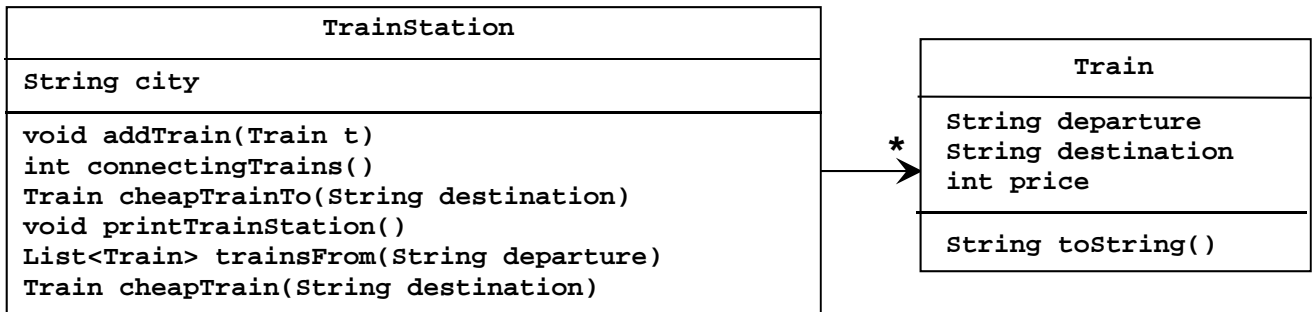


Train



1. Opret en klasse, *Train*, hvis objekter repræsenterer et tog. 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:

"From Aarhus to Odense for 150 DKK".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Train*-objekter i *test*-metoden, via objektreferencer *t1*, *t2*, *t3*, *t4* og *t5*, 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, *TrainStation*, hvis objekter repræsenterer en station med tog. Klassen og dens relation til *Train*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addTrain*, der tilføjer *Train*-objektet *t* til *TrainStation*-objektet.
6. Opret et objekt af typen *TrainStation* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Train*-objekter hertil.
7. Programmér metoden *connectingTrains*. Metoden skal returnere antallet af tog, der starter eller slutter i den by, hvor stationen ligger. Udvid *Train*-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 *cheapTrainTo*. Metoden skal returnere det billigste tog med den angivne destination (det er lige meget, hvor toget starter). Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printTrainStation*. Metoden skal udskrive byen, hvori stationen ligger, efterfulgt af alle tog sorteret alfabetisk efter startby. Hvis to tog starter i samme by, sorteres efter pris (lavest til højest). 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 *trainsFrom*. Metoden skal returnere alle tog, der starter i den angivne by og slutter i den by, hvor stationen ligger. 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 *cheapTrain*. Metoden skal returnere det billigste tog, der starter i den by, hvor stationen ligger, og slutter i den angivne by. Afprøv den skrevne metode i *test*-metoden.

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