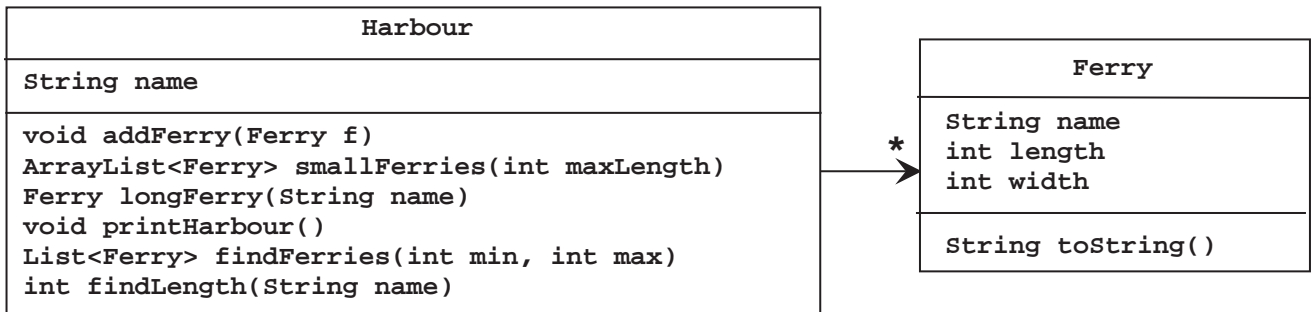


# Ferry



1. Opret en klasse, *Ferry*, hvis objekter repræsenterer en færge. 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:

**"Nautilus 200 x 50 meter".**

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Ferry*-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, *Harbour*, hvis objekter repræsenterer en færgehavn med færger. Klassen og dens relation til *Ferry*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addFerry*, der tilføjer *Ferry*-objektet *f* til *Harbour*-objektet.
6. Opret et objekt af typen *Harbour* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Ferry*-objekter hertil.
7. Programmér metoden *smallFerries*. Metoden skal returnere alle de færger, der højst har den angivne længde. Udvid *Ferry*-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 *longFerry*. Metoden skal returnere den længste færge 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 *printHarbour*. Metoden skal udskrive navnet på færgehavnen efterfulgt af alle færger sorteret alfabetisk efter navn. Hvis to færger har samme navn, sorteres efter længde (lavest til højest) og dernæst efter bredde (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 *findFerries*. Metoden skal returnere alle færger, hvis bredde 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 *findLength*. Metoden skal returnere den samlede længde af de færger, 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 (T6).**