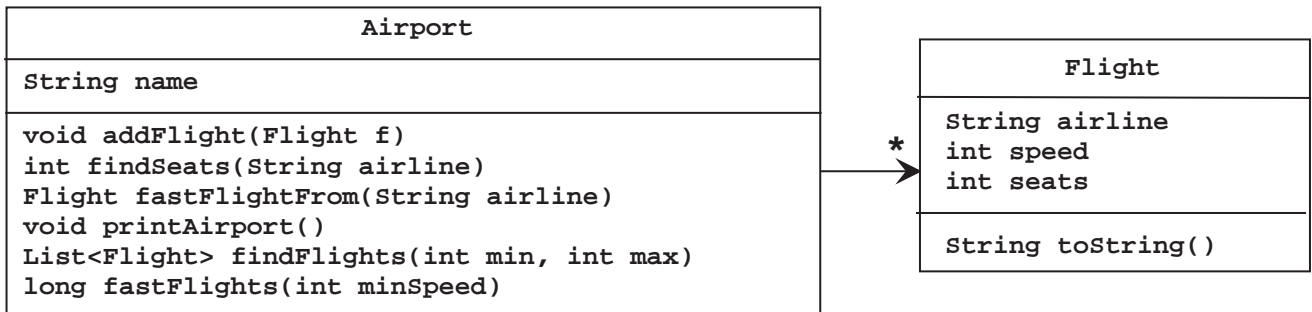


# Flight



1. Opret en klasse, *Flight*, hvis objekter repræsenterer et fly. Klassen er specificeret til højre i UML-diagrammet. Feltvariableerne skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en tekststreng på formen:

**"SAS, 800 km/hour, 100 seats".**

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Flight*-objekter i *test*-metoden, via objektreferencer *f1*, *f2*, *f3*, *f4* og *f5*, og udskriv disse ved hjælp af *toString*-metoden.

**Tilkald en instruktør og demonstrer det, som du har lavet indtil nu (T1).**

4. Opret en ny klasse, *Airport*, hvis objekter repræsenterer en lufthavn med fly. Klassen og dens relation til *Flight*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addFlight*, der tilføjer *Flight*-objektet *f* til *Airport*-objektet.
6. Opret et objekt af typen *Airport* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Flight*-objekter hertil.
7. Programmér metoden *findSeats*. Metoden skal returnere det samlede antal sæder på de fly, der tilhører det angivne luftfartsselskab. Udvid *Flight*-klassen med de nødvendige accessormetoder.
8. Afprøv den skrevne metode i *test*-metoden i *TestDriver*-klassen.

**Tilkald en instruktør og demonstrer det, som du har lavet siden forrige tjekpunkt (T2).**

9. Programmér metoden *fastFlightFrom*. Metoden skal returnere det hurtigste fly fra det angivne luftfartsselskab. Afprøv den skrevne metode i *test*-metoden.

**Tilkald en instruktør og demonstrer det, som du har lavet siden forrige tjekpunkt (T3).**

10. Programmér metoden *printAirport*. Metoden skal udskrive navnet på lufthavnen efterfulgt af alle fly sorteret alfabetisk efter luftfartsselskab. Hvis to fly tilhører samme selskab, sorteres efter antal sæder (højest til lavest). Afprøv den skrevne metode i *test*-metoden.

**Tilkald en instruktør og demonstrer det, som du har lavet siden forrige tjekpunkt (T4).**

11. Brug funktionel programmering til at implementere metoden *findFlights*. Metoden skal returnere alle fly, der har en hastighed, der ligger mellem de to parametre (begge inklusive). Det kan antages, at den anden parameter er større end eller lig den første. Afprøv den skrevne metode i *test*-metoden.

**Tilkald en instruktør og demonstrer det, som du har lavet siden forrige tjekpunkt (T5).**

12. Brug funktionel programmering til at implementere metoden *fastFlights*. Metoden skal returnere antallet af fly, der mindst har den angivne hastighed. Afprøv den skrevne metode i *test*-metoden.

**Tilkald en instruktør og demonstrer det, som du har lavet siden forrige tjekpunkt (T6).**