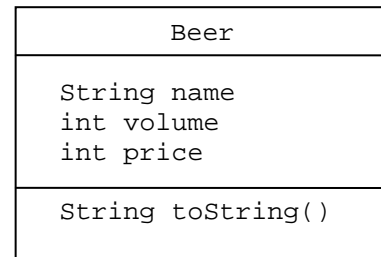


## Eksamensprojekt 1

1. Opret en klasse, *Beer*, der repræsenterer en øl; klassen *Beer* er specificeret i UML-diagrammet til højre. De tre feltvariabler skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en streng-repræsentation for en øl, f.eks.

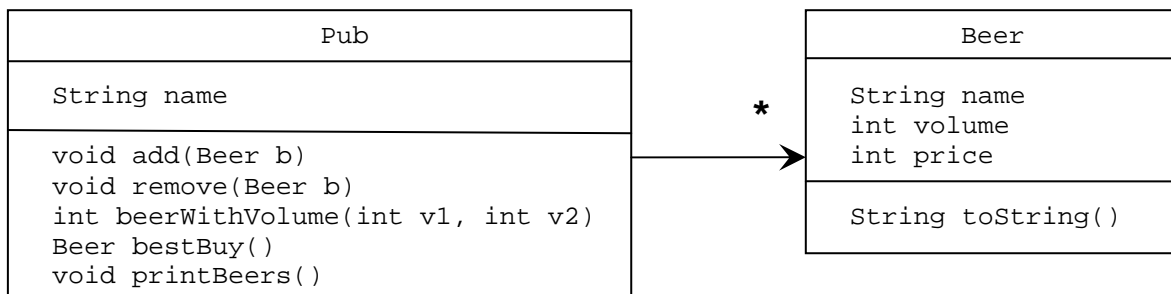
`"Carls Special, 33, 25"`



2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og ingen parametre.
3. Opret tre velvalgte *Beer*-objekter, via objektreferencer *b1*, *b2* og *b3*, i *exam*-metoden og udskriv disse vha. *toString*-metoden.

**Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.**

4. Opret en ny klasse, *Pub*, der repræsenterer en bar; klassen *Pub*, og dens relation til klassen *Beer*, er specificeret i følgende UML-diagram:



5. Programmér metoderne *add* og *remove* der henholdsvis tilføjer og fjerner *Beer*-objektet *b* til/fra *Pub*-objektet.
6. Opret et objekt af typen *Pub* i *exam*-metoden i *Driver*-klassen og knyt de allerede oprettede *Beer*-objekter hertil.
7. Programmér metoden *beerWithVolume*. Metoden skal returnere antallet af øl med en volumen på mindst *v1* og højst *v2*. Udvid *Beer*-klassen med de nødvendige get-metoder.
8. Benyt metoden *beerWithVolume* til i *exam*-metoden i *Driver*-klassen at udskrive antallet af øl for hvert af intervallerne 25-35, 36-50, og 51-100.

**Tilkald tilsynsførende og demonstrer det du har lavet indtil nu.**

9. Programmér metoden *bestBuy*. Metoden skal returnere det bedste *Beer*-objekt fra et *Pub*-objekt bedømt efter hvor meget øl man får for pengene (*volume/price*). Hvis flere øl er lige gode køb, er det underordnet hvilken af disse der returneres. Udvid *Beer*-klassen med de nødvendige get-metoder.
10. Programmér metoden *printBeers*. Metoden skal udskrive en liste af samtlige øl i en pub ordnet efter hvor meget øl man får for pengene.