

E K S A M E N

Emnekode: IS-202
Emnenavn: Programmeringsrelaterte emner

Dato: 28 februar 2008
Varighet: 0900-1300

Antall sider inkl. forside: 5

Målform: Norsk

Tillatte hjelpemidler: Alle trykte og skrevne

Merknader: Les alle oppgavene og tenk gjennom hvordan du vil løse dem før du begynner å skrive

Oppgave 1

Hva er hensikten med programmet ant? Forklar hva det kan gjøre, og hvorfor og hvordan det blir brukt. (teller 25 %)

Oppgave 2

Du arbeider med en enkel nettbankløsning for Krek & bandittkassen, og mangler bare servleten som registrerer betalingsoppdragene.

Vedlegg 1 inneholder en skisse til en servlet som registrer overføringer mellom konti. Servleten bruker tre databasetabeller: BANKKUNDE, KONTO og BETALINGSOPPDRAG, som er beskrevet i vedlegg 3.

Et eksempel på en html form som bruker servleten er vist i vedlegg 2. Følgende parametere blir sendt til servleten:

- KUNDENR – brukerens kundenr
- PASSORD – brukerens passord
- FRAKONTO – nummeret til kontoen penger skal overføres fra
- TILKONTO – nummeret til kontoen penger skal overføres til
- BELOP – beløpet som skal overføres
- DATO – når beløpet skal overføres

Servleten må først sjekke at alle parametre har fornuftige verdier. Deretter må den sjekke at passordet er korrekt, at frakonto tilhører brukeren og at tilkonto finnes. Hvis alle betingelser er oppfylt skal den legge inn overføringen som en ny rad i tabellen BETALINGS_OPPDRAG.

- a) Skriv ferdig metoden `kontrollerOppdrag()`. Metoden skal returnere true hvis riktig passord er oppgitt, begge kontoene eksisterer, og frakonto tilhører kunden.
(teller 25 %)
- b) Skriv ferdig metoden `registrerOppdrag()`. Metoden skal registrere oppdraget i tabellen BETALINGSOPPDRAG.
(teller 25 %)
- c) Skriv ferdig metoden `behandleOppdrag()`. Metoden skal gjennomføre hele registreringen av betalingsoppdrag.
(teller 25 %)

Vedlegg 1 – Klassen RegistrerOppdragServlet

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet som registrerer betalingsoppdrag. Alle http request blir
 * behandlet av metoden behandleOppdrag().
 */
public class RegistrerOppdragServlet extends HttpServlet {

    /** POST request sendes videre til behandleOppdrag */
    protected void doPost(HttpServletRequest req,
                          HttpServletResponse resp) {
        behandleOppdrag(req, resp);
    }

    /** GET request sendes videre til behandleOppdrag */
    protected void doGet(HttpServletRequest req,
                        HttpServletResponse resp) {
        behandleOppdrag(req, resp);
    }

    /**
     * Registrerer et betalingsoppdrag. Hvis parametere mangler eller har
     * ugyldige verdier blir oppdraget avvist. Det blir også avvist hvis
     * ikke korrekt kundenr og passord for frakonto er oppgitt.
     */
    protected void behandleOppdrag(HttpServletRequest req,
                                    HttpServletResponse resp) {

        /* Oppgave 2c */
    }

    /**
     * Hjelpemetode som sjekker at riktig kundenr og passord er oppgitt
     * for frakonto, og at tilkonto eksisterer.
     * @param kundenr
     * @param passord
     * @param frakonto kontonummer
     * @param tilkonto kontonummer
     * @return true hvis betingelsene er oppfylt, false ellers
     */
    private boolean kontrollerOppdrag(String kundenr, String passord,
                                      String frakonto, String tilkonto) {

        /* Oppgave 2a */
        return false;
    }

    /**
     * Hjelpemetode som registrerer oppdrag i databasen.
     * @param frakonto kontonr som skal belastes
     * @param tilkonto kontonr som skal godskrives
     * @param belop beløp som skal overføres
     * @param dato oppdragsdato
     *
     * @return true hvis oppdraget ble registrert,
     * false hvis noe gikk galt.
     */
    private boolean registrerOppdrag(String frakonto, String tilkonto,
                                     String belop, String dato) {

        /* Oppgave 2b */
        return false;
    }
}
```

```

/**
 * Hjelpemetode som du kan bruke til utskrift av svaret.
 * @out PrintWriter objekt som siden skal skrives til.
 * @param melding melding som skal vises på siden.
 */
private void skrivSide(PrintWriter out, String melding) {
    /* Anta at denne metoden skriver ut html-koden for siden som
     * skal vises.
     * Du trenger IKKE skrive ferdig denne metoden.*/
}
}

```

Vedlegg 2 – BekreftOppdrag.html

```

<html>
<body>

    <h1>Bekreft oppdrag</h1>

    <form action="https://www.krek.bandittkassen.no/registreropdrag"
          method="POST">

        <label for="KUNDENR">Kundenr:</label>
        <input name="KUNDENR" type="text"><br>

        <label for="PASSWORD">Passord:</label>
        <input name="PASSWORD" type="password"><br>

        <label for="FRAKONTO">Fra konto:</label>
        <input name="FRAKONTO" type="text"><br>

        <label for="TILKONTO">Til konto:</label>
        <input name="TILKONTO" type="text"><br>

        <label for="DATO">Oppdragsdato:</label>
        <input name="DATO" type="text"><br>

        <label for="BELOP">Beløp:</label>
        <input name="BELOP" type="text"><br>

        <input type="submit" value="Registrer oppdrag">
    </form>
</body>
</html>

```

Vedlegg 3 – Database og tabeller

DBMS info

Servernavn: krek.bandittkassen.no

Portnummer: 1521

SID: kbk

Brukernavn: krek

Passord: banditt

```
CREATE TABLE bankkunde (  
  kundenr NUMBER NOT NULL,  
  passord VARCHAR2(32),  
  navn VARCHAR2(128),  
  adresse VARCHAR2(128),  
  
  CONSTRAINT bankkunde_pk PRIMARY KEY (kundenr)  
);  
  
CREATE TABLE konto (  
  kontonr VARCHAR2(32) NOT NULL,  
  saldo NUMBER,  
  eier NUMBER,  
  sperret NUMBER,  
  
  CONSTRAINT konto_pk PRIMARY KEY (kontonr),  
  CONSTRAINT konto_fk FOREIGN KEY (eier) REFERENCES bankkunde (kundenr)  
);  
Sperret har verdien 1 hvis kontoen er sperret, 0 hvis den ikke er det.  
  
CREATE TABLE betalingsoppdrag (  
  oppdragsnr NUMBER NOT NULL,  
  frakonto VARCHAR2(32) NOT NULL,  
  tilkonto VARCHAR2(32) NOT NULL,  
  belop NUMBER,  
  dato DATE,  
  
  CONSTRAINT oppdrag_pk PRIMARY KEY (oppdragsnr),  
  CONSTRAINT oppdrag_fk1 FOREIGN KEY (frakonto) REFERENCES konto (kontonr),  
  CONSTRAINT oppdrag_fk2 FOREIGN KEY (tilkonto) REFERENCES konto (kontonr),  
);  
CREATE SEQUENCE oppdrag_auto_seq;  
CREATE OR REPLACE TRIGGER oppdrag_auto_number  
BEFORE INSERT ON betalingsoppdrag  
FOR EACH ROW  
BEGIN  
  SELECT oppdrag_auto_seq.nextval INTO :new.oppdragsnr FROM dual;  
END;
```