

Opgaver tirsdag den 7. september

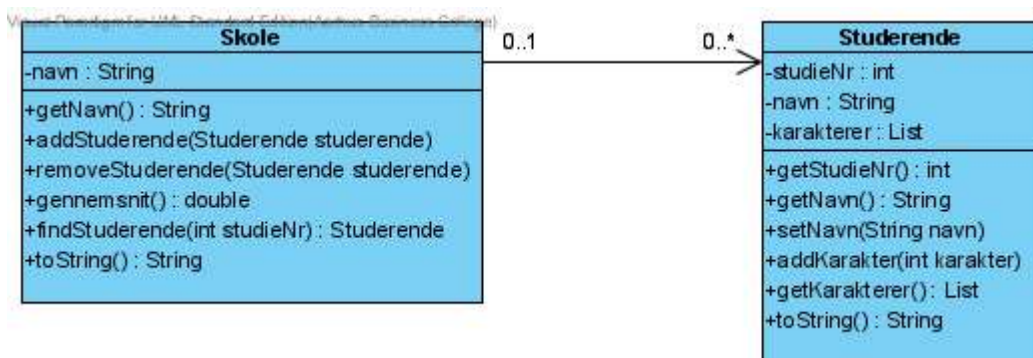
Opgave 1

Lav en klasse `AnvendMetoderPaaHashSet` – klassen skal have en `main()` metode.

- 1) Tilføj til `main` erklæring og oprettelse af en mængde baseret på et `HashSet`. Mængden skal indeholde heltal.
- 2) Indsæt tallene 34,12,23,45,67,34,98 i mængden.
- 3) Udskriv indholdet af mængden.
- 4) Indsæt tallet 23 i mængden.
- 5) Udskriv indholdet af mængden.
- 6) Fjern elementet 67 fra mængden.
- 7) Udskriv indholdet af mængden.
- 8) Undersøg om mængden indeholder elementet 23
- 9) Udskriv hvor mange elementer der er i mængden

Opgave 2

Betragt følgende designklassediagram:



Opgave 2.1

Programmer klasserne med de angivne attributter og metoder idet du tilføjer passende constructors til klasserne. Associeringen **skal** implementeres som en **List**.

- Metoden *gennemsnit* skal beregne det samlede gennemsnit af alle karakter, for de studerende på skolen.
- Metoden *findStuderende* skal returnerer en studerende med det angivne *studieNr*, hvis en sådan studerende ikke findes, skal der returneres *null*.
- Lav et test program der opretter en skole med et antal studerende. Udskriv skolen.

Opgave 2.2

Kopier klasserne over i en ny pakke i projektet og lav de ændringer der skal til, når associeringen mellem **Skole** og **Studerende** i stedet realiseres som et **Set**. Anvend den konkrete type *HashSet*.

Opgave 2.3

Kopier klasserne over i en ny pakke i projektet og lav de ændringer der skal til, når associeringen mellem **Skole** og **Studerende** i stedet realiseres som en **Map** idet *studieNr* anvendes som nøgle i mappen. Anvend den konkrete type *HashMap*.

Opgave 3

Lad klassen *Studerende* fra opgave 2 implementere *Comparable* interfacet, idet to *Studerende* opfattes som værende ens, hvis de har samme *studieNr*.

I nogle tilfælde er det hensigtsmæssigt at kunne sortere de studerende efter *navn* og ikke efter *studieNr*. Derfor skal der laves en klasse *StuderendeComparator* der implementerer *Comparator* interfacet. I denne klasse sammenlignes studerende efter deres navn. Du kan teste din *Comparator* ved at angive den i constructor til et *TreeSet*, indsætte studerende i *TreeSet* og dernæst udskrive mængden.

Opgave 4

Du har en tekst fil med karakter for film. Hver film har en karakter mellem 1 (dårlig) og 5 (fremragende). Den første linje i filen angiver hvor mange karakterer der er i filen. Hver karakter består af to linjer: Navnet på filmen og karakteren for filmen. Her er et eksempel på fire forskellige film og de 7 karakterer der er givet.

```
7
Harry Potter og de vises sten
4
Harry Potter og de vises sten
5
Min Søsters børn i Afrika
1
Harry Potter og de vises sten
4
Min Søsters børn i Afrika
2
Løvernes konge
3
Junglebogen
4
```

Skriv et program der læser en sådan fil, og beregner den gennemsnitlige karakter for hver film og udskriver gennemsnittet for hver fil med angivelse af hvor mange karakterer der er givet. Output kunne f.eks se således ud:

```
Harry Potter og de vises sten: 3 karakterer, gennemsnit 4,3
Min Søsters børn i Afrika: 2 karakterer, gennemsnit 1,5
Løvernes konge: 1 karakter, gennemsnit 3
Junglebogen: 1 karakter, gennemsnit 4
```

Anvend en eller flere *Map* for at beregne output.

Hjælp: Key værdi i *map* skal være *String* (navn på film)