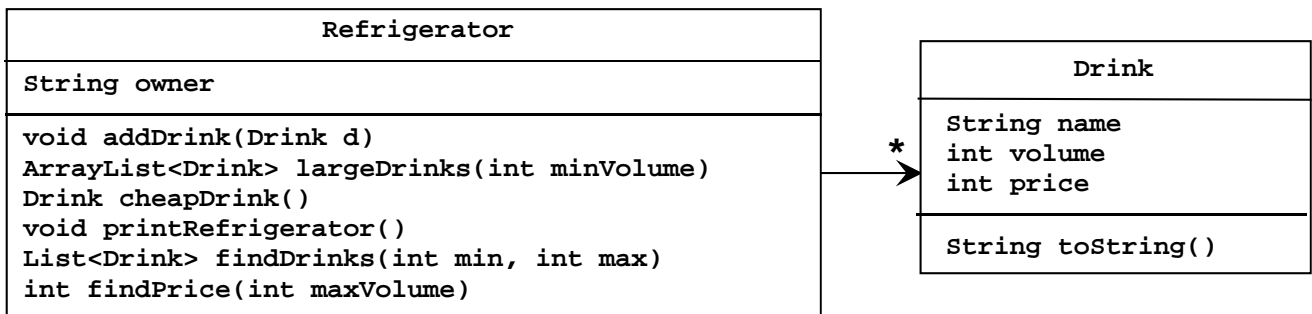


Drink



1. Opret en klasse, *Drink*, hvis objekter repræsenterer drikkevarer. 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:

"Milk, 100 cl for 8 DKK".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Drink*-objekter i *test*-metoden, via objektreferencer *d1*, *d2*, *d3*, *d4* og *d5*, 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, *Refrigerator*, hvis objekter repræsenterer et køleskab med drikkevarer. Klassen og dens relation til *Drink*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addDrink*, der tilføjer *Drink*-objektet *d* til *Refrigerator*-objektet.
6. Opret et objekt af typen *Refrigerator* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Drink*-objekter hertil.
7. Programmér metoden *largeDrinks*. Metoden skal returnere alle de drikke, der mindst har det angivne volumen. Udvid *Drink*-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 *cheapDrink*. Metoden skal returnere den billigste drik. Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printRefrigerator*. Metoden skal udskrive ejeren af køleskabet efterfulgt af alle drikkevarer sorteret alfabetisk efter navn. Hvis to drikke har samme navn, 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 *findDrinks*. Metoden skal returnere alle drikke, hvis pris 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 *findPrice*. Metoden skal returnere den samlede pris for de drikke, der højst har det angivne volumen. Afprøv den skrevne metode i *test*-metoden.

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