

## Programming Essentials 2

### Werkcollege 6

De opdracht van dit werkcollege is het ontwikkelen van een applicatie voor een luchthaven.

#### Deel 1: Passagiers

Er zijn 3 soorten passagiers: *VIP*, *Business* en *Economy*. Van alle soorten passagiers houden we de naam bij en de opstartdatum van hun dossier. De datum wordt automatisch toegevoegd bij de aanmaak van hun dossier. Bekijk de Javadoc documentatie van de klassen *LocalDate*, *LocalTime* en *LocalDateTime*. Zoek een manier om de huidige datum bij te houden.

Zorg voor een constructor waar je de naam kan aan meegeven.

Alle soorten passagiers moeten via `System.out.println` hun naam en hun type afdrukken. Verplicht alle soorten passagiers de methode *geefSpecifiekeInfo* te implementeren.

*VIP* en *Business* passagiers hebben een aantal waardebonnen (voucher). Een waardebon heeft een identificatienummer en dient specifiek voor een gratis maaltijd, een frisdrank of een alcoholische drank (gebruik hiervoor een enumeratie). Passagiers kunnen nooit tweemaal dezelfde voucher bezitten.

#### Deel 2: Vluchten

Een vlucht heeft een identificatienummer (kan niet aangepast worden na creatie) en een aantal passagiers. Kies een collection die toelaat om de passagier samen met zijn zitplaats (voorbeeld *A1,B2*) bij te houden. (Hint: de passagier vormt de *key*, de zitplaats is de *value*.) Zorg ervoor dat de collection gesorteerd is op de naam van de passagier.

Voorzie ook de volgende methoden (bepaal zelf welke argumenten er nodig zijn en wat het beste type is ervoor):

- *boarding*: voegt een passagier toe en gooit een *PassagierReedsAanwezig*-exception indien de passagier reeds aanwezig is
- *save*: bewaart de vluchtgegevens. We kiezen voor een leesbaar (unicode) tekstbestand. Als naam van het bestand neem je identificatienummer.csv. (Waarbij je natuurlijk identificatienummer vervangt door het identificatienummer van de vlucht.) Per regel zet je de naam van de passagier en zijn zitplaats, gescheiden door een komma. Deze methode is zelf verantwoordelijk voor de foutafhandeling en geeft aan via een boolean of de gegevens al dan niet werden bewaard.
- *zoek*: geeft de passagier terug of gooit een *OnbekendePassagier*-exceptie als de passagier zich niet in de collectie bevindt.