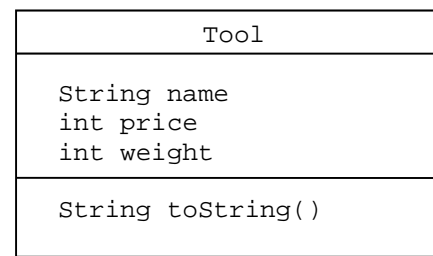


## Tool

1. Opret en klasse, *Tool*, der repræsenterer et værktøj. Klassen 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 et *Tool*-objekt på formen

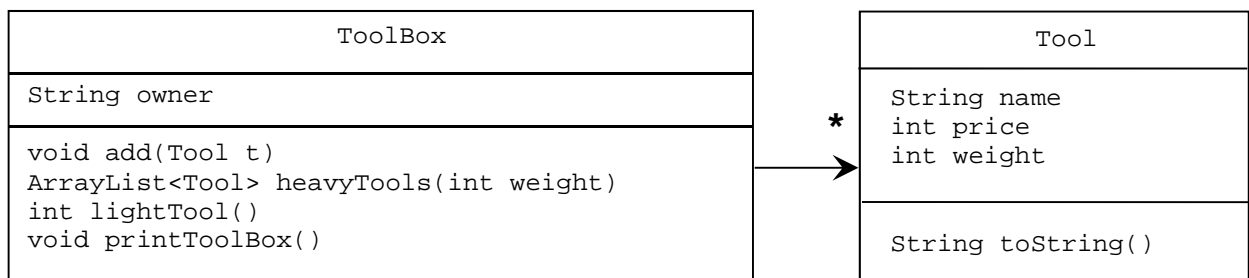
```
"hammer: 240 kr., 150 gram"
```



2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og være uden parametre.
3. Opret fem valgte *Tool*-objekter i *exam*-metoden, via objektreferencer *t1*, *t2*, *t3*, *t4* og *t5*, og udskriv disse vha. *toString*-metoden.

**Tilkald en instruktør og demonstrer det du har lavet indtil nu.**

4. Opret en ny klasse, *ToolBox*, der repræsenterer en værktøjskasse med værktøjer. Klassen *ToolBox*, og dens relation til klassen *Tool*, er specificeret i følgende UML-diagram:



5. Programmér metoden *add*, der tilføjer *Tool*-objektet *t* til *ToolBox*-objektet.
6. Opret et objekt af typen *ToolBox* i *exam*-metoden i *Driver*-klassen og knyt de allerede oprettede *Tool*-objekter hertil.
7. Programmér metoden *heavyTools*. Metoden skal returnere alle de værktøjer, der mindst har den angivne vægt. Udvid *Tool*-klassen med de nødvendige accessormetoder.
8. Afprøv metoden *heavyTools* i *exam*-metoden i *Driver*-klassen.

**Tilkald en instruktør og demonstrer det du har lavet indtil nu.**

9. Programmér metoden *lightTool*. Metoden skal returnere prisen på det værktøj, der vejer mindst. Hvis der ikke findes et sådant værktøj returneres 0. Afprøv *lightTool* i *exam*-metoden.

**Tilkald en instruktør og demonstrer det du har lavet indtil nu.**

10. Programmér metoden *printToolBox*. Metoden skal udskrive navnet på værktøjskassens ejer efterfulgt af alle værktøjer sorteret efter vægt (lavest til højest). Hvis to værktøjer har samme vægt sorteres efter pris (lavest til højest). Afprøv *printToolBox* i *exam*-metoden.

**Tilkald en instruktør og demonstrer din færdige løsning.**