



## EKSAMENSOPPGAVE

**EMNE:** DAT101 PROGRAMMERING GRUNNKURS  
**FAGLÆRER:** Folke Haugland

<b>Klasse(r):</b> alle	<b>Dato:</b> 25.mai 2007	<b>Eksamenstid, fra-til:</b> 09.00 - 13.00	
<b>Eksamensoppgaven består av følgende</b>	<b>Antall sider:</b> 5 inkl. forside	<b>Antall oppgaver:</b> 10	<b>Antall vedlegg:</b> 0
<b>Tillatte hjelpemidler er:</b>	<b>Alle skrevne og trykte, alle kalkulatorer</b>		
<b>KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG</b>			

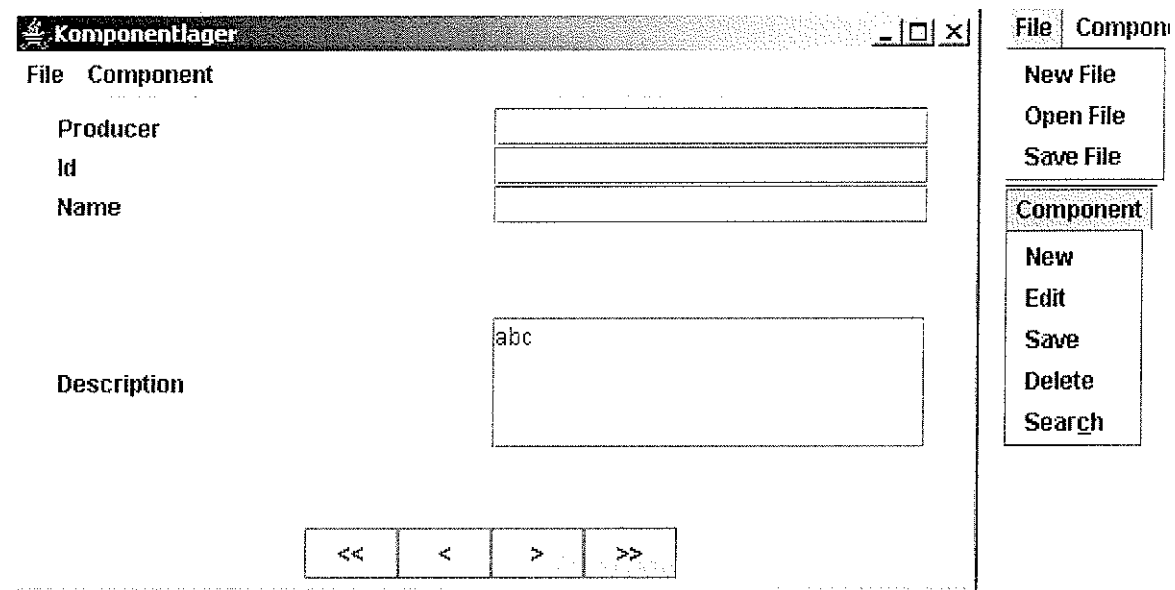
I denne eksamensoppgaven skal vi se på et program som holder rede på et enkelt arkiv over komponenter i et lager. Du kan selv presisere oppgavene dersom du finne det nødvendig for å besvare oppgavene.

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 skrivarbeidet.**

```
// 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)

Skriv koden som skal til for å få et sentrert ramme som er 500 pixler bredt og 300 pixler høyt med tittelen "Komponentlager".

Skriv koden for å få de 4 knappene i et panel i et panel i sør i ramma.

Skriv koden som skal til for at du får et panel med navn `jp_oppe` med de 3 labelene "Producer", "id" og "Name" med 3 tekstfelt med lengde 20 som vist på figuren over. Du velger selv en passende layout. Du skal ikke plassere dette panelet før i neste oppgave

### Oppgave 2)

Skriv koden som skal til for å få et panel med navn `jp_nede` med labelen "Description" og et skrollbart tekstområde som vist på figuren. Velg en passende layout

Skriv koden som skal til for at `jp_nede` og `jp_oppe` blir plassert i hvert sitt panel. Disse to nye panelene skal plasseres i et panel med passende layout i center av ramma.

### Oppgave 3)

Skriv koden for å få menysystemet som vist i figuren.

Forklar hovedtrekken i eventhandling og skriv koden som skal til for at alle tekstfelt og tekstområder blir tømt for innhold når brukeren velger Component/New

### Oppgave 4)

Skriv koden for at klasse Component som kan inneholde de 4 opplysningene om en komponent som vist i figuren samt de tilhørende metodene for å hente/endre disse opplysningene. Alle variabelene skal være av typen String.

---

Klassen Arkiv inneholder en ArrayList som inneholder instanser av klassen Component. (Du kan alternativt bruke Vector eller ArrayList<Component>). Du trenger ikke å ta hensyn til at registeret blir for stort. Klassen er ikke fullstendig.

```
import java.util.ArrayList;

public class Arkiv {
    ArrayList al=new ArrayList();
    public Arkiv(){
    }
    public void add(Component bf){
        al.add(bf);
    }
    public int GetComponent(String id){
        //ikke ferdig
        return null;
    }
}
```

Klassen ArrayList inneholder bl.a. metodene  
boolean add(Object o) som lagrer objektet  
int size() som returnerer hvor mange objekter som er lagret  
Object get(i) som returnerer objekt nr i i lista. Nummereringa starter på 0.

---

### Oppgave 5)

Klassen Arkiv har en metode

```
public int GetComponent(String id)
```

Metoden returnerer -1, dersom det ikke finnes noen eksisterende komponent i arkivet med en id som inneholder delteksten id. Ellers skal metoden returnere indeksen til den aktuelle komponenten i arraylista.

Skriv en metode

```
public Component GetComponent(int index)
```

Metoden skal returnere null hvis index er utenfor det gyldige området av lovlige indekser til arraylisten. Ellers skal metoden returnere komponenten som hører til indeksen i arraylista.

### Oppgave 6)

Skriv koden som skal til for at når brukeren velger Component/Save så skal de opplysningen som er i vinduet lagres i arkivet som en instans av klassen Component. Du skal sørge for at samme id ikke brukes to ganger. For å få fullgodt svar må du ta med en nødvendig feiltesting.

### **Oppgave 7)**

Skriv koden for en metode

```
public String saveArkiv(String filnavn)
```

som lagrer arkivet på fil. Innparameteren er navnet på filen arkivet skal lagres i. Du skal returnere null hvis lagringen gikk feilfri, ellers skal du returnere en feilmelding.

Vis hvordan denne metoden kan brukes til å få menyvalget File/New File til å virke slik at brukeren må oppgi et filnavn og programmet prøver å lagre arkivet på denne fila.

### **Oppgave 8)**

Menyvalget File/Save File skal virke slik at brukeren får valget mellom å lagre på det sist lagrede filnavnet (om det finnes) eller eventuelt måtte oppgi et nytt filnavn, før arkivet lagres. Presiser ved å beskrive denne logikken for brukeren og skriv koden for dine valg.

### **Oppgave 9)**

Skriv koden slik at menyvalget File/Open File virker. Dersom det allerede er åpnet er arkiv (som eventuelt er endret) så skal brukeren først få mulighet til lagre arkivet med samme logikk som i oppgave 8.

### **Oppgave 10)**

Lag et forslag til hvordan du kan implementere en blade funksjon. Når for eksempel brukeren har søkt og funnet en komponent, så skal > gi neste komponent (i arraylisten).>> den siste. Det skal til enhver tid bare være de lovlige menyvalg/knappevalg som er mulige. Skriv koden for dette.

### **Oppgave 11)**

Lag et forslag til hvordan Component/Edit valget kan implementeres. Beskriv logikken og skriv koden

---

Lykke til!!  
God sommer!!!