakPodia lees je de info uit het meegegeven bestand om te weten hoeveel podia er zijn (= aantal regels tekst), hoeveel line-ups/dagen per podium (= aantal getallen per regel) en hoeveel artiesten per line-up (= waarde van het getal). Op die manier maak je een lijst met Podium-objecten waarbij je per Podium de benodigde LineUps maakt en per LineUp alle artiesten toevoegt uit de meegegeven lijst van Artiest-objecten.
- De methode schrijfFestivalWeg serialiseert het meegegeven Festival-object naar het bestand "festival.ser".

Package ui:



- **Klasse ConsoleApplicatie**

- Bevat een constructor die een Festival-object binnenkrijgt. De constructor roept de methode toonEnSchrijf aan.
- De methode toonEnSchrijf toont alle gegevens van het Festival-object en roept de passende methode uit Festival-object aan om het object te serialiseren.

Met de gegevens van de voorbeeldbestanden, maak je volgende festivallayout:

	DAG 1		DAG 2		DAG 3	
PODIUM 1	Tom Jones	classic	Triggerfinger	rock	Milk Inc	pop
			Will Tura	vlaams		
			Britney Spears	pop		
PODIUM 2	David Guetta	elektro	Betty	vlaams	LEEG	
	Skrillex	elektro	Christoff	vlaams		

Dit resulteert in volgende voorbeelduitvoer:

```
run:
Festival met 2 podia

Podium 1
Dag 1
Artiest Tom Jones speelt het genre CLASSIC

Dag 2
Artiest Triggerfinger speelt het genre ROCK
Artiest Will Tura speelt het genre VLAAMS
Artiest Britney Spears speelt het genre POP

Dag 3
Artiest Milk Inc speelt het genre POP

Podium 2
Dag 1
Artiest David Guetta speelt het genre ELEKTRO
Artiest Skrillex speelt het genre ELEKTRO

Dag 2
Artiest Betty speelt het genre VLAAMS
Artiest Christoff speelt het genre VLAAMS

BUILD SUCCESSFUL (total time: 0 seconds)
```

Main-package:

- **Klasse StartUp:**
 - Start de applicatie op de correcte manier.

4. Verbouwing

Bouw volgend project uit om de gebruiker toe te laten een gepaste aannemer te kunnen kiezen bij een verbouwing.

package domein

Klasse Aannemer

- Van een Aannemer houden we bij wat de naam is van zijn bedrijf, welke type bedrijf het is, wat de afstand in kilometer is tot de plaats van de verbouwing en hoeveel procent de aannemer onder de basisprijs wil gaan. Daarnaast bevat de klasse Aannemer ook informatie over de kilometerprijs (0,50 euro – niet wijzigbaar) en de basisprijs, die voor alle aannemers dezelfde zijn.
- Er is ook een constructor die naam, type en afstand instelt en een random percentage uit het interval [0,20[bepaalt. De afstand en de basisprijs moeten positief zijn.

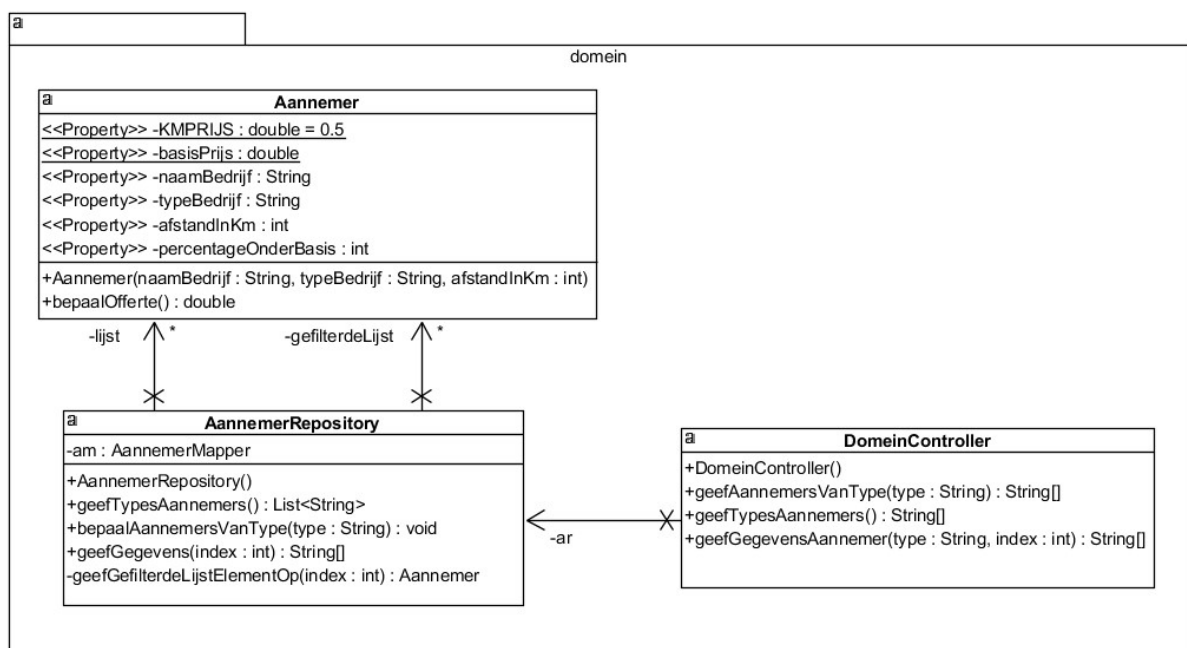
- De methode bepaalOfferte berekent de prijs van de offerte: deze ligt het opgegeven aantal percent onder de basisprijs en voegt daarbij nog de kilometerkost horend bij de af te leggen afstand.

Klasse AannemerRepository

- In de AannemerRepository hou je een mapper-object bij en 2 lijsten van Aannemers.
- In de constructor stel je de eerste lijst in door via de mapper alle Aannemers op te halen. De gefilterde lijst is leeg.
- De methode geefTypesAannemers geeft een lijst van Strings terug die alle verschillende types bevat van de Aannemers uit de (niet gefilterde) lijst.
- De methode bepaalAannemersVanType stelt de gefilterdeLijst in op alle aannemers uit de volledige lijst die van het juiste bedrijfstype zijn.
- De methode geefGegevens maakt gebruik van de hulpmethode geefGefilterdeLijstElementOp die de aannemer op de opgegeven index uit de gefilterde lijst haalt. De gegevens (naam bedrijf, type bedrijf en offerteprijs) van deze aannemer worden dan in deze methode in een array van (3) Strings gegoten.

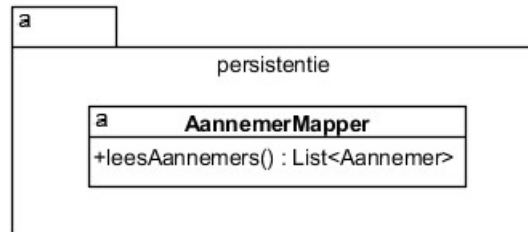
Klasse DomeinController

- Deze klasse bevat een verwijzing naar de AannemerRepository, die in de constructor wordt ingesteld.
- In geefAannemersVanType wordt een array van Strings teruggegeven met daarin de namen van de aannemers uit de gefilterde lijst met Aannemers van een bepaald type.
- De methode geefTypesAannemers geeft een array van Strings terug die alle verschillende types bevat die in de volledige lijst met aannemers voorkomen.
- In de methode geefGegevensAannemer worden eerst de aannemers van het opgegeven type bepaald en daarna worden de gegevens opgevraagd van de aannemer op de opgegeven index.



package persistentie

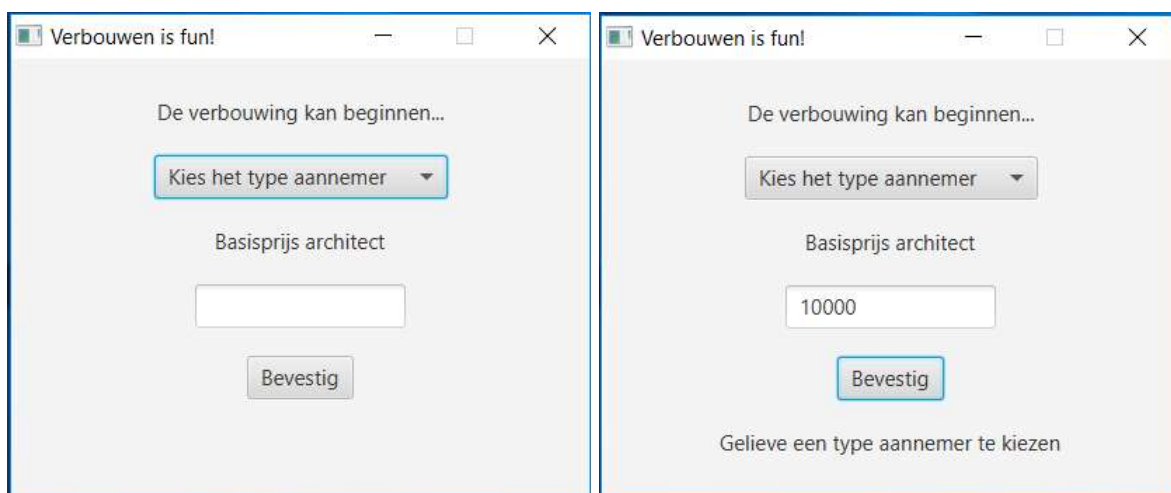
De klasse AannemerMapper bevat de methode leesAannemers die de lijst van Aannemers (in zijn geheel opgeslagen) uit het geserialiseerd bestand "aannemers.ser" haalt. Zorg ervoor dat alle mogelijke fouten opgevangen worden!



package gui

Klasse VerbouwingScherm

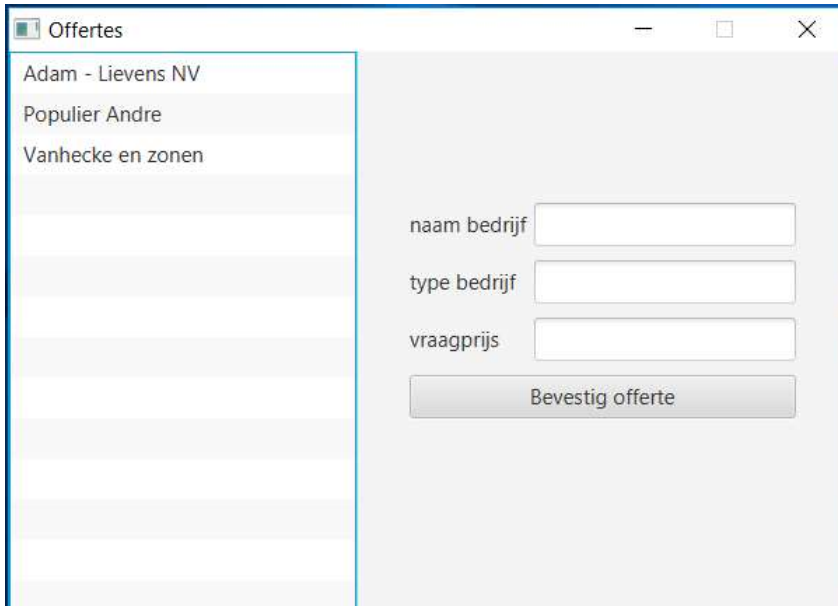
Op dit scherm kiest de gebruiker een type aannemer en vult hij de basisprijs in die de architect voorziet voor deze verbouwing.



- Methode buildGui:
 - Schrijf code om in de ComboBox een type aannemer te kunnen kiezen. Eens gekozen, wordt dit type bijgehouden in het attribuut.
 - Voorzie ook code die beschrijft wat er gebeurt als je op de knop "Bevestig" drukt:
 - Haal de prijs uit het tekstvak en stel deze in als basisprijs voor alle aannemers
 - Check of het type aannemer is ingesteld.
 - Indien ja, dan wordt de methode toonOfferteScherm aangeroepen.
 - Indien nee, dan wordt een passende foutboodschap op het Label getoond.
 - Vang ook andere mogelijke fouten op en toon telkens een gepaste foutmelding op het Label.
- Methode toonOfferteScherm:
 - Er wordt een nieuw scherm getoond met als titel "Offertes".
 - Dit scherm is 600 op 400 pixels groot.
 - Het OfferteScherm krijgt bij creatie de domeincontroller en het type mee.

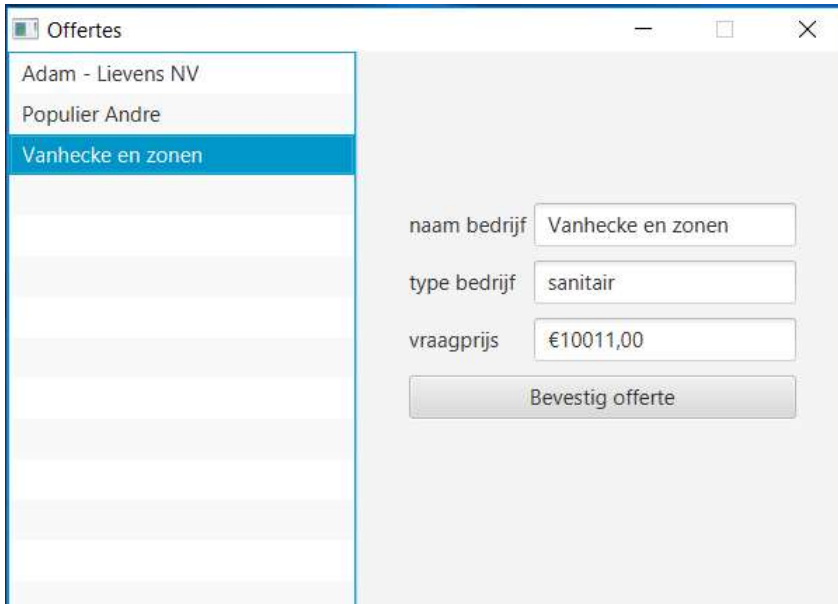
Klasse OfferteScherm

Op dit scherm kiest de gebruiker een aannemer uit de ListView zodat op het OffertePaneel de gegevens van deze aannemer zichtbaar worden.



The screenshot shows a window titled "Offertes". On the left is a list view containing three items: "Adam - Lievens NV", "Populier Andre", and "Vanhecke en zonen". The right side of the window contains three text input fields labeled "naam bedrijf", "type bedrijf", and "vraagprijs". Below these fields is a button labeled "Bevestig offerte".

Schrijf in de methode buildGui aan met code om in de ListView een aannemer te kunnen kiezen. Stuur de opdracht "update" naar het OffertePaneel en geef de juiste parameters mee.



The screenshot shows the same "Offertes" window. In the list view, "Vanhecke en zonen" is now selected and highlighted in blue. The form fields on the right are populated with data: "naam bedrijf" contains "Vanhecke en zonen", "type bedrijf" contains "sanitair", and "vraagprijs" contains "€10011,00". The "Bevestig offerte" button remains at the bottom.

Klasse OffertePaneel

Op dit paneel worden de gegevens van de geselecteerde aannemer getoond. Indien de gebruiker deze aannemer wenst te contracteren klikt hij op de knop om de offerte te aanvaarden. In dat geval wordt een popup getoond met alle gegevens van de offerte en wordt het programma afgesloten.



- Methode buildGui:
 - Voorzie code die beschrijft wat er moet gebeuren als je op de knop "Bevestig" drukt:
 - Toon een popup met volgende gegevens:
 - Titel: Offerte ... aanvaard (type bedrijf invullen)
 - Hoofding: Offerte van ... aangenomen (naam bedrijf invullen)
 - Tekst: Je betaalt ... voor deze aannemer (vraagprijs invullen)
 - Als de gebruiker op de OK-knop van deze popup klikt, moet het programma beëindigd worden.
- Methode update:
 - @param type: String die het type aannemer voorstelt
 - @param index: int die de index in de gefilterde lijst voorstelt
 - Haal de gegevens van de gekozen aannemer op en vul deze in de juiste tekstvakken in.

