



Øving 8

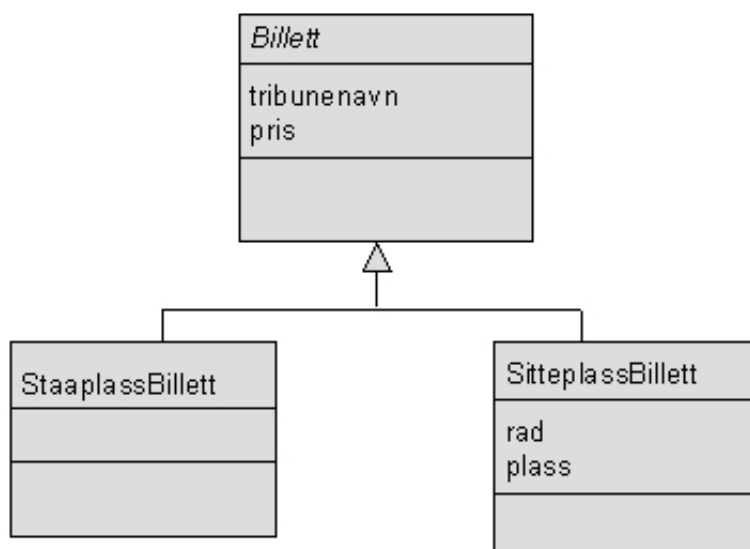
TDAT1005 Databaser med videregående programmering
Institutt for datateknologi og informatikk (IDI), NTNU

Arv og polymorfi 2

I denne øvingen skal du jobbe med en eksamensoppgave i Videregående programmering fra mai 2002. Oppgaven telte 60 % av en 4 timers eksamen.

Du skal jobbe med et system for salg av billetter til et sportsarrangement med ulike tribune- og billett typer.

Vi begynner med billettene. Det selges to typer billetter, ståplassbilletter og sitteplassbilletter. Dette er modellert som vist i figur 1. Operasjoner er ikke vist i diagrammet. Disse klassene får dere ferdig programmert: [Billett.java](#) (studentene fikk ikke dette på eksamen, men de trengte ikke å programmere dem heller).



Figur 1: Klassediagram over billetter

Nå går vi over til å se på tribunenene. Man har ståtribuner og sittetribuner. Alle tribuner har et navn, en kapasitet og en pris. Sittetribuner har i tillegg et antall seterader. Vi antar at alle radene er like lange, slik at antall seter

pr. rad er kapasiteten delt på antall rader.

For ståtribunene trenger man bare å holde rede på hvor mange plasser som er solgt, mens på sittedribunene må man vite hvor mange som er solgt på hver rad. Vi antar at plassene selges i stigende rekkefølge innenfor hver rad. I tillegg antar vi at programmet, ikke kjøperen, bestemmer hvilken rad som skal benyttes, noe du vil få bruk for i oppgave c.

Noen sittedribuner er VIP-tribuner. Der vil vi også holde rede på navnet på de som skal sitte på de ulike plassene.

Oppgave A:

Tribuneklassene skal ha følgende objektvariabler (ikke gitt i eksamensoppgaven, men gis her pga at du enda ikke er kommet så langt som til eksamen):

Klassen Tribune:

```
private final String tribunenavn;  
private final int kapasitet;  
private final int pris;
```

Klassen Staa:

```
private int antSolgteBilletter;
```

Klassen Sitte:

```
private int [] antOpptatt; // tabellstørrelse: antall  
rader
```

Klassen VIP:

```
private String[][] tilskuer; // tabellstørrelse:  
antall rader * antall plasser pr rad
```

Tegn et klassediagram tilsvarende figur 1, for tribune-klassene. Les hele oppgaveteksten før du tegner, og oppdater også figuren underveis dersom du finner ut at det trengs. Du skal totalt ha 4, eventuelt 5, tribuneklasser.

I det følgende antar vi at alle objektvariabler er private. Konstruktører og tilgangsmetoder må du programmere etter hvert som du trenger dem.

Oppgave B:

For alle tribunetyper skal vi ha metodene finnAntallSolgteBilletter() og finnInntekt(). Inntekten er prisen multiplisert med antall solgte billetter.

Programmer dette.

(Tips: Begge de nevnte metodene skal eksistere i klassen Tribune.)

Oppgave C:

Nå skal du programmere billett kjøp. Dette skal gjøres ved å lage to metoder (i tribune-klassene) som begge heter kjøpBilletter(), og som returnerer en tabell av billett-objekter. Den første metoden tar som parameter antall billetter som ønskes, den andre tar som parameter en tabell av tekststrenger (navnet på de som skal ha billettene). For alle tribunetyper unntatt VIP-tribuner, skal metode to gi samme resultat som metode en, mens for VIP-tribuner skal metode en returnere null (ingen får kjøpe billett på VIP-tribuner uten å oppgi navn). For alle tribunetyper gjelder at hvis man ønsker seg flere billetter enn det som kan effektueres, får man ingen. På sittetribuner begrenses dette av at alle billetter som selges i en bestilling, skal være på samme rad. På ståtribuner er det bare den totale kapasiteten, og hvor mange som hittil er solgt, som setter begrensninger.

(Tips: Begge kjøpBilletter()-metodene skal eksistere i klassen Tribune.)

Oppgave D:

Lag et lite testprogram der du oppretter to ståtribuner, en vanlig sittetribune og en VIP-tribune. Samle tribunene i en tabell av typen Tribune[].

Kjøp billetter på alle tribunene, og skriv ut billettene.

For hver tribune skal du også skrive ut tribunenavn, kapasitet, antall solgte billetter og inntekten. (Du kan gjerne la en toString()-metode i klassen Tribune gjøre jobben.)

Oppgave E (repetisjon, men ikke frivillig):

1. Vis hvordan vi sorterer Tribune-tabellen etter inntekt ved å bruke en metode i klassen Arrays.
2. Lagre alle tribuneobjektene på fil. Les dem inn fra fil igjen og skriv ut.

Siden dette er en eksamensoppgave, så vil det være noen metoder du ikke får beskjed om å programmere i oppgavene. Du skal levere inn et kjørbart program, så hvis du mener det mangler noen metoder så programmer disse også.

Institutt for datateknologi og informatikk (IDI), NTNU

