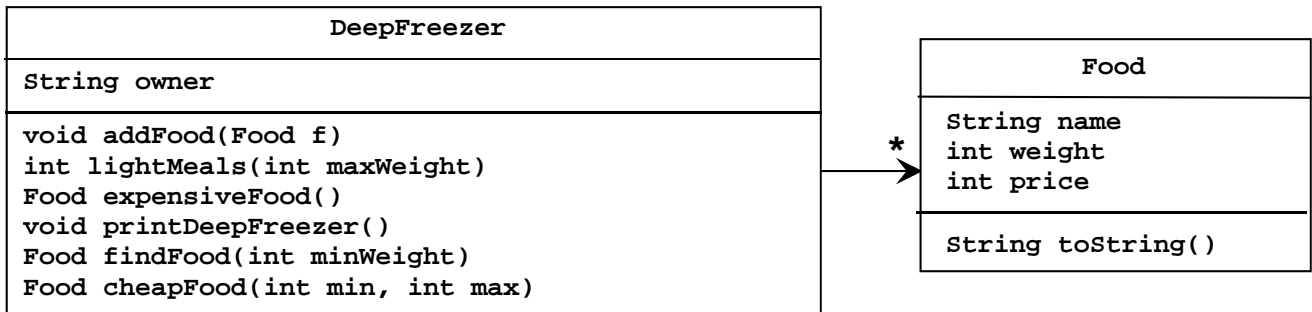


Food



1. Opret en klasse, *Food*, hvis objekter repræsenterer en madvare. 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:

"200 gram cheese for 15 DKK".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Food*-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, *DeepFreezer*, hvis objekter repræsenterer en dybfryser med madvarer. Klassen og dens relation til *Food*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addFood*, der tilføjer *Food*-objektet *f* til *DeepFreezer*-objektet.
6. Opret et objekt af typen *DeepFreezer* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Food*-objekter hertil.
7. Programmér metoden *lightMeals*. Metoden skal returnere antallet af madvarer, hvis vægt er mindre end den angivne. Udvid *Food*-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 *expensiveFood*. Metoden skal returnere den dyreste madvare. Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printDeepFreezer*. Metoden skal udskrive ejeren af dybfryseren, efterfulgt af alle madvarer sorteret alfabetisk efter navn. Hvis to madvarer har samme navn, sorteres efter vægt (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 *findFood*. Metoden skal returnere en madvare, der mindst har den angivne vægt. 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 *cheapFood*. Metoden skal returnere den billigste madvare, hvis vægt 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 (T6).