**Opgave 3: Class diagrams and Java code - Basic**

### 3.1 Domeinmodel

Maak een UML-klassediagram op basis van de onderstaande Leverbon met de aanvullende informatie uit de praktijk van Praxa.

Onderken klassen, attributen en associaties en teken de meest passende UML-concepten.  
Ga uit van het domein klassendiagram dat we in de opgave 1 hebben opgesteld en gebruik daar klassen en associaties uit. Gebruik dezelfde klassennamen.

**PRAXA Leveringsbon**

Aan Vestiging: Deventer

Adres: Ysselkade 123

Leveringsnummer: 268156

Leveringsdatum: 15-07-20XX

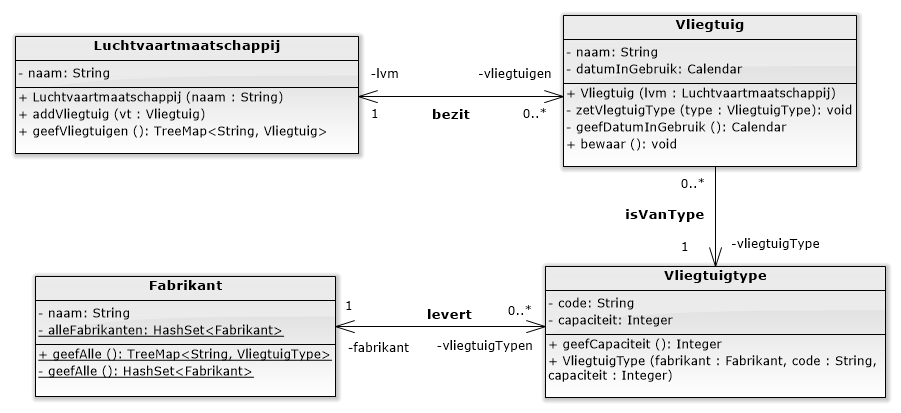
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ProductNr** | **Omschrijving** | **Hoeveelheid** | **BestelNr** | **BestelDatum** |
| 159012 | Schroeven 4\*35 100st | 100 | 137524 | 14-07-20XX |
| 159014 | Schroeven 4\*25 100st | 200 | 137524 | 14-07-20XX |
| 159015 | Schroeven 4\*20 100st | 100 | 137499 | 13-07-20XX |
| 157084 | Boorset steen/metaal | 150 | 137499 | 13-07-20XX |
| 157084 | Boorset steen/metaal | 50 | 137524 | 14-07-20XX |
| 157088 | Boorset steen | 100 | 137376 | 08-07-20XX |

Aanvullende informatie:

* Van product 159012 zijn er 300 besteld in bestelling 137524 door vestiging Deventer.
* Let op het verschil in de twee leverregels van product 157084!
* Als de actuele voorraad van een product in een winkel beneden de normvoorraad (minimum voorraad van dat product in die winkel) komt, wordt door het systeem voorgesteld om de aanvulnorm (het aantal bij te bestellen van dat product in die winkel) bij te bestellen. De magazijnbeheerder kan dit aantal te bestellen verhogen of verlagen, voordat de bestelling de deur uit gaat.
* Bestel- en levernummers moeten uniek zijn (over alle winkels heen).
* Als een bestelling gedaan wordt krijgt deze de status “geregistreerd”. Als de bestelde producten gedeeltelijk geleverd zijn, dan krijgt het de status “geleverd”. Als alle producten geleverd zijn krijgt de bestelling de status “afgehandeld”. Dan wordt tevens de datum vastgelegd bij de bestelling, waarop de laatste levering verwerkt is.

**3.2 Implementatie-klassediagram vertalen naar Java-code**

Vertaal het onderstaande klassediagram naar Java code. Pas de besproken vertaalregels toe.



**Java Collections**

* A collection is a data structure for holding elements.  
  A Java collection is a class that holds objects
* Set collections do not allow an element in the class to occur more than once.
* HashSet<Fabrikant> 🡪 Ongesorteerde Set van Fabrikant-objecten.
* List collections allow duplicates.
* ArrayList<Fabrikant> 🡪 Ongesorteerde List van Fabrikant-objecten.
* Map collections deal with collections of ordered pairs of *key* and *value*.
* TreeMap<String, VliegtuigType> 🡪 Map van in dit geval *pairs* van
* Key: String = de code v/h VliegtuigType
* Value: VliegtuigType = link naar een VliegtuigType object
* De Map is gesorteerd op de key.
* De methode get(key) retourneerd de value die bij de key hoort.  
  Je kan dus makkelijk zoeken in een Map.