INFO233: Obligatorisk oppgave 3

Innleveringsfrist: 18. april, kl. 14:00

Grafiske brukergrensesnitt og databaseintegrering

I denne obligatoriske oppgaven så skal vi se på grafiske grensesnitt og integrasjon mot databaser. Dere kommer til å jobbe med JavaFX og Sqlite. Oppgaven er å implementere et salgsregisteringssystem. Systemet skal kunne registere personer, bedrifter, produkter og faktura. I denne oppgaven så kan du bruke alle deler av klassebiblioteket som følger med Java.

Legg ved det du blir bedt om å forklare, gjøre rede for, eller analysere i en egen fil. Les gjennom alle oppgavene før du begynner. **Legg spesielt merke til deloppgave 6**. Det er ikke nødvendig å gjennomføre deloppgavene i den samme rekkefølge de er oppført.

For å bestå oppgaven må du oppnå minst 80/100 poeng og ha svart på deloppgave 6.

Om du ikke skulle bestå så tilbyr vi også egenretting. Du kvalifiserer til egenretting ved å oppnå minst 40/100 poeng.

Deloppgaver

Hver deloppgave beskriver hva som skal gjøres og hvor mange poeng som kan oppnås. En delvis implementasjon gir ikke full pott, men kan gi en redusert poengmengde hvis implementasjonen er i riktig retning.

Deloppgave 1) – 10 poeng

Implementer en main-metode som bruker den vedlagte sql-schema filen til å opprette en ny database med sqlite. Hvis databasen allerede er opprettet så skal den ikke kjøre filen en gang til. Legg merke til at hvis schema-filen blir kjørt flere ganger så vil enkelte av datapunktene bli lagt til flere ganger i tabellene.

> NB: Det er ikke nødvendig å bruke den vedlagte schemafilen, du kan også implementere ditt eget schema, eller endre den vedlagte filen,om du skulle ønske det. Legg da ved en schemafil. Schemaet du lager må inneholde minst den informasjonen som er i den vedlagte filen.

Deloppgave 2) – 30 poeng

Implementer et vindu som kan vise en faktura. Dvs at den viser informasjonen til kunden, fakturadato, produkter som er kjøpt, og totalsum. Fakturaen kan gjerne følge formatet til (en forenklet versjon av) en vanlig norsk faktura. Altså, dere viser den informasjonen som er tilgjengelig for en faktura i databasen og den dere kan regne ut.

Deloppgave 3) – 25 poeng

Gjør det mulig å legge til nye kunder, addresser, produkter, produktkategorier, og fakturaer via et grafisk grensesnitt. Legg til noen elementer av hver type. Legg også ved en eksempel databasefil når du leverer oppgaven.

Deloppgave 4) – 25 poeng

Gjør det mulig å endre på kunder, addresser, produkter, produktkategorier, og fakturaer via et grafisk grensesnitt.

Deloppgave 5) – 10 poeng

Gjør det mulig å bla i de ulike typene, f.eks. produkter pr. produktkategori, faktura pr. kunde, og kunder via et grafisk grensesnitt.

Deloppgave 6) – 10 poeng

Reflekter over designmalene du har brukt i oppgaven. Hvilken designmaler har du brukt? Hvordan har du brukt dem? Hvorfor har du brukt dem? Hvordan er de nyttig? Er det andre designmaler du kunne ha brukt som kunne gjort utviklingen enklere? Det er viktig at du svarer på alle disse spørsmålene.

OBS: Det er obligatorisk å gjennomføre denne oppgaven.

Ekstraoppgaver

Ekstraoppgave 1 – 10 peong

Se over javafx.scene.chart og implementer 3 forskjellige diagrammer som viser ulike salgsrapporter fra databasen. F.eks. inntekt pr. kategori.

Database-schema

```
CREATE TABLE IF NOT EXISTS address (
       address_id INTEGER PRIMARY KEY,
       street_number TEXT,
       street_name TEXT,
       postal_code TEXT,
       postal_town TEXT,
       UNIQUE(street_number, street_name, postal_code, postal_town)
);

CREATE TABLE IF NOT EXISTS customer (
       customer_id INTEGER PRIMARY KEY,
       customer_name TEXT,
       address INTEGER,
       phone_number TEXT,
       billing_account TEXT,
       UNIQUE(billing_account),
       UNIQUE(phone_number),
       FOREIGN KEY(address) REFERENCES address(address_id)
);

CREATE TABLE IF NOT EXISTS category (
       category_id INTEGER PRIMARY KEY,
       category_name TEXT,
       UNIQUE(category_name)
);

CREATE TABLE IF NOT EXISTS product (
       product_id INTEGER PRIMARY KEY,
       product_name TEXT,
       description TEXT,
       price REAL,
       category INTEGER,
       FOREIGN KEY(category) REFERENCES category(category_id)
);

CREATE TABLE IF NOT EXISTS invoice (
       invoice_id INTEGER PRIMARY KEY,
       customer INTEGER,
       dato TEXT,
```

```
        FOREIGN KEY(customer) REFERENCES customer(customer_id)
);

CREATE TABLE IF NOT EXISTS invoice_items (
        invoice INTEGER,
        product INTEGER,
        FOREIGN KEY (invoice) REFERENCES invoice(invoice_id),
        FOREIGN KEY (product) REFERENCES product(product_id)
);

INSERT OR IGNORE INTO address (
        address_id,
        street_number,
        street_name,
        postal_code,
        postal_town
) VALUES (
     1,
     "6",
     "Fosswinckelsgate",
     "5007",
     "Bergen"
);

INSERT OR IGNORE INTO customer  (
        customer_id,
        customer_name,
        address,
        phone_number,
        billing_account
) VALUES (
  1,
  "Institutt for InfoMedia",
  1,
  "+47 55 58 91 00",
  "706741409023"
);

INSERT OR IGNORE INTO category (
        category_id,
        category_name
```

```
) VALUES (
       1,
       "books"
);

INSERT OR IGNORE INTO product (
       product_id,
       product_name,
       description,
       price,
       category
) VALUES (
  1,
  "Structure and interpretation of computer programs",
  "Book about programming",
  499.00,
  1
);

INSERT OR IGNORE INTO invoice (
       invoice_id,
       customer,
       dato
) VALUES (
  1, 1, "04.04.2018"
);

INSERT OR IGNORE INTO invoice_items (
   invoice,
   product
) VALUES (
  1, 1
);
```