

HO GENT

OOSDII

Polymorfisme & Interfaces - Oefeningen

Table of Contents

1. Oefening - Container	1
1.1. Gegeven	1
1.2. Opdracht	1
2. Oefening - Voorwerp - Interface Draagbaar	2
3. Oefening - Verplaatsing	3
3.1. package domein	4
3.2. Package persistentie	5
3.3. Package main	5
3.4. Package ui	5

1. Oefening - Container



Gebruik het startproject OOSDII_Polymorfisme_En_Interfaces_Oef_Container_Start vanop Chamilo

1.1. Gegeven

Container
<<Property>> -eigenaar : String <<Property>> -volume : int <<Property>> -massa : int <<Property>> -serialNumber : Integer
+Container(eigenaar : String, volume : int, massa : int, serialNumber : int) -setEigenaar(eigenaar : String) : void -setVolume(volume : int) : void -setMassa(massa : int) : void -controleerSerialNumber(serialNumber : Integer) : void

1.2. Opdracht

Vul de ontbrekende code aan zodat de applicatie correct kan worden uitgevoerd. Bij de start van de applicatie worden vier containerobjecten aangemaakt.

Onderaan vind je een voorbeeld van de uitvoer.

- Stap 1:
 - Breid de klasse Container uit zodat deze op een natuurlijke wijze kan gesorteerd worden. Een natuurlijke sortering vindt plaats op basis van het volume.
 - Voer deze sortering toe in de applicatie en genereer de uitvoer.

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

- Stap 2:
 - Breid het domein uit zodat, naast de natuurlijke sortering, ook kan gesorteerd worden op massa. Doe dit met een gewone klasse.
 - Voer deze sortering toe in de applicatie en genereer de uitvoer.

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

- Stap 3:
 - Breid het domein uit zodat, naast de natuurlijke sortering, ook kan gesorteerd worden op eigenaar. Let op dat je beschikbare code herbruikt: het type van het attribuut `eigenaar` is String. De String klasse implementeert de interface Comparable.
 - Voer deze sortering toe in de applicatie en genereer de uitvoer.

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

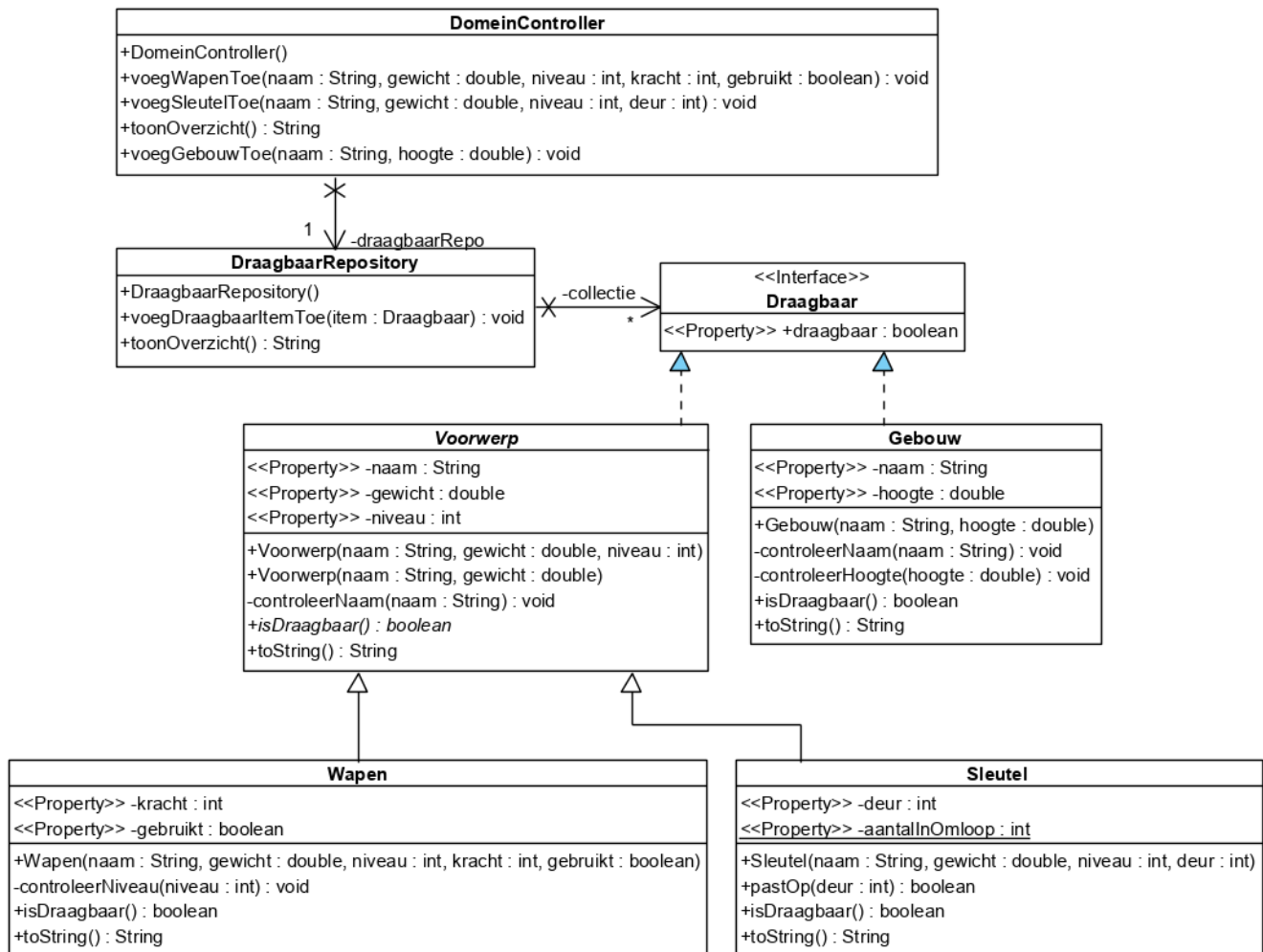
2. Oefening - Voorwerp - Interface Draagbaar



Werk verder op de oefening Voorwerp uit OOSD1 - H7. Vertrek van het project 'OOSDII_Polymorfisme_En_Interfaces_Oef_VoorwerpDraagbaar_Start'.

Breid het vorig domein verder uit met de interface Draagbaar die de methode isDraagbaar bevat, de klasse Gebouw en de DraagbaarRepository (ipv VoorwerpRepository) – **zie UML** (De hoogte van een gebouw is minimum 3 ; een gebouw is NIET draagbaar)

De klasse Voorwerp én de klasse Gebouw implementeren de interface Draagbaar:



Pas de andere klassen aan waar nodig en zorg ervoor dat je in de collectie OOK een Gebouw-object kan stockeren!

Wapen Colt met gewicht 1,500 kg uit niveau 3 en met kracht 6 nog niet gebruikt.
 Sleutel Voordeur met gewicht 0,500 kg uit niveau 3 past op deur 1.
 Er zijn 2 sleutel(s) in omloop.
 Wapen Brown met gewicht 0,500 kg uit niveau 1 en met kracht 23 al gebruikt.
 Sleutel Achterdeur met gewicht 0,500 kg uit niveau 1 past op deur 2.
 Er zijn 2 sleutel(s) in omloop.
 residentie Frankenstein met hoogte 4,5 is niet draagbaar.

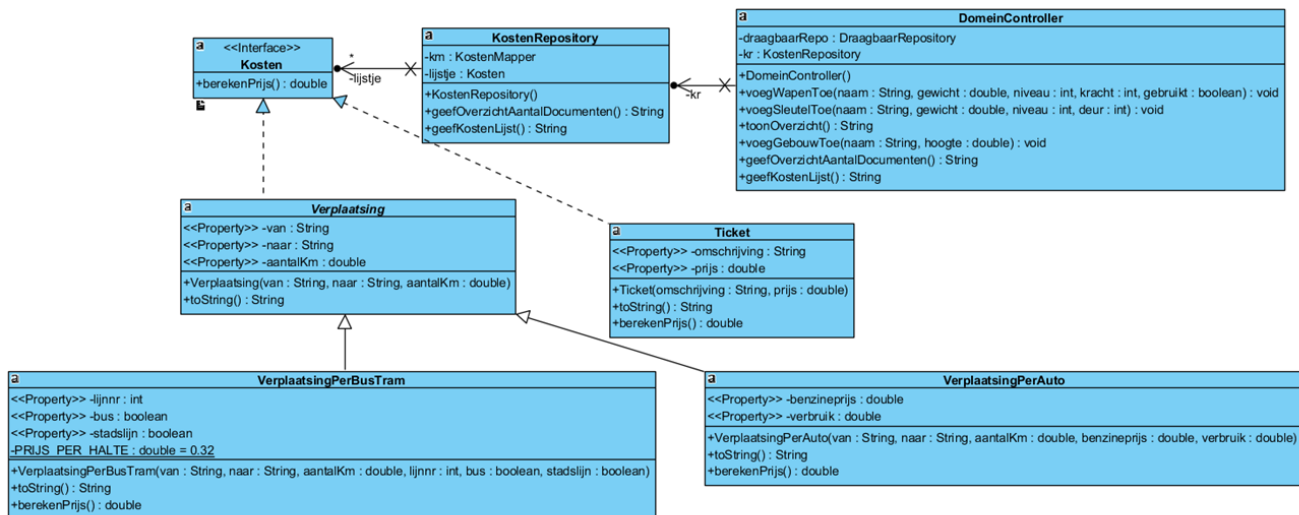
3. Oefening - Verplaatsing



Werk verder op de oefening Verplaatsing uit OOSD1 - H7. Vertrek van het project 'OOSDII_Polymorfisme_En_Interfaces_Oef_Voorwerp_Start'.

Breid het domein verder uit met de interface Kosten, de klasse Ticket, de KostenRepository en de DomeinController – zie UML

3.1. package domein



De interface Kosten:

- bevat de methode `berekenPrijs`.

De klasse Ticket (zie ook UML en testklasse):

- 2 attributen: omschrijving (verplicht, String) en prijs (double)
- constructor met 2 parameters
- getter voor elk attribuut
- setter voor de prijs:
 - prijs moet een strikt positief getal zijn, anders passende exception gooien met boodschap (via de methode `controleerPrijs`)
- De prijs van een ticket wordt berekend door de opgegeven prijs terug te geven.
- `toString`: voorzie 20 posities voor de omschrijving en een totale breedte van 10 posities voor de prijs (2 cijfers na de komma)

De klasse KostenRepository:

- 2 attributen : km (KostenMapper) en lijstje (List<Kosten>)
- constructor : lijst van kosten wordt hierin opgehaald uit de persistentielaag ahv het mapperobject
- methode `geefOverzichtAantalDocumenten`: geeft een String weer met de tekst “Volgende documenten werden ingediend: x ticket(s), y verplaatsing(en) per auto en z verplaatsing(en) per bus/tram” waarbij x, y en z in de methode bepaald worden ahv de gegevens uit de lijst
- methode `geefKostenlijst` geeft een String terug met daarin in tabelvorm (kolom 1: 70 posities breed, kolom 2: 15 posities breed) de gegevens van de kostenpost en de bijhorende prijs
 - voor een ticket worden als gegevens alleen de omschrijving getoond
 - voor een verplaatsing wordt 21% btw van de prijs afgetrokken alvorens de prijs getoond

wordt In deze methode bepalen we ook de totale kosten en geven dit in de String al laatste door.

De klasse Domeincontroller:

- 1 attribuut : kr (KostenRepository)
- constructor : de KostenRepository kr wordt gecreëerd en de domeincontroller kan hieraan vragen stellen om de informatie van de KostenRepository door te geven aan de GUI via de terugkeerwaarde van een methode.

3.2. Package persistentie

De klasse KostenMapper:

De methode geefKostenLijst geeft een lijstje terug met enkele kostenposten (verplaatsingen per auto en per bus/tram zowel als tickets door elkaar):

```
new VerplaatsingPerBusTram("Voskenslaan Gent", "Veldstraat Gent", 4.2, 1, false,
true));
new Ticket("Parkeerticket Flanders Expo", 5));
new VerplaatsingPerAuto("Gent", "Oudenaarde", 29.4, 1.591, 0.0538));
new Ticket("Toegangsticket beurs", 10));
new VerplaatsingPerBusTram("Laarne dorp", "Gent Sint-Pieters", 14.6, 34, true,
false));
new VerplaatsingPerAuto("Gent", "Brussel", 53.7, 1.488, 0.0692));
new VerplaatsingPerAuto("Gent", "Kortrijk", 47.1, 1.649, 0.084));//foutieve waarde!
```

3.3. Package main

In de klasse StartUp wordt de main-methode geschreven die de methode start uit de ui-klasse oproept. Vergeet niet een object van de DomeinController aan te maken.

3.4. Package ui

De start-methode in de klasse VerplaatsingApplicatie verzorgt onderstaande uitvoer. Zorg ervoor dat de overzichtstabel mooi uitgelijnd is zoals in onderstaand voorbeeld.

Mogelijke uitvoeren:

- Foutmelding in console door foutieve waarde in laatste object: -> een Exception wordt gegoooid.

Het verbruik moet tussen de 0.02 en de 0.07 liter per km liggen

- 0.084 wijzigen in 0.034 levert volgende output op:

Volgende documenten werden ingediend:

2 ticket(s), 3 verplaatsing(en) per auto en 2 verplaatsing(en) per bus/tram.

Overzicht gemaakte kosten:

	Kostenpost	Bedrag
verplaatsing van Voskenslaan Gent naar Veldstraat Gent met stadstram 1		2,12
	Parkeerticket Flanders Expo	5,00
verplaatsing van Gent naar Oudenaarde per auto		4,16
	Toegangsticket beurs	10,00
verplaatsing van Laarne dorp naar Gent Sint-Pieters met bus 34		7,93
verplaatsing van Gent naar Brussel per auto		9,14
verplaatsing van Gent naar Kortrijk per auto		4,36

Totaal gedeclareerde kosten = 42,71