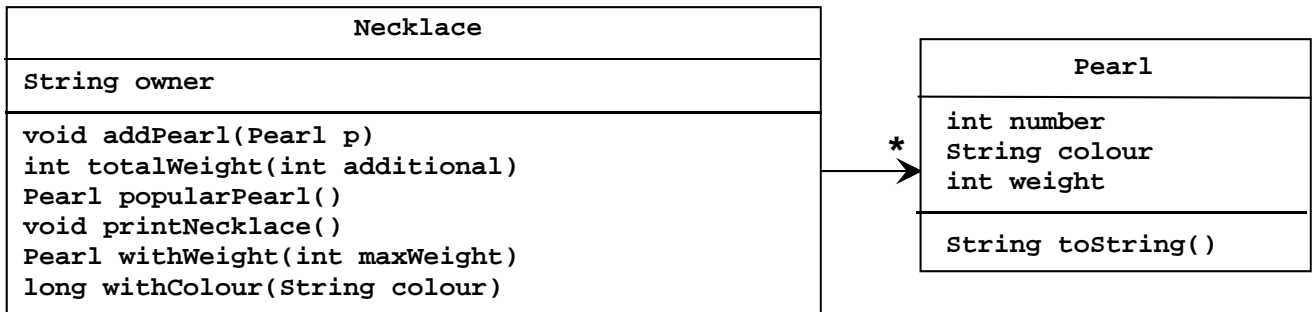


Pearl



1. Opret en klasse, *Pearl*, hvis objekter repræsenterer en type perler. Klassen er specificeret til højre i UML-diagrammet (hvor vægten angiver den enkelte perles vægt). Feltvariablerne skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en tekststreng på formen:

"5 green with weight 10 gram".

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Pearl*-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, *Necklace*, hvis objekter repræsenterer en halskæde med perler. Klassen og dens relation til *Pearl*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addPearl*, der tilføjer *Pearl*-objektet *p* til *Necklace*-objektet.
6. Opret et objekt af typen *Necklace* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Pearl*-objekter hertil.
7. Programmér metoden *totalWeight*. Metoden skal returnere halskædens vægt, idet parameteren angiver snorens og låsemekanismens totale vægt. Udvid *Pearl*-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 *popularPearl*. Metoden skal returnere den type perler, der er flest af. Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printNecklace*. Metoden skal udskrive ejeren af halskæden, efterfulgt af alle perler sorteret alfabetisk efter farve. Hvis to perler har samme farve, sorteres efter antal (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 *withWeight*. Metoden skal returnere en type perle, der højst 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 *withColour*. Metoden skal returnere antallet af perler, der har den angivne farve (hvis en type perle forekommer flere gange skal alle forekomster tælles med). Afprøv den skrevne metode i *test*-metoden.

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