Autoworld

Opmerking,

In deze tekst zijn per class de belangrijkste eigenschappen opgesomd. In de bijhorende JUnit test classes kunnen nog methods staan die je zal moeten implementeren om het geheel te laten werken. Die namen van die methods, hun parameters, de volgorde van de parameters haal je uit de voorbeelden in de test classes.

Nummerplaat

Maak een immutable class nummerplaat (immutable → final fields).
Plaats de class in een package be.vdab.voertuigen.div
Er is één constructor en die aanvaardt een String plaat en heeft default visibility.
Voorzie een getPlaat().
Voorzie een toString, een equals en een hashCode.
Zorg ervoor dat nummerplaten in een OutputStream kunnen bewaard worden.
Implementeer de interface Comparable.

DIV

Maak een class DIV.
Plaats de class in een package be.vdab.voertuigen.div.
Deze class is een singleton, gebruik hiervoor volgende code:

```
private static final DIV instance = new DIV();
```

Maak een method getInstance() die de waarde van de instance terug geeft.
Maat een method getNummerplaat, die nummerplaat objecten terug geeft.
Om de complexiteit rond de nummerplaat te beperken spreken we af dat:

- een nummerplaat start met AAA gevolgd door 3 cijfers. Je start met 001.
- telkens een nieuwe nummerplaat gevraagd wordt, verhoogd de nummer.
- eenmaal aan 999 gekomen mag terug verder gegaan worden met 001.

Zorg ervoor dat DIV nummerplaatobjecten kan maken, maar andere classes (buiten het package be.vdab.voertuigen.div) niet.

Datum

Maak een immutable class Datum.
Plaats de class in be.vdab.util
De class heeft 3 integer fields: dag, maand, jaar. Zorg ervoor deze fields slechts één maal een waarde kunnen krijgen (onmiddellijk bij de declaratie of in de constructor).
De class aanvaardt alleen geldige datums tussen 01/01/1583 en 31/12/4099.

Een poging om een foute datum te creëren leidt tot een DatumExcepetion.
De constructor aanvaart 3 integers (dag, maand, jaar).
Voorzie de class van de nodige getters.
Voorzie een toString (in het formaat dd/mm/jjjj) , een equals en een hashCode.
Zorg ervoor dat datums in een OutputStream kunnen bewaard worden.
Implementeer de interface Comparable.

DatumException

De class is afgeleid van Exception en heeft 4 constructors, net zoals Exception.
Verder zijn er geen methods nodig.
Plaats de class in be.vdab.util

Rijbewijs

De class zit in het package be.vdab.util.mens
Mogelijke waarden: A, B, C, D, BE, CE, DE

Bij het omzetten naar een string wordt BE, CE en De getoond met een + teken tussen de twee letters.

Mens

De class zit in het package be.vdab.util.mens
De class heeft de volgende fields:

- de string naam.
- een verzameling rijbewijzen.

Voertuig

Plaats de class in be.vdab.voertuigen.
De class heeft de volgende fields:

- nummerplaat, class Nummerplaat. Het field nummerplaat wordt bij declaratie onmiddellijk een waarde gegeven en kan later niet meer worden gewijzigd (Systeem van NL, een nummerplaat wordt toegekend aan een voertuig en niet aan de eigenaar).
- Merk, een String
- DatumEersteIngebruikname, een Datum
- Aankoopprijs, een int.
- Zitplaatsen, een final int.

De class is abstract.
De eerste vijf parameters van de constructor van Voertuig zijn: een merk, een datum van eerste ingebruikname en een aankoopprijs, aantal zitplaatsen en een Mens.

Er kan eventueel ook een zesde, zevende,… n-tigste parameter zijn. Dit is telkens een Mens.
De eerste Mens uit de lijst van de parameters is de bestuurder van het voertuig. De andere (de optionele parameters) zijn inzittenden. Wanneer er teveel inzittenden zouden zijn dan ontstaat de MensException, die in hetzelfde package als Mens zit.
Als je de lijst van ingezetenen opvraagt, zit de bestuurder bij in deze lijst.
Er kunnen niet meer ingezetenen zijn dan er zitplaatsen zijn in een voertuig.
De bestuurder moet een geschikt rijbewijs hebben voor het voertuig.
Er moeten inzittenden kunnen uitstappen en bij instappen. Er moet een andere bestuurder kunnen plaats nemen. Een voertuig moet steeds een bestuurder hebben. Elk voertuig heeft minstens één inzittende, anders krijg ontstaat er een IllegalArgumentException.
Voorzie de nodige setters en getters.
Voorzie een toString, equals en HashCode.
De equals maak je op basis van de nummerplaat.
Implementeer de interface Comparable op basis van nummerplaat.
Zorg ervoor dat het mogelijk is om voertuigen in een OutputStream kunnen bewaard worden.
Voorzien een Comparator op basis van merk en eentje op basis van aankoopprijs.
Werk de comparators uit als inner classes.
Maak voor elke comparator een static getter method om de comparator op te vragen.

Personenwagen

Leidt de class af van voertuig.
Plaats de class in be.vdab.voertuigen.
Er is één field , de ava.awt.Color kleur.
Voorzie de benodigde setters en getters.
Een personenwagen heeft een constructor met parameters om de fields te initialiseren.
Override de nodige methods.
Een personenwagen heeft maximaal 8 zitplaatsen. Een ongeldige waarde leidt tot een IllegalArgumentException.

Maat

Creëer een enum Maat met waardes: centimeter, decimeter en meter.

Volume

Plaats de class in be.vdab.util
De class volume is immutable, de 4 fields kunnen slechts één maal een waarde krijgen (onmiddellijk bij de declaratie of in de constructor).
Een volume heeft 4 fields zijn hoogte, breedte, diepte en maat
Maat is een Maat object, de andere field zijn int's.
Een volume heeft een constructor met 4 parameters.

Voorzien een method getVolume die het volume berekent, het resultaat is een long.
De equals maak je op basis van de 4 fields.
Implementeer de interface Comparable op basis van volume.
Zorg ervoor dat volumes in een OutputStream kunnen bewaard worden.
Negatieve volume's kunnen niet en leiden tot een VolumeException.

VolumeException

Plaats de class in be.vdab.util
De class is afgeleid van Exception en heeft 4 constructors, net zoals Exception.
Verder zijn er geen methods nodig.

Laadbaar

Deze interface zit in de package be.vdab.util
De interface definieert een getter en een setter voor laadvolume.

Pickup

Deze class in afgeleid van Personenwagen en implementeert Laadbaar.
Voorzie de nodige getters en setter, override de nodige methods.
De class heeft een constructor om alle fields te initialiseren.

Vrachtwagen

Deze class in afgeleid van voertuig en implementeert Laadbaar.
De class heeft 3 fields laadvolume, de int masimaalToegelatenMassa en de int aantalAssen.
Voorzie de nodige getters en setter, override de nodige methods.
Een vrachtwagen heeft maximum 3 zitplaatsen, anders ontstaat er een IllegalArgumentException.

Boekentas

De class Boekentas zit in het package be.vdab.schoolgerief en implementeert Laadbaar.
De class heeft twee fields, laadvolume en kleur (een String).
Voorzien een constructor met parameters om de fields te initialiseren.
Voorzie de nodige getters en setters en override de nodige methods.
Zorg ervoor dat Boekentassen in een OutputStream kunnen bewaard worden.
Voorzie een toString, equals en HashCode.
De equals maak je op basis van de laadvolume en kleur.
Laadvolume en kleur moeten ingevuld worden, zoniet wordt een IllegalArgumentException gethrowd.

Programma

Creëer een SortedSet van voertuigen en voorzie die van 6 voertuigen (2 personenwagens, 2 pick-ups en 2 vrachtwagens) en print de lijst uit.
Zorg ervoor dat 2 voertuigen tot één en hetzelfde merk behoren.

Kopieer de eerste sorted set in een tweede en sorteer die op basis van aankoopprijs (in omgekeerde volgorde) en print de lijst uit.

Kopieer de eerste sorted set in een derde en sorteer die op basis van merk. Print deze lijst uit en bewaar de voertuigen in een bestand “wagenpark.ser”.

Lees “wagenpark.ser” in een vierde sorted set en print deze uit.