Plenum: Invarianter og repetisjon

Ahmed Waseem Saeed

Mars 2024

Introduksjon

Velkommen til ny plenumstime! Denne uken skal fokusere på invarianter, men repetere en del tidligere stoff. Vi skal modellere fotballag, innebandy og en serietabell. Vi skal berøre tema som: innlesing av fil, lenkelister med prioritert innsetting, interface, samt litt rekursjon.

Måten vi skal behandle invarianter på er følgende: vi skal holde en oversikt over de ulike tilstandene til programmet vårt. Skriv derfor kommentarer som beskriver hvordan og når tilstandene endres i koden, eksempelvis i metoder, løkker og betingelser.

1 ErIdrettslag

Interfacet ErIdrettslag skal bestemme oppførselen til et gitt idrettslag. Vi ønsker at alle idrettslag har følgende attributter: navn, år laget var grunnlagt, samt antall poeng (for en gitt sesong).

Interfacet ser slik ut:

```
interface ErIdrettsLag {
    String hentNavn();

    int hentAar();

    int hentAntallPoeng();
}
```

Men, vi er ikke ferdig. Sørg for at interfacet implementerer interfacet Comparable, der du skal kunne sammenligne andre objekter (referanser egentlig, objektene har en annen type) av typen ErIdrettslag (vi skal bruke dette senere for å legge til sorterte lag basert på poeng). Husk: hvordan kan et interface implementere et annet interface?

2 Fotballag

Skriv klassen Fotballag, som implementerer interfacet ErIdrettslag. La interfacet spesifisere oppførselen.

3 Innebandylag

Skriv klassen Innebandylag, som implementerer interfacet ErIdrettslag. La interfacet spesifisere oppførselen.

Reflektér: Hvorfor bruker vi interface her fremfor arv (kunne hatt en abstrakt klasse Lag, som de andre arver fra)?

4 SerieSystem

Før vi begynner å skrive klassen SerieSystem, skal vi opprette en enum-type vi kaller for SerieType, som definerer hvilke typer serier vi kan ha. Akkurat nå kan du legge til FOTBALL og INNEBANDY.

Skriv klassen SerieSystem. Den har en liste over objekter av typen ErIdrettslag, samt et navn som beskriver serien (eks: "Eliteserien fotball 2024"). Klassen skal også ta inn et filnavn, som referer til en fil med innhold om lagene i serien. Se plenumssiden for eksempelfiler (fotball.txt og innebandy.txt). Til slutt må klassen vite hvilken SerieType den er (tas inn i konstruktøren). Klassen må ha en metode *beregnTotalPoengSum()* som skal returnere antall poeng summert for lag i serien.

I tillegg skal klassen ha en private metode lesFraFil() som skal lese inn data om ulike lag, der hver linje inneholder tre felter skilt med komma: navn på laget, året det ble stiftet, samt antall poeng på nåværende stadie. Bruk serietypen til å avgjøre hvilket objekt du skal legge til i listen din over lagene i serien. Kall på denne metoden i konstruktøren til klassen.

4.1 Innføre prioritet basert på poeng

Et seriesystem er ganske ubrukelig dersom vi ikke kan sortere lagene basert på poeng. For å håndtere dette skal vi nå benytte en prioritert lenkeliste (en ganske simpel en, sammenlignet med det dere har skrevet i tidligere innleveringer).

"Nodene" i vår lenkeliste blir objekter av typen ErIdrettslag. Vi skal ha en enkeltlenket liste. Legg til en metodesignatur i interfacet ErIdrettslag med følgende metode: *ErIdrettslag hentNeste()*, som skal returnere neste-pekeren for en gitt node.

4.1.1 Comparable

Implementer metoden compareTo() for både Fotballag- og Innebandylag-klassene, som sammenligner to objekter basert på poeng.

4.1.2 PoengPrioLenkeliste

Skriv klassen PoengPrioLenkeliste. Klassen har kun en start-peker av typen ErIdrettslag som initieres til null. Skriv deretter en metode leggTil(ErIdrettslag lag) som skal sette inn et lag på riktig plass i lenkelisten, basert på antall poeng. Skriv først metoden iterativt. Deretter skal du skrive metoden ved hjelp av rekursjon.

HUSK: tegn, tegn, tegn, og ... tegn! Husk også at det er to ting vi må huske på ved rekursive metoder, hvilke?

Utvid nå SerieSystem med en ny instansvariabel: sortertListe som er av typen PoengPrioLenkeliste. Modifiser lesFraFil til å kalle på leggTil i sortertListe-variabelen. Skriv også en metode skrivSortert() som skriver ut innholdet i en den sorterte lista. Dette krever at du skriver en ny metode i PoengPrioLenkeliste, *skrivUt()* som skriver ut innholdet i lenkelisten ved hjelp av en while-løkke (eller rekursjon :D). Kall på metoden i SerieSystem og verifiser at metoden skriver ut innholdet på en riktig måte.