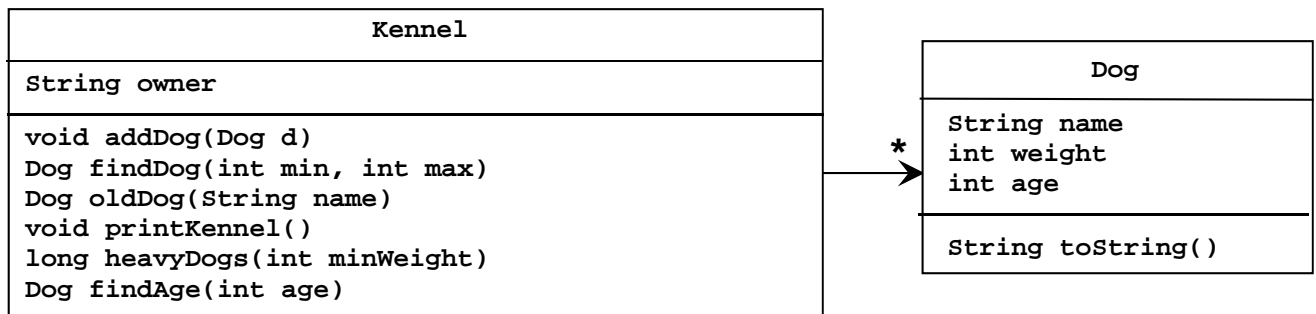


Dog



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

"Basse (15 kg and 4 years old)".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Dog*-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, *Kennel*, hvis objekter repræsenterer en kennel, der har nogle hunde. Klassen og dens relation til *Dog*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addDog*, der tilføjer *Dog*-objektet *d* til *Kennel*-objektet.
6. Opret et objekt af typen *Kennel* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Dog*-objekter hertil.
7. Programmér metoden *findDog*. Metoden skal returnere en hund, hvis vægt ligger mellem de to parametre (begge inklusive). Det kan antages, at den anden parameter er større end eller lig den første. Udvid *Dog*-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 *oldDog*. Metoden skal returnere den ældste hund, der har det angivne navn. Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printKennel*. Metoden skal udskrive ejeren af kennelen efterfulgt af alle hunde sorteret efter alder (højest til lavest). Hvis to hunde har samme alder, sorteres alfabetisk efter navn. 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 *heavyDogs*. Metoden skal returnere antallet af hunde, 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 *findAge*. Metoden skal returnere en hund, der har den angivne alder. Afprøv den skrevne metode i *test*-metoden.

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