

## **E K S A M E N**

<b>Emnekode:</b>	<b>IS202</b>
<b>Emnenavn:</b>	<b>Emner i programmering</b>
Dato:	16. desember 2004
Varighet:	0900 - 1300
Antall sider inkl. forside:	3
Målform:	Norsk
Tillatte hjelpemidler:	Alle trykte og skrevne.
Merknader:	Les gjennom hele oppgaven før du begynner å svare. Oppgavene bør betraktes som en helhet.

Det kan være enklere å besvare oppgave 1 og 2 ved å skrive komplette klassesdeklarasjoner i stedet for å besvare spørsmålene individuelt. Bruk i tilfelle kommentarer til å vise hvor svarene på enkeltoppgavene kan finnes.

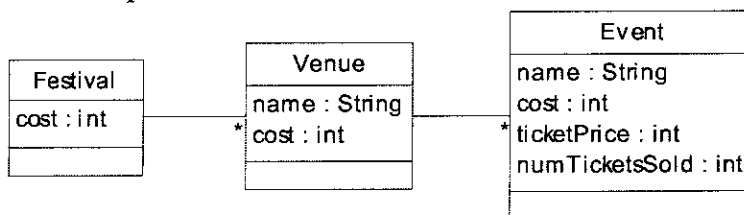
---

Musikkfestivalen må unngå å tape penger for å overleve. Helst bør den gå med overskudd slik at den kan bygge opp et fond som kan brukes til å komme gjennom dårlige år. Festivalledelsen er ikke spesielt sterke på økonomi, men de har besluttet at de trenger et program som kan gi dem enkle økonomirapporter. I denne oppgaven skal vi se på hvordan vi kan lage disse rapportene.

Det de er interessert i å vite er om festivalen går med overskudd, og hvor mye hver konsert (Event) og hver scene (Venue) bidrar til overskuddet. Den tilgjengelige informasjonen er:

- antall billetter som er solgt til hver konsert (numTicketsSold).
- billettprisen for hver konsert (ticketPrice)
- kostnader som kan knyttes direkte til hver konsert (f.eks. betale artisten) (cost)
- kostnader forbundet med scenen (f.eks. leie, og forberedelser til festivalen) (cost)
- kostnader som er felles for hele festivalen, og ikke kan knyttes til en bestemt konsert eller scene (f.eks. markedsføring) (cost)

Dette kan representeres med denne entitetsmodellen:



### Oppgave 1 – Beregne overskudd (teller 40 %)

- Skriv klassesdeklarasjoner for Festival, Venue and Event.
- Skriv metoden `getIncome()` i Event. Den skal beregne nettoinntekt (dekningsbidrag) fra en konsert.
- Skriv metoden `getIncome()` i Venue. Den skal beregne nettoinntekt (dekningsbidrag) fra en scene. (dvs. inntektene fra konsertene minus kostnadene knyttet til konserter og scene)
- Skriv metoden `getProfit` i Festival. Den skal beregne overskuddet (eller underskuddet) for hele festivalen.

### Oppgave 2 – Finne problemer (teller 30 %)

Nå har vi et program som beregner overskudd fra festivalen. I tillegg trenger festivalledelsen en funksjon som gir dem oversikt over konserter som ikke bidrar til overskuddet, dvs. at inntektene fra konserten er mindre enn direkte kostnader.

Skriv metoden `getPoorEvents()` i Festival. Den skal returnere en Vector som inneholder alle konsertene med negativt dekningsbidrag. Du kan legge til hjelpemetoder i Festival eller du andre klassene hvis du trenger det.

### **Oppgave 3 – Finne konserter (teller 30 %)**

Festivalledelsen ønsker et system for å se og redigere konsertdata. For å få gode responstider må det gå raskt å finne data om en bestemt konsert.

- a) Anta at vi beholder datastrukturen fra forrige oppgave. Da er tiden vi trenger for å finne en konsert når vi vet navnet proporsjonal med antall scener. Forklar hvorfor det er slik.
- b) Vi kan få bedre responstider hvis vi har pekere til Event-objektene i Festival-objektet. Hva slags datastruktur bør vi bruke? Begrunn svaret. (Vi er bare interessert i raske oppslag. Andre operasjoner som innsetting og fjerning er ikke viktige)
- c) Kan vi bruke klasser fra standardbiblioteket i Java? Hvilke?

# E K S A M E N

<b>Emnekode:</b>	<b>IS202</b>
<b>Emnenavn:</b>	Programming Topics
<b>Dato:</b>	16. December 2004
<b>Varighet:</b>	0900 - 1300
<b>Antall sider inkl. forside:</b>	3
<b>Målform:</b>	English
<b>Tillatte hjelpemidler:</b>	Any printed or written material.
<b>Merknader:</b>	Please read all of the following text before you start writing. The problems given below are related, each asking for a part of a larger program.  You may solve problems 1 and 2 by writing complete class declarations rather than responding to the individual questions. Please use comments to indicate where you have answered each question.

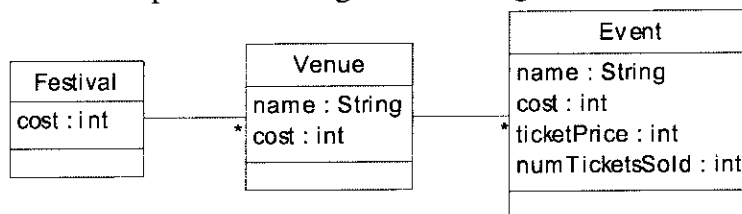
---

To ensure the survival of the Rock Festival, it must avoid losing money. Preferably the festival should give a profit, so funds can be built up to get through the occasional bad year. The festival management doesn't know too much about economy and all that stuff, so they have decided that they should start by getting some simple reports. In this problem set we work on a program to provide those reports.

Basically the management wants to know if the festival is making a profit, and how much the venues and events are contributing to the profit. The available information is:

- the number of tickets sold to each event
- the ticket price for each event
- the cost directly attributable to each event (e.g. paying the performer(s))
- the cost of running each venue (e.g. rent paid and preparing it for the festival)
- costs that cannot be connected to any specific venue or event (e.g. marketing)

This can be represented using the following entities:



### Problem 1 – Rock Festival Profit (counts 40 %)

- a) Write the class declarations for Festival, Venue and Event.
- b) Write a method `getIncome()` in Event that calculates the net income generated by an event.
- c) Write a method `getIncome()` in Venue that calculates the net income generated by the events at that venue. (ie. the income from the events minus the cost of running the events and venue costs)
- d) Write a method `getProfit` in Festival that calculates the total profit (or loss) generated by the festival.

### Problem 2 – Find the culprits (counts 30 %)

Now we have a program that will calculate the profit of the festival. In addition the festival management need to find the events that do not contribute to the economic result of the festival, that is the events with no net income.

Write a method `getPoorEvents()` in Festival, which returns a Vector containing the events with negative net income. You may add helper methods in Festival or the other classes.

### **Problem 3 – Event lookup (counts 30 %)**

The festival management wants a system that gives fast access to the data about individual events. This is necessary if we want good response times in an application for viewing and editing the event data.

- a) Assume that we keep the structure from the previous problems. Then the time needed to find an event by name is proportional to the number of venues. Explain why.
- b) We may get better response times if we store pointers to the event objects in the festival object. What is a good data structure to use for this? Why? (We are only interested in fast lookup. Other operations e.g. insert and delete are unimportant).
- c) Is there a class in the Java standard library you can use? If so, which one?