Høgskolen i Agder
Avdeling for økonomi- og samfunnsfag

# EKSAMEN

| | |
|---|---|
| **Emnekode:** | **IS-202** |
| **Emnenavn:** | **Programmeringsrelaterte emner** |
| Dato: | **1. mars 2006** |
| Varighet: | **0900-1300** |
| Antall sider inkl. forside: | 3 |
| Målform: | **Norsk** |
| Tillatte hjelpemidler: | **Alle trykte og skrevne** |
| Merknader: | Les hele oppgaven før du begynner å svare.<br><br>Du kan besvare oppgave 2 og 3 ved skrive et sammenhengende program i stedet for å besvare hver deloppgave for seg. Bruk i tilfelle kommentarer til å markere svaret på hvert spørsmål. |

---

## Bakgrunn

Du skal lage et system som studentene kan bruke til å be om hjelp hvis det er feil eller problemer med maskinene på laben. Det er meningen at dette skal være et web-basert system slik at meldingene kan sendes direkte fra hver maskin på laben.

Meldingene om problemer blir vist fram på operatørenes skjerm. Når operatørene har løst problemet og registrert det i systemet og meldingen blir slettet.

## Oppgave 1 (teller 20%)

Det er bestemt at operatørene skal behandle meldingene i samme rekkefølge som de kommer inn. Det vil si at når det kommer en melding på operatørens skjerm må han/hun løse problemet, registrere det som løst, og så kommer neste melding, osv.

Hva slags datastruktur kan du bruke her? Kan du bruke klasser fra standardbiblioteket til Java? I tilfelle hvilke? Begrunn svaret.

## Oppgave 2 (teller 50%)

Operatørene misliker systemet fordi mange studenter rapporterer de samme problemene. De mener også at studentene kunne løst mange av problemene selv.

For å bøte på dette blir det besluttet å utvide systemet med en erfaringsdatabase. Når operatøren har løst et problem skal hun registre en beskrivelse av løsningen i systemet. For å kunne finne igjen løsningen må hun også registrere et eller flere søkeord.

Når studentene melder problemer i den nye versjonen av systemet skal systemet søke i erfaringsdatabasen etter løsninger på problemet, og vise disse for studenten. Hvis studenten ikke klarer å løse problemet selv med hjelpen fra erfaringsdatabasen kan han sende problemet videre til operatørene.

a) Skriv en klasse som kan brukes til å holde på løsningsbeskrivelsene. Data som må registreres om hver løsning er: Problembeskrivelsen (spørsmålet), løsningsbeskrivelsen, søkeord for løsningen og brukernavn på operatøren som la inn løsningen. Resten av oppgaven forutsetter at klassen heter Solution, og at hvert Solution objekt inneholder løsningen på ett problem.
b) For å gjøre responsen til systemet så rask som mulig skal hele erfaringsdatabasen holdes i en datastruktur i minnet. Hva slags datastruktur bør du bruke til dette? Hvorfor? Kan du bruke klasser fra standardbiblioteket
c) Skriv deklarasjonene du trenger i klassen ExperienceDb (se vedlegg )
d) Skriv ferdig metoden addSolution() som brukes når operatørene lagrer en ny løsning i erfaringsdatabasen
e) Skriv ferdig metoden findSolution(Vector<String> keyWords) som brukes til å finne forslag til løsning for studentene

## Oppgave 3 (teller 30%)

Skolen ønsker en oversikt over hvor mange problemer hver operatør har løst.
Skriv ferdig metoden findSolution(String operator) som finner alle løsningene som er lagt inn av en operatør.

## Vedlegg 1 class ExperienceDb

```
/**
 * ExperienceDb brukes til å holde erfaringsdatabasen
 * i minnet.
 */
public class ExperienceDb {
   // Deklarasjoner av datastruktur her (oppg 2c)


   /** Metode som brukes til å legge inn nye erfaringer. Metoden
    * lager et nytt Solution objekt og lagrer det slik at det
    * kan finnes igjen ved søk på et av ordene i keywords.
    *
    * @param keywords en Vector som inneholder søkeord for løsningen
    * @param solution en beskrivelse av løsningen
    * @param operator brukenavnet til operatøren som la inn løsningen
    */
   public void addSolution(Vector<String> keywords,
                           String solution,
                           String operator) {
      // oppgave 2d
   }


   /** Metode som brukes til å finne en løsning på et problem. Den
    * finner alle Solution objekter hvor minst et av ordene i words
    * finnes i keywords.
    *
    * @param words en Vector som inneholder ordene det skal søkes etter
    * @return en Vector som inneholder alle Solution objekter med minst
    * et av ordene i words i keywords.
    */
   public Vector<Solution> findSolution(Vector<String> words) {
      // oppgave 2e
   }


   /* Metode som finner alle Solution objekter som er registrert av
    * en bestemt operatør.
    *
    * @param operator brukernavnet til operatøren.
    * @return en Vector som inneholder alle Solution objekter
    * registrert av denne operatøren
    */
   public Vector<Solution> findSolution(String operator) {
      // oppgave 3
   }
```

## Vedlegg 2 class Solution

```
/**
 * Klasse som inneholder en løsning
 */
public class Solution {
      // oppgave 2a
}
```