



## Øving 19

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

---

# Lagvis databaseprogrammering med transaksjonshåndtering

Du skal arbeide med deler av et utlånssystem for et bibliotek. Her er et [script](#) med oversikt over boktitler og eksemplarer. Merk at informasjon om den enkelte tittelen ligger i tabellen boktittel, mens oversikt over de konkrete eksemplarene fins i tabellen eksemplar. En låntaker låner alltid ett bestemt eksemplar og registreres derfor i eksemplar-tabellen.

*Kanskje du trenger du å repetere litt JDBC-programmering før du begynner på denne øvingen? Se mappen fra workshopen (DP5). Der finner du bl.a. en frivillig øving med løsning. Den øvingen bruker de samme databasetabellene som du skal bruke her. - Hvis du trenger å repetere enda mer, se leksjonen som du finner øverst i DP5-mappen.*

Videre til dagens øving:

Klassen [Bok](#) er gitt.

Du skal lage klassen **Database** med følgende innhold:

### Konstruktør

Her skal en databaseforbindelse settes opp. Dette skal være en objektvariabel som kan brukes i alle metodene i klassen.

### **public void lukkForbindelse()**

Denne metoden skal lukke forbindelsen som ble satt opp i konstruktøren.

***For hver av metodene nedenfor skal du vurdere behovet for transaksjonshåndtering og gjennomføre det, der det er nødvendig (se boka side 848-851 og leksjon publisert under DP18). Med dette menes at du skal slå av autocommit, og bruke commit() dersom alt går bra og rollback() dersom det ikke går bra. Husk å slå på autocommit til slutt.***

Husk også alltid å frigjøre databaseressurser (legges i finally-delen i en try-catch-finally-setning, eventuelt Java7-varianten).

I SQL-setningene nedenfor er data som skal ligge i variabler skrevet i *kursiv* og omsluttet av <>, eksempel <isbn>.

### **public boolean regNyBok(Bok nyBok)**

Denne metoden skal registrere en ny tittel og samtidig eksemplar nummer 1 av denne tittelen.

SQL-setninger:

```
insert into boktittel(isbn, forfatter, tittel)
values (<isbn>, <forfatter>, <tittel>)
insert into eksemplar(isbn, eks_nr) values (<isbn>, 1);
```

Metoden skal returnere **false** dersom bok med dette ISBN er registrert fra før. Da skal ingen oppdatering skje.

### **public int regNyttEksemplar(String isbn)**

Denne metoden skal registrere et nytt eksemplar av en tittel som allerede skal være registrert i databasen. Ekseplarnummeret skal være 1 høyere enn hittil største eksemplarnummer. For å være sikret dette, kan ikke sekvenser/autonummerering brukes. (Du har ikke noen garanti for at slike sekvenser er ubrutt). Du må i stedet bruke teknikken som er brukt i metoden **registrerNyPerson()**, se leksjonen, eventuelt 4.utgave av boka, side 829.

Metoden skal returnere 0 hvis ingen tittel med dette ISBN eksisterer, ellers skal metoden returnere eksemplarnummeret.

SQL-setning:

```
insert into eksemplar(isbn, eks_nr) values (<isbn>,
<eks_nr>);
```

### **public boolean lånUtEksemplar(String isbn, String navn, int eksNr)**

Denne metoden skal registrere at ei bok er utlånt. Kolonnen *utlånt* for det aktuelle eksemplaret skal settes lik navnet til låneren. Du trenger ikke ta hensyn til om dette feltet allerede er utfyllt (vi har ingen metode for å returnere ei bok).

Metoden skal returnere **false** dersom ISBN og/eller eksemplarnummer er ugyldig. (Tips: Det er nok å sjekke at update-setningen returnerer 0, det vil si at ingen rader er oppdatert).

update eksemplar set laant\_av = <navn> where isbn = <isbn> and eks\_nr

= <eks\_nr>

**Lag et enkelt testprogram.**

Lykke til!

---

Institutt for datateknologi og informatikk (IDI), NTNU