



EKSAMENSOPPGAVE

EMNE: DAT101 PROGRAMMERING GRUNNKURS
FAGLÆRER: Folke Haugland

Klasse(r): alle	Dato: 21.sept 2005	Eksamenstid, fra-til: 09.00 - 14.00	
Eksamensoppgaven består av følgende	Antall sider: 5 inkl. forside	Antall oppgaver: 10	Antall vedlegg: 0
Tillatte hjelpemidler er:	Alle skrevne og trykte, alle kalkulatorer		
KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG			

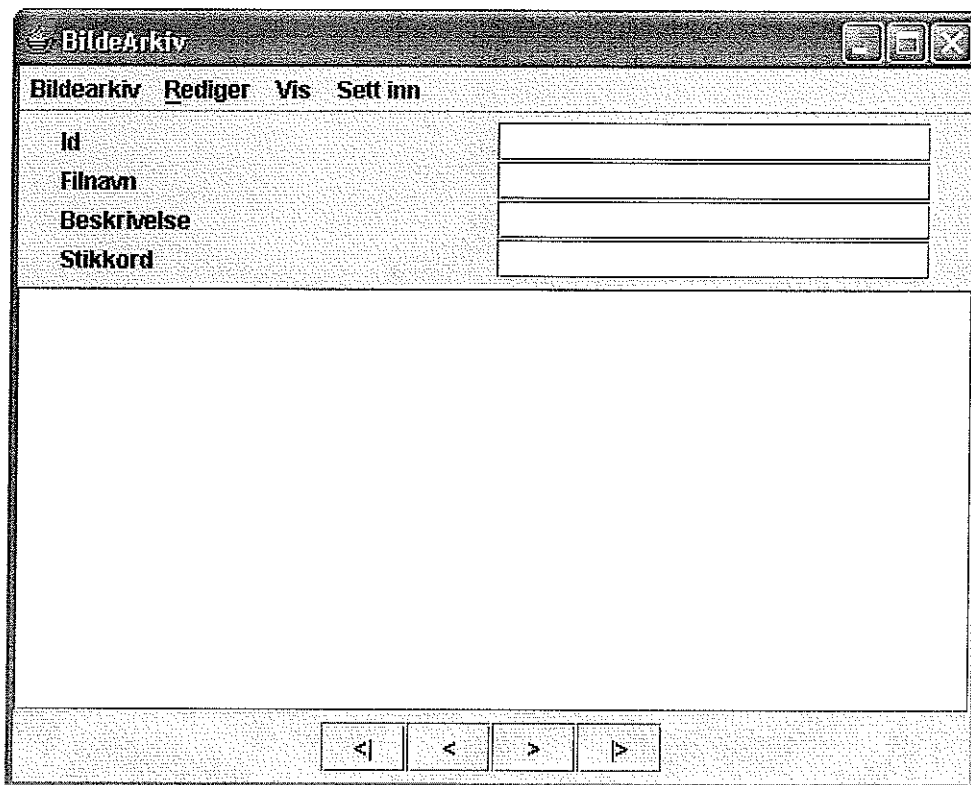
I denne eksamensoppgaven skal vi se på et program som holder rede på et enkelt arkiv over bildefiler slik som gif eller jpg. Du kan selv presisere oppgavene dersom du finne det nødvendig for å besvare oppgavene.

Du kan bruke følgende som utgangspunkt for å besvare oppgavene. Skriv de linjene med kode du mener er nødvendig og si hvilket område koden skal plasseres i, jfr teksten som vises i figur 1 nedenfor. Du kan bruke henvisning til figur 1 for å slippe å gjenta kode. **Du står fritt til å lage din egen notasjon, men det anbefales å gjøre noe som ligner på nedenstående for å redusere skrivearbeidet.**

```
// område A
public class MinRamme extends JFrame{
// område B
public minramme(){
// område C
}
// område D
public static void main(String[] args){
MinRamme m=new MinRamme();
m.setVisible(true);
// område E
}
// område F
}
```

Figur 1 Forslag til notasjon for plassering av kode

Figur 2 under viser hvordan grensesnittet skal se ut. Du behøver kun å bruke de vanlige grafiske komponentene i Java.



Figur 2 Programmet slik det ser ut under kjøring

Oppgave 1)

Rammen har BorderLayout, se figur 2 over.

Skriv koden som skal til for å få

- Et skrollbart tekstområde i senter av ramma
- fire knapper i sør som vist. Teksten i knappene er <|, <, >, >|
- ramma skal ved programstart være 500 pixler bred og 400 pixler høy
- ramma skal sentreres på skjermen
- ramma skal ha tittelen "BildeArkiv"

Du trenger ikke å ta hensyn til om ramma har plass i skjermbildet.

Oppgave 2)

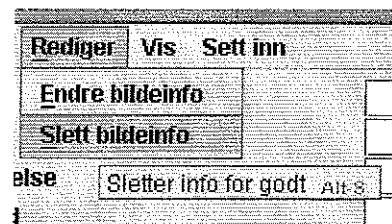
Skriv koden som skal til for å få et panel med 4 labler og 4 tekstfelt som vist i figur 2 over.

For å få et fullgodt svar må du legge et panel (med en passende layout) med komponentene (labler og tekstfelt) i et panel for å få utseende som vist på figur 2.

Tekstfeltene skal ha plass til 20 tegn.

Oppgave 3)

Skriv koden for å få fram meny systemet med hovedmenyen i figur 2. Når brukeren velger "Rediger" skal det vises en undermeny som vist i figuren til høyre, inklusiv hjelpeteksten "Sletter info for godt" når brukeren lar musa hvile på menyvalget "Slett bildeinfo".



Klassen BildeArkiv inneholder en ArrayList (eller lignende du kan velge) som inneholder instanser av klassen BildeFil. Du trenger ikke å ta hensyn til at registeret blir for stort. Klassen er ikke fullstendig.

```
import java.util.ArrayList;
public class BildeArkiv {
    ArrayList al=new ArrayList();
    public BildeArkiv(){
    }
    public void add(BildeFil bf){
        // ikke ferdig
        al.add(bf);
    }
    public BildeFil getBildeFil(String filnavn){
        //ikke ferdig
        return null;
    }
}
```

Koden til klassen BildeFil under kan utvides/endres.

```
public class BildeFil {
    int id=0;
    String filnavn="",beskrivelse="",stikkord="";
    public void setBeskrivelse(String beskrivelse) {
        this.beskrivelse = beskrivelse;
    }
    public void setFilnavn(String filnavn) {
        this.filnavn = filnavn;
    }
    public void setId(int id) {
        this.id = id;
    }
    public void setStikkord(String stikkord) {
        this.stikkord = stikkord;
    }
    public int getId() {return id;}
    public String getBeskrivelse() {return beskrivelse; }
    public String getFilnavn() {return filnavn; }
    public String getStikkord() {return stikkord; }
    public String toString(){return ""+id+":"+filnavn+":"+beskrivelse;}
}
```

Klassen ArrayList innholder bl a. metodene

boolean add(Object o) som lagrer objektet

int size() som returnere hvor mange objekter som er lagret

Object get(i) som returnere objekt nr i i lista. Nummereringa starter på 0.

Det er et krav til filnavnet at teksten alltid skal starte enten med en bokstav og : (dvs C:) eller en url (http://). Det er ikke skille på små og store bokstaver. Det er et krav til stikkord at det skal være en liste av ord uten mellomrom som er skilt med komma for eksempel "Grimstad,Luftsloppet,konsert"

Oppgave 4)

Klassen `BildeArkiv` har en metode

```
public BildeFil getBildeFil(String filnavn)
```

Metoden skal sjekke om teksten `filnavn` er korrekt i forhold til kravet til `filnavn` beskrevet nederst på forrige side. Dersom kravet ikke er oppfylt, så skal metoden returnere **null**.

Metoden returnere også **null**, dersom det ikke finnes noen eksisterende bildefil i arkivet med inn-parameteren `filnavn`. Ellers skal metoden returnere den aktuelle bildefila.

Oppgave 5)

Klassen `Arkiv` har en metode

```
public boolean add(BildeFil bf)
```

Dersom inn-parameteren `bf` er `null`, så skal det ikke legges noe til i arkivet og metoden skal returnere `usann`. Det samme gjelder dersom objektet som ønskes satt inn har `filnavn` `null` eller lengde 0 eller ikke oppfyller kravet til `filnavn` nederst på forrige side. Det samme gjelder dersom kravet til stikkord ikke er oppfylt. Det samme gjelder også dersom det allerede finnes en bildefil i arkivet med samme `filnavn`. I alle andre tilfeller skal metoden returnere `sann` og bildefila lagres i arkivet.

Skriv koden for metoden `add`.

Hvor i koden vil du opprette et objekt av klassen *Arkiv*?

Oppgave 6)

Feltet `stikkord` i klassen `BildeFil` kan brukes til å lagre en liste med stikkord som på et senere tidspunkt kan brukes til å finne tilbake bilder.

Skriv en metode i klassen `BildeArkiv`:

```
public String getBilderFraStikkord (String søk)
```

Denne metoden skal finne om det finnes bildefiler med et stikkord som inneholder teksten *søk*. Små eller store bokstaver skal ikke bety noe. Dersom ingen bildefiler oppfyller kravet, så skal metoden returnere "Ingen bildefil med stikkordet xxx ". Ellers skal metoden returnere en linje per bildefil, se metoden `toString()` i klassen `BildeFil`.

Oppgave 7)

Metoden `getBilderFraStikkord` i oppgave 7 har en svakhet. Dersom det er svært mange bildefiler med et stikkord som inneholder teksten *søk*, så returnerer metoden en streng med mange linjer.

Endr metoden slik at dersom det er mer enn 40 bildefiler som oppfyller søkekriteriet, så skal metoden returnere strengen "Søket gav treff på mer enn 40 bildefiler, snevr inn søket". Ellers skal metoden returnere samme streng som i oppgave 6.

Oppgave 8)

Klassen `BildeFil` er lite effektiv når det gjelder å skaffe informasjon om de enkelte stikkord. Endr metoden `setStikkord` slik at metoden også oppretter en `ArrayList` med et element for hvert stikkord. For eksempel skal stikkord "Grimstad,Luftslottet,konsert" bli lagret i en `ArrayList` med 3 elementer "Grimstad", "Luftslottet" og "konsert".

Oppgave 9)

Skriv en metode `public boolean existStikkord(String ord)` i klassen `BildeFil` som returnerer sann dersom ord er en del av et av enkeltstikkordene i instansvariabelen **stikkord**.

Endr koden for metoden

```
public String getBilderFraStikkord (String søk)
i klassen BildeArkiv slik at den bruker metoden existStikkord
```

Oppgave 10)

Endr metoden `public boolean existStikkord(String ord)` i klassen `BildeFil`. Dersom innparameteren `ord` består av mellomrom, så skal det tolkes som at det skal letes etter bilder som oppfyller flere krav. Dersom for eksempel `ord` er "Grimstad Luftslettet", så skal det letes etter bilder som både inneholder stikkordet `Grimstad` og stikkordet `Luftslettet`.

Du kan selv gjøre dine egne preiseringer.

Lykke til!!

God sommer!!!