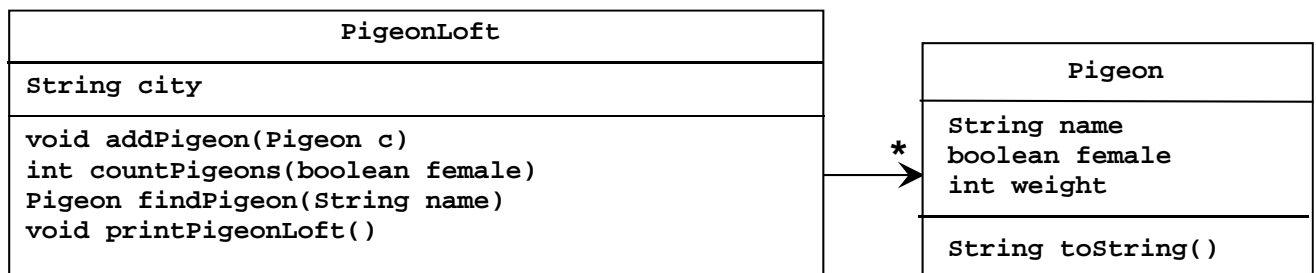


Pigeon



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

"Anna: female pigeon weighing 700 gram"

"Anton: male pigeon weighing 800 gram".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Pigeon*-objekter i *test*-metoden, via objektreferencer *p1*, *p2*, *p3*, *p4* og *p5*, 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, *PigeonLoft*, hvis objekter repræsenterer et dueslag med duer. Klassen og dens relation til *Pigeon*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addPigeon*, der tilføjer *Pigeon*-objektet *p* til *PigeonLoft*-objektet.
6. Opret et objekt af typen *PigeonLoft* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Pigeon*-objekter hertil.
7. Programmér metoden *countPigeons*. Metoden skal returnere antallet af duer med det angivne køn. Udvid *Pigeon*-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 *findPigeon*. Metoden skal returnere en due med 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 *printPigeonLoft*. Metoden skal udskrive den by, som dueslaget ligger i, efterfulgt af alle duer sorteret alfabetisk efter navn. Hvis to duer har samme navn, sorteres efter vægt (lavest til højest). Afprøv den skrevne metode i *test*-metoden.

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