**Klassendiagram**

**Rapport**: week, boekjaar  
**Medewerker**: naam, geboortedatum, functie, straat, huisnummer, postcode, woonplaats  
**Consultant**: Specialisatie, vraagtarief, contracturen, contracttarief  
**Verkoper**: verkoopdoel, gerealiseerd, bonus  
**Contracten**: contractnr, contractDatum, omschrijving, contractWaarde, beginDatum, ingangsdatum  
**Klant**: klantCode, plaats

Bonus: bedrag  
Medewerker is een superklasse  
Verkoper en consultant implementeren medewerker

Rapport 🡪 Verkoper  
Verkoper 🡪 Contract 🡪 Klant  
Contract 🡪 Consultant

**Sequence Diagram**

**Na opstarten:**2x getAlleVliegvelden (loop) note: alfabetisch gesorteerd.  
output 🡪 Vertrekpunt en Bestemming

**Na selectie vertrekpunt en bestemming:**getGeldigeVluchten():getVertrekPunt (loop) note: vluchten waarvan vertrekpunt overéénkomen (for-loop)  
getBestemming (loop) note: vluchten waarvan bestemming overéénkomen (for-loop)  
getVertrekDatum(loop) note: vluchten waarvan vertrekdatum groter of gelijk (for-loop)  
output 🡪 vertrektijden (datum en tijd)

**Na selectie van vertrekdatum en tijd**  
getVlucht() note: vind Vlucht object die geselecteerde Vlucht equals  
getBoeking() (loop) note: laat alles zien (for-loop)  
perBoeking 🡪 getNaam() getPlaats, getAantalStoelen, get KlasType  
output 🡪 klant-naam, -plaats, aantal stoelen en type klasse  
Vlucht{  
 For-loop {  
 Boeking for-loop{  
 klant.getNaam(), klant.getPlaats(), boekingRegel.getAantalStoelen(),  
 boekingregel.getKlasType()  
 }  
 }  
}

**Cancel()**Ga terug naar de aanroepende functie

note: vertrekdatum groter of gelijk is een systeemdatum (for-loop)