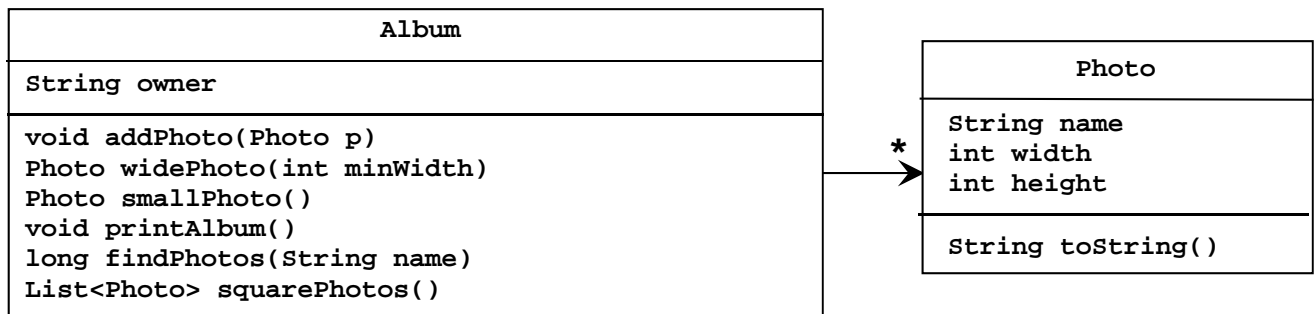


Photo



1. Opret en klasse, *Photo*, hvis objekter repræsenterer et fotografi. 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:

"Beach: 56 mm x 30 mm"

2. Lav en *TestDriver*-klasse med en klassemetode *test*. Metoden returnerer intet og har ingen parametre.
3. Opret fem velvalgte *Photo*-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, *Album*, hvis objekter repræsenterer et album med fotografier. Klassen og dens relation til *Photo*-klassen, er specificeret i ovenstående UML-diagram.
5. Programmér metoden *addPhoto*, der tilføjer *Photo*-objektet *p* til *Album*-objektet.
6. Opret et objekt af typen *Album* i *test*-metoden i *TestDriver*-klassen og knyt de allerede oprettede *Photo*-objekter hertil.
7. Programmér metoden *widePhoto*. Metoden skal returnere et billede, der mindst har den angivne bredde. Udvid *Photo*-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 *smallPhoto*. Metoden skal returnere det fotografi, der er mindst (målt som bredde gange højde) Afprøv den skrevne metode i *test*-metoden.

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

10. Programmér metoden *printAlbum*. Metoden skal udskrive albummets ejer, efterfulgt af alle fotografier sorteret alfabetisk efter navn. Hvis to fotografier har samme navn sorteres efter bredde (højest til lavest) og dernæst efter højde (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 *findPhotos*. Metoden skal returnere antallet af fotografier, 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 (T5).

12. Brug funktionel programmering til at implementere metoden *squarePhotos*. Metoden skal returnere alle de fotografier, hvor bredde og højde er identisk. Afprøv den skrevne metode i *test*-metoden.

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