

# Carpet

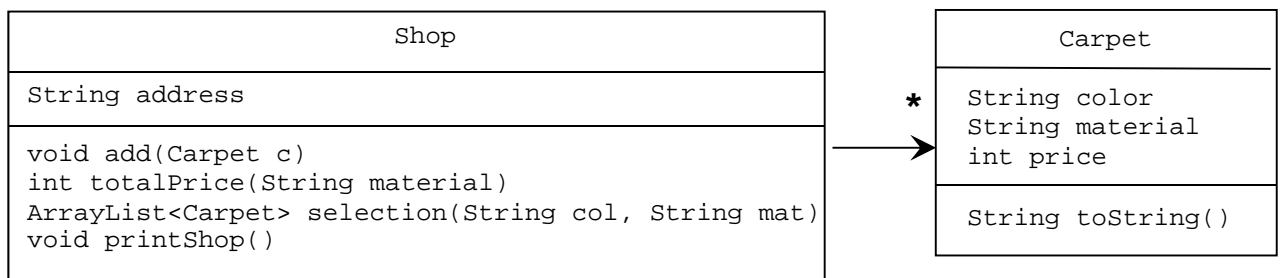
1. Opret en klasse, *Carpet*, der repræsenterer et tæppe. Klassen er specificeret i UML-diagrammet til højre. De tre feltvariabler skal initialiseres i en konstruktør (via parametre af passende type). Metoden *toString* skal returnere en streng-repræsentation for et *Carpet*-objekt på formen

"black wool: 1250 kr."

2. Lav en *Driver*-klasse med en *exam*-metode. Metoden skal være static, have returtype void og være uden parametre.
3. Opret fem valgte *Carpet*-objekter i *exam*-metoden, via objektreferencer *c1*, *c2*, *c3*, *c4* og *c5*, og udskriv disse vha. *toString*-metoden.

**Tilkald en instruktør 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 oprettede *Carpet*-objekter hertil.
7. Programmér metoden *totalPrice*. Metoden skal returnere den samlede pris for alle de tæpper, der er lavet af det angivne materiale. Udvid *Carpet*-klassen med de nødvendige accessormetoder.
8. Afprøv metoden *totalPrice* i *exam*-metoden i *Driver*-klassen.

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

9. Programmér metoden *selection*. Metoden skal returnere alle de tæpper, der har den angivne farve og er lavet af det angivne materiale. Afprøv *selection* i *exam*-metoden.

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

10. Programmér metoden *printShop*. Metoden skal udskrive butikkens adresse efterfulgt af alle tæpper sorteret alfabetisk efter materiale. Hvis to tæpper er af samme materiale sorteres efter pris (lavest til højest). Afprøv *printShop* i *exam*-metoden.

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