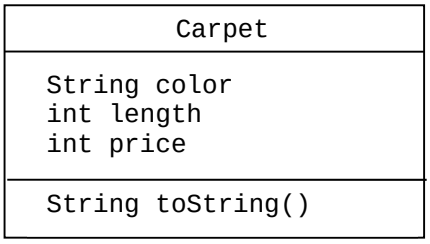
Carpet



1. Opret en klasse, *Carpet*, der repræsenterer et tæppe. 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 *Carpet*-objekt på formen

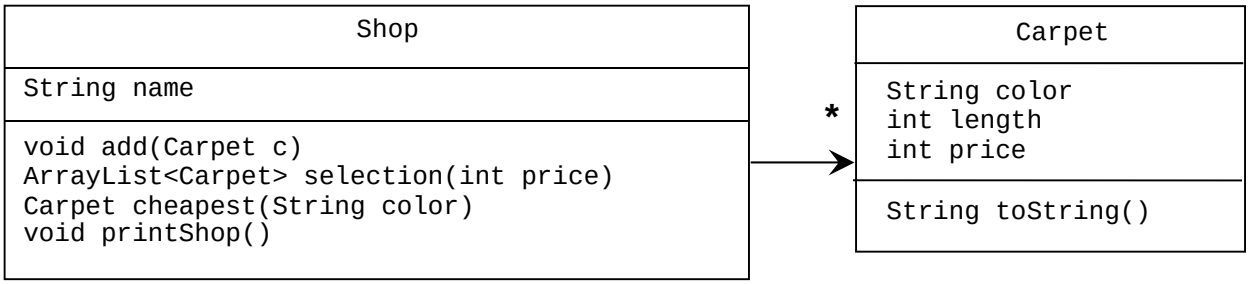
   ```
   "grey carpet, 3 meter of 250 kr."
   ```

2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og være uden parametre.

3. Opret fem velvalgte *Carpet*-objekter i *exam*-metoden, via objektreferencer *c1, c2, c3, c4* og *c5,* og udskriv disse vha. *toString*-metoden.

Tilkald en instruktor og demonstrer det du har lavet indtil nu.

4. Opret en ny klasse, *Shop*, der repræsenterer en butik med tæpper. Klassen *Shop*, og dens relation til klassen *Carpet*, er specificeret i følgende UML-diagram:



5. Programmér metoden *add,* der tilføjer *Carpet*-objektet *c* til *Shop*-objektet.

6. Opret et objekt af typen *Shop* i *exam*-metoden i *Driver*-klassen og knyt de allerede oprettede *Carpet*-objekter hertil.

7. Programmér metoden *selection*. Metoden skal returnere alle de tæpper, der højst koster den angivne pris. Udvid *Carpet*-klassen med de nødvendige accessormetoder.

8. Afprøv metoden *selection* i *exam*-metoden i *Driver*-klassen.

Tilkald en instruktor og demonstrer det du har lavet indtil nu.

9. Programmér metoden *cheapest*. Metoden skal returnere det tæppe af den angivne farve, der er billigst (pr. meter). Hvis der ikke findes en sådant tæppe returneres null. Afprøv *cheapest* i *exam*-metoden.

Tilkald en instruktor og demonstrer det du har lavet indtil nu.

10. Programmér metoden *printShop*. Metoden skal udskrive butikkens navn efterfulgt af alle tæpper sorteret efter pris (per meter) (lavest til højest). Hvis to tæpper har samme pris sorteres efter længde (højest til lavest). Afprøv *printShop* i *exam*-metoden.

Tilkald en instruktor og demonstrer din færdige løsning.