

EKSAMENSOPPGAVE

EMNE: DAT101 PROGRAMMERING GRUNNKURS
FAGLÆRER: Folke Haugland

Klasse(r): 1DT,DT,1SIV	Dato: 19.mai 2004	Eksamenstid, fra-til: 09.00 - 13.00	
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			

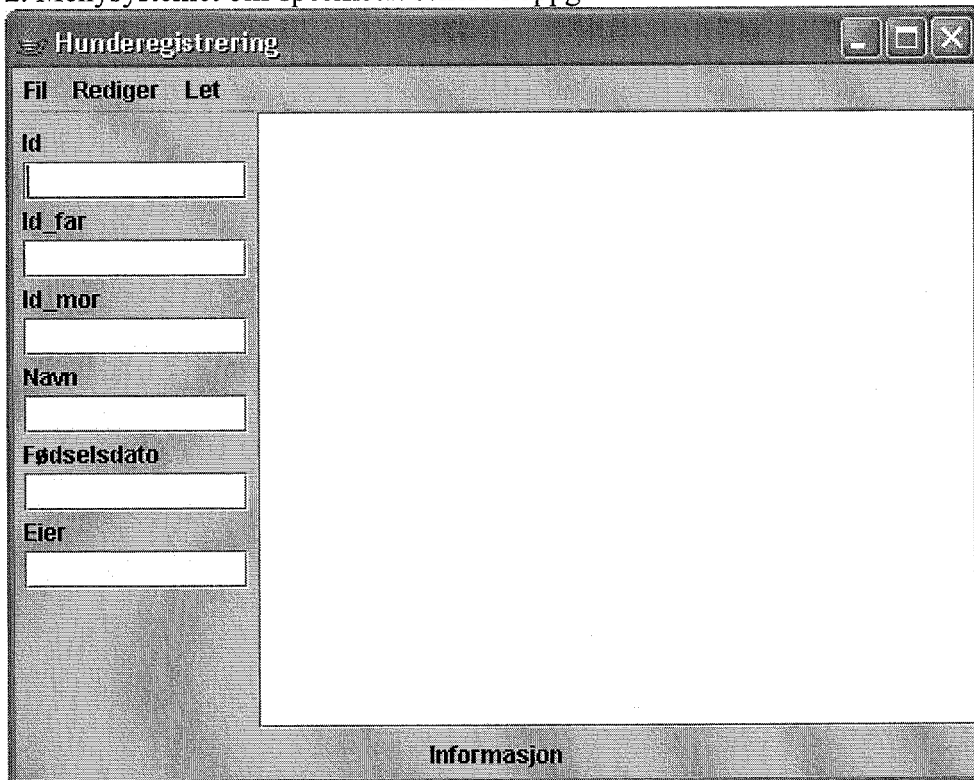
I denne eksamensoppgaven skal vi se på et program som holder rede på et enkelt arkiv over hunder. Du kan selv presisere oppgavene dersom du finne det nødvendig for å besvare oppgavene.

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. Du trenger ikke å lage layout nøyaktig som på figur 2. Menysystemet blir spesifisert senere i oppgaven.



Figur2 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 rammen
- en label i sør med teksten "Informasjon"
- ramma skal ved programstart være 500 pixler bred og 400 pixler høy
- ramma skal plasseres 100 pixler fra både toppen og venstre kant på skjermen
- ramma skal ha tittelen "Hunderegistrering"

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 6 labler og 6 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.

Oppgave 3)

På neste side står litt av koden til klassen Hund. Klassen Hund skal ha en metode

```
public String toString()
```

som returnerer en tekst uten linjesift der det står <id>:<navn>:<eier> for eksempel 400:Pluto:Mikke (Det står : uten mellomrom foran og bak.)

Skriv koden for metoden toString.

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

```
class Arkiv{
    private ArrayList al=new ArrayList ();
    private String filnavn="arkiv.txt";
    public Arkiv(){
    public boolean add(Hund h){/*ikke ferdig*/}
    public String skrivTilFil(){/*ikke ferdig*/}
    public String hentFraFil(){/*ikke ferdig*/}

    public void setFilNavn(String navn ){filnavn=navn;}
    public String getFilNavn(){return filnavn;}
    public int getSize(){return al.size();}
}
/**
 * Klasse som holder rede på alle opplysningene om en hund.
 *
 * Dersom id til foreldrene ikke er kjent,
 * så skal id_far /id_mor være 0.
 * @author FH
 */
public class Hund{
    private int id=0;
    private int id_mor=0;
    private int id_far=0;
    private String navn="";
    private String eier="";
    private String fdato="2000-01-01";
    public Hund(){
    }
    public void setId(int id){this.id=id;}
    public void setId_mor(int id_mor){this.id_mor=id_mor;}
    public void setId_far(int id_far){this.id_far=id_far;}
    public void setNavn(String navn){this.navn=navn;}
    public void setEier(String eier){this.eier=eier;}
    public void setFdato(String fdato){this.fdato=fdato;}
    public int getId(){return id;}
    public int getId_mor(){return id_mor;}
    public int getId_far(){return id_far;}
    public String getNavn(){return navn;}
    public String getEier(){return eier;}
    public String getFdato(){return fdato;}
}
```

Oppgave 4)

Klassen Arkiv har en metode

public boolean add(Hund hund)

Dersom inn-parameteren hund er null, så skal det ikke legges hund til i arkivet og metoden skal returnere usann. I alle andre tilfeller skal metoden returnere sann.

Skriv koden for metoden add.

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

Oppgave 5)

Det finnes flere måter å håndtere hendelser. Forklar kort to ulike måter å håndtere hendelser på.

Skriv koden for å få fram menysystemet for Fil vist i figuren til høyre.

Skriv koden som er nødvendig for å få lagret arkivet til fil ved å velge Fil/Lagr arkiv fra menyen.



Du kan forutsette at metoden *skrivTilFil* i klassen *Arkiv* på forrige side er skrevet og virker. Men brukeren skal ha mulighet til selv å velge navn og sti til filen (katalog og filnavn)

Oppgave 6)

Skriv koden til metodene *skrivTilFil* og *hentFraFil* i klassen *Arkiv*. Du må selv velge format på tekstfilen.

Begge metodene skal returnere enten feilmelding eller en passende tekst om at metoden ikke hadde feil.

Oppgave 7)

Det kan være aktuelt å ha mulighet til å spørre arkivet om det finnes en hund med en bestemt id.

Klassen *Arkiv* skal ha metoder for å finne om en hund med gitt id finnes. Skriv metodene:

```
/**
 * Finn om det finnes en hund med en gitt id
 * @param id id det skal letes etter
 * @return hund hvis det finnes hund med den gitte id
 *         null hvis det ikke finnes hund med den gitte id
 */
private Hund getHund(int id)
```

```

/**
 * Finn om det finnes en hund med en gitt id
 * @param id id det skal letes etter
 * @return true hvis det finnes hund med den gitte id
 *         false hvis det ikke finnes hund med den gitte id
 */
public boolean existsHund(int id)

```

Metoden existsHund kan bruke metoden getHund()

Oppgave 8)

Skriv en metode i klassen Arkiv:

```
public String getChildren(int id)
```

Metoden skal returnerer "Hund med id xxx finnes ikke" hvis det ikke er registrert hund med inn-parameteren id. Hvis det ikke er registrert noen barn, så skal metoden returnere "Ikke registrert barn". Ellers skal metoden returnere en linje per barn og på hver linje skal det stå <id>:<navn>:<eier>for eksempel

400: Pluto:Mikke

på samme måte som i oppgave 4.

Oppgave 9)

Det er et krav om at brukeren av programmet fritt kan velge id til en Hund, men det skal ikke legges til en instans av klassen Hund til arkivet dersom id-en finnes fra før. Forklar hvordan du kan sikre at det ikke finnes to hunder med samme id i arkivet.

Oppgave 10)

Det er en arvelig hundesykdom som bare spres med mor. Dersom en hund er syk, er det derfor viktig å finne alle mødrene i stamtreet - det vil si mor, mormor etc. for å undersøke spredningen av sykdommen.

Skriv en metode

```
public String getMortre(int id)
```

som returnerer

en linje hver for mor, mormor etc til hund med inn-parameteren id. Dataene skal gå så langt tilbake det finnes data. Hver linje skal være som i oppgave 4

<id>:<navn>:<eier>

eventuelt gis det en feilmelding som svar dersom inn-parameteren ikke er id-en til en hund i arkivet.

Du kan forutsette at alle id-er som er større enn 0, er id-en til en Hund i arkivet. En id som er 0, betyr at hunden ikke er kjent i arkivet.

Lykke til!!

God sommer!!!