



EKSAMENSOPPGAVE

EMNE: DAT101 PROGRAMMERING GRUNNKURS
FAGLÆRER: Folke Haugland

Klasse(r): alle	Dato: 31.mai 2006	Eksamenstid, fra-til: 09.00 - 13.00	
Eksamensoppgaven består av følgende	Antall sider: 5 inkl. forside	Antall oppgaver: 9	Antall vedlegg: 0
Tillatte hjelpemidler er:	Alle skrevne og trykte, alle kalkulatorer		
KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG			

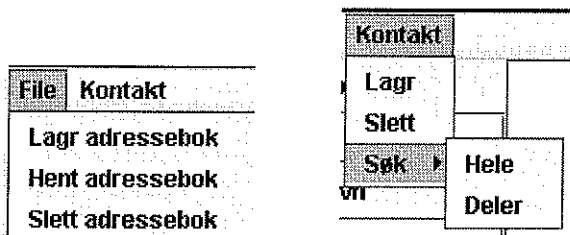
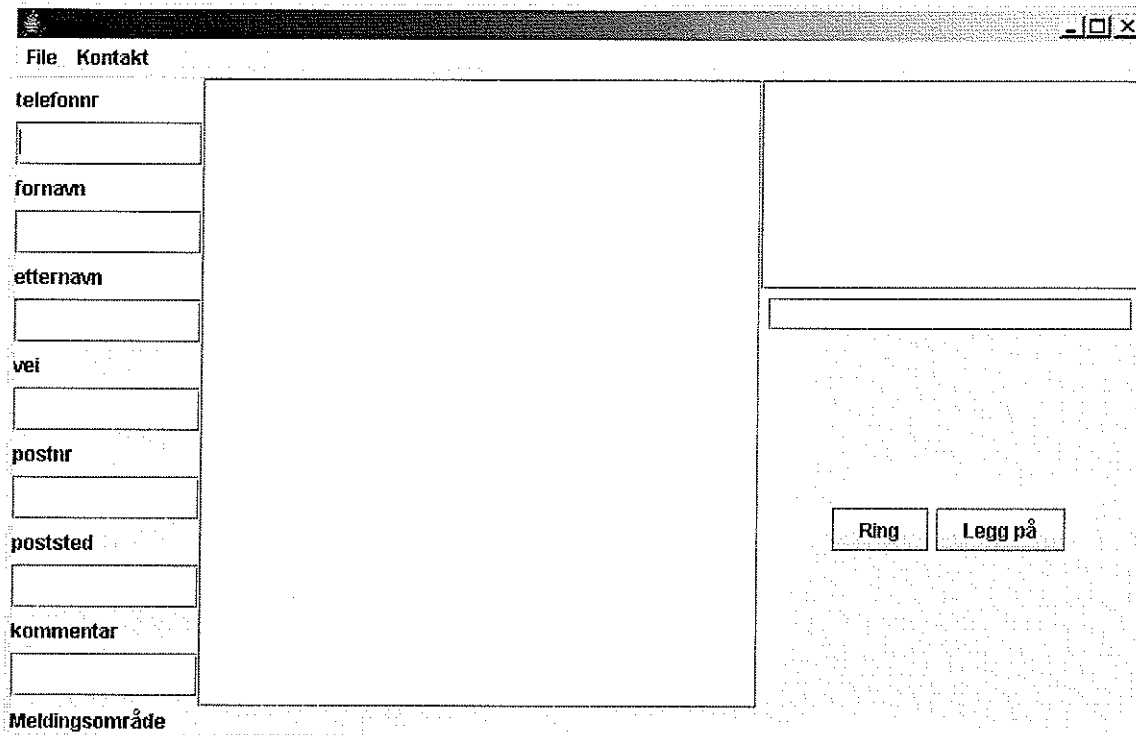
I denne eksamensoppgaven skal vi se på et program som holder rede kontaktene i en telefon. Du kan selv presisere oppgavene dersom du finne det nødvendig for å besvare oppgavene.

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
- En label i sør som vist.
- ramma skal ved programstart være 600 pixler bred og 450 pixler høy
- ramma skal plasseres 100 pixler fra både toppen og venstre kant på skjermen
- ramma skal ha tittelen "IP-Telefon"

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 i øst (høyre side) som vist i figur 2 over. Øverst er det et skrollbart tekstområde med i utgangspunktet 7 linjer og plass til 20 tegn på hver linje. De to knappene er plassert i et eget panel. Tekstfeltet har i utgangspunktet plass til 20 tegn og plasseres i et eget panel.

Oppgave 3)

Under står litt av koden til klassen Kontakt. Klassen Kontakt skal ha en metode

```
public String toString()
```

som returnerer en tekst med linjesift der det står

```
<fornavn>:<etternavn >
<telefonnr>
<kommentar>
```

Hvis det mangler informasjon på en linje, så skal det ikke skrives en tom linje. Hvis det for eksempel ikke finnes noen kommentar om Folke Haugland, så vil metoden kunne returnere strengen under, med to linjer tekst og linjeskift etter telefonnr.

```
Folke Haugland
37253000
```

Skriv koden for metoden toString

Klassen Adressebok inneholder en ArrayList (eller lignende du kan velge) som inneholder instanser av klassen Kontakt. Du trenger ikke å ta hensyn til at lista blir for lang. Klassene er ikke fullstendige.

```
import java.util.ArrayList;
public class Adressebok {
    ArrayList al=new ArrayList();
    public Adressebok (){
        //slutt konstruktør
    public void add(Kontakt k){//ikke ferdig
        al.add(k);
    }//slutt add
    public Kontakt getKontakt (String telefonnr){
        //ikke ferdig
        return null;
    }//slutt getKontakt
} //slutt klasse
```

Klassen ArrayList inneholder 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.

```
public class Kontakt{
    String telefonnr,fornavn,etternavn,vei,poststed,kommentar,fdato;
    int postnr;
    public Kontakt(String telefonnr,String fornavn,String etternavn,
        String vei,int postnr,String poststed,String fdato,String kommentar){
        this.telefonnr= telefonnr;
        this.fornavn = fornavn;
        this.etternavn = etternavn;
        this.vei = vei;
        this.postnr = postnr;
        this.poststed = poststed;
        this.fdato=f.dato;
        this.kommentar=kommentar;
    } //slutt konstruktør
    public String getTelefonnr() {return telefonnr ;}
    public String getFornavn () {return fornavn;}
    public String getEtternavn () {return etternavn;}
}
```

```

public String getVei () {return vei;}
public int getPostnr() {return postnr;}
public String getPoststed () {return poststed;}
public String getFdato () {return fdato;}
public String getKommentar(){return kommentar;}
public String toString(){//ikke ferdig
    String svar="";
    return svar;
} //slutt toString
} //slutt klasse

```

Oppgave 4)

Klassen Adressebok har en metode

```
public Kontakt getKontakt(String telefonnr)
```

Metoden returnere null, dersom det ikke finnes noen eksisterende kontakt i adresseboka med et telefonnr lik inn-parameteren telefonnr. Ellers skal metoden returnere den aktuelle kontakten.

Det skal tas høyde for at med IP-telefoni, så kan telefonnummer være en tekst med navnet på personen. Det må derfor ikke skilles mellom små og store bokstaver.

Det skal tas høyde for eventuelle feilsituasjoner som kan oppstå.

Oppgave 5)

Klassen Adressebok har en metode

```
public String add(Kontakt k)
```

Dersom inn-parameteren k er null, så skal det ikke legges noe til i arkivet. Det samme gjelder dersom objektet som ønskes satt inn, ikke har telefonnr dvs telefonnr er null eller har lengde 0. I alle andre tilfeller skal kontakten lagres i adresseboka. Metoden skal returnere en passende tekst avhengig av hva som skjer i metoden.

Skriv koden for metoden add.

Hvor i koden vil du opprette en instans av klassen Adressebok?

Oppgave 6)

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

Skriv koden for å få fram menysystemet som vist i figur 2 på side 2.

Skriv koden som er nødvendig for å få lagret en ny instans av klassen Kontakt i adresseboka ved å velge Kontakt/Lag fra menyen. Eventuelle meldinger skal skrives i labelen i sør i ramma.

Du kan forutsette at metoden add i klassen Adressebok (se oppgave 5), er skrevet og virker.

Oppgave 7)

Hver kontakt inneholder informasjon om telefonnr.

Skriv en metode i klassen Adressebok:

```
public String getKontakter (String søk)
```

Denne metoden skal finne om det finnes kontakter med et telefonnr som inneholder teksten *søk*. Dersom ingen kontakter oppfyller kravet, så skal metoden returnerer "Ingen kontakter med telefonnr som inneholder xxx " der xxx er teksten *søk*. Ellers skal metoden returnere en linje per kontakt og på hver linje skal det stå <telefonnr>:<fornavn>:<etternavn>.

Dersom det finnes mer enn 40 kontakter som oppfyller søket, så skal metode returnere en liste over de 40 første kontaktene og deretter en egen linje med meldingen: "Det er flere kontakter, innsnevr søket".

Oppgave 8)

Du har fått et java-bibliotek som kan brukes til å ringe fra pc-en. Disse forutsetter at pc-en har lydkort, mikrofon og høyttalere og eventuelt annet utstyr som er nødvendig for å brukes datamaskinen som telefon.

For hver telefonsamtale må det opprettes en instans av klassen under, se klassesdiagrammet.

com.yyy.Samtale
public Samtale(String telefonnr)
public String KoblOpp()
public String TaNed()
public String getTid()

Konstruktøren krever at telefonnr det skal ringes til er innparameter.

Metoden KoblOpp har ingen inn-parameter, men foretar en faktisk oppkobling mot det ønskede telefonnr. Når samtalen avsluttes eller mislykkes, så returnerer metoden en aktuell melding om samtalen slik som feilmelding, mottaker ikke tilgjengelig, mottaker avviste eller lengden og kostnaden til samtalen når samtalen er ferdig.

Metoden TaNed() avslutter samtalen og fører til at metoden KoblOpp avsluttes.

Metoden getTid() returnerer samtalens lengde i millisekunder.

Skriv koden for at knappene "Ring" og "Legg På" virker med dette biblioteker. Telefonnr skal hentes fra tekstfeltet i høyre del av ramma. Aktuelle meldinger skal skrives i labelen i sør i ramma.

Oppgave 9)

Denne oppgaven tar for seg situasjonen at brukeren velger Kontakt/Søk/Deler. Skisser en løsning der brukeren får opp en dialogboks og kan gi deler av telefonnr. Det skal da komme en liste over kontakter i tekstområdet i senter. Bare dersom det finnes kun en kontakt, skal det legges aktuell informasjon i tekstområdet slik som beskrevet i metoden toString i klassen Kontakt og telefonnr i tekstfeltet til høyre.

Dersom brukeren selv skriver et telefonnr i tekstfeltet i høyre del av ramma, og velger "Ring", så skal det komme informasjon om kontakten (hvis det finnes) i tekstområdet i høyre del av ramma.

Lykke til!!

God sommer !!!



EKSAMENSOPPGAVE

EMNE: DAT1

GRUNNKURS

FAGLÆRER: F

Stor sluff

Klasse(r): alle	Dat 200	tid: 09.00 - 13
Eksamensoppgave består av følgende	An sid forside	Antall vedlegg: 0
Tillatte hjelpemidler er:	Alle skrevne og trykte, alle kalkulatorer	
KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETIET ER FULLSTENDIG		

I denne eksamensoppgaven skal vi se på et program som holder rede kontaktene i en telefon. Du kan selv presisere oppgavene dersom du finne det nødvendig for å besvare oppgavene.

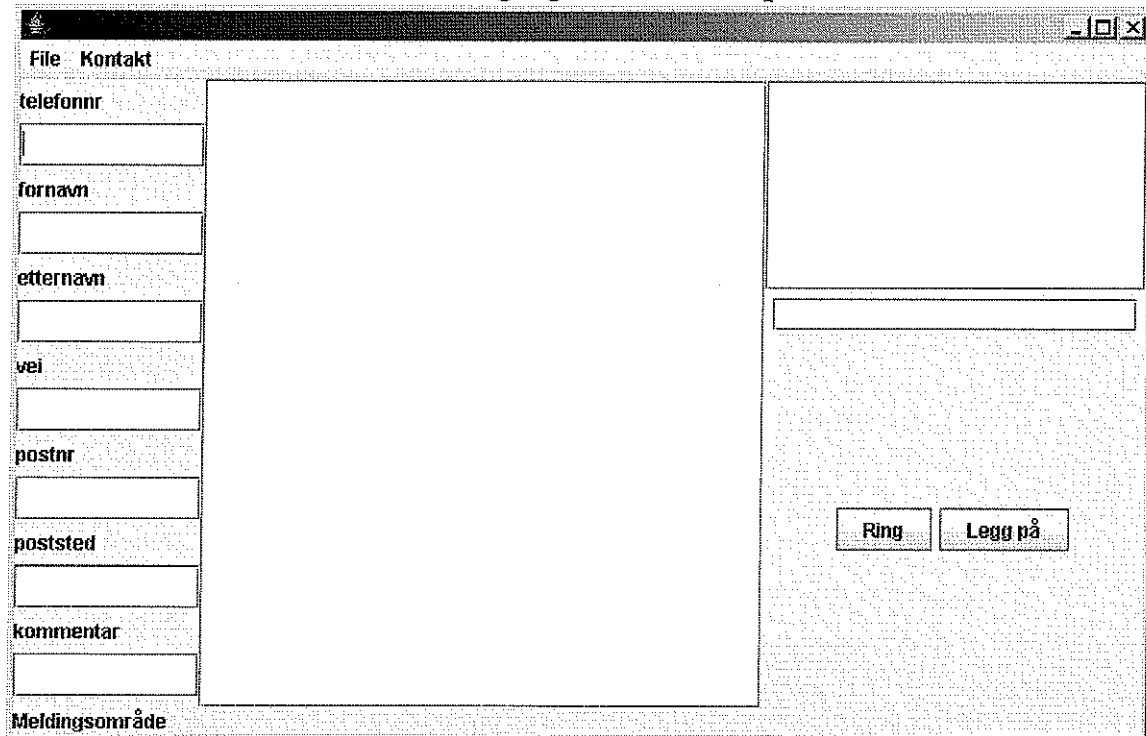
Du kan bruke følgende som utgangspunkt for å besvare oppgavene. Skriv de linjene med kode du mener er nødvendig og si hvilket område koden skal plasseres i, jfr teksten som vises i figur 1 nedenfor. Du kan bruke henvisning til figur 1 for å slippe å gjenta kode.

Du står fritt til å lage din egen notasjon, men det anbefales å gjøre noe som ligner på nedenstående for å redusere skrivearbeidet.

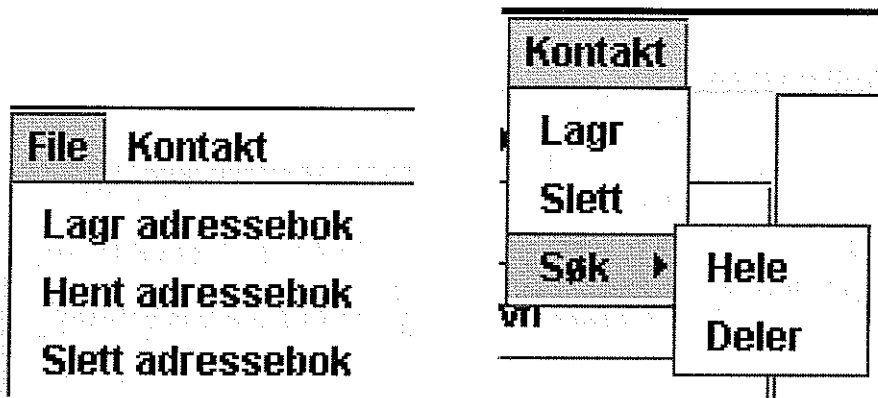
```
// område A
public class MinRamme extends JFrame{
    // område B
    public MinRamme(){
        // område C
    }
    // område D
    public static void main(String[] args){
        MinRamme m=new MinRamme();
        m.setVisible(true);
    // område E
    }
    // område F
}
```

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.



The image shows a Java Swing window titled "Kontakt" (Contact). The window has a standard Mac OS-style title bar with a menu bar containing "File" and "Kontakt". The main content area is divided into two columns. The left column contains a vertical stack of input fields, each with a label to its left: "telefonnr", "fornavn", "etternavn", "vei", "postnr", "poststed", "kommentar", and "Meldingsområde". The right column contains a large empty rectangular area at the top, a horizontal input field below it, and two buttons at the bottom labeled "Ring" and "Legg på".



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 ramma
- En label i sør som vist.
- ramma skal ved programstart være 600 pixler bred og 450 pixler høy
- ramma skal plasseres 100 pixler fra både toppen og venstre kant på skjermen
- ramma skal ha tittelen "IP-Telefon"

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 i øst (høyre side) som vist i figur 2 over. Øverst er det et skrollbart tekstområde med i utgangspunktet 7 linjer og plass til 20 tegn på hver linje. De to knappene er plassert i et eget panel. Tekstfeltet har i utgangspunktet plass til 20 tegn og plasseres i et eget panel.

Oppgave 3)

Under står litt av koden til klassen Kontakt. Klassen Kontakt skal ha en metode

```
public String toString()
```

som returnerer en tekst med linjesift der det står

```
<fornavn>:<etternavn >
<telefonnr>
```


<kommentar>

Hvis det mangler informasjon på en linje, så skal det ikke skrives en tom linje. Hvis det for eksempel ikke finnes noen kommentar om Folke Haugland, så vil metoden kunne returnere strengen under, med to linjer tekst og linjeskift etter telefonnr.

Folke Haugland
37253000

Skriv koden for metoden toString.

Klassen Adressebok inneholder en ArrayList (eller lignende du kan velge) som inneholder instanser av klassen Kontakt. Du trenger ikke å ta hensyn til at lista blir for lang. Klassene er ikke fullstendige.

```
import java.util.ArrayList;
public class Adressebok {
    ArrayList al=new ArrayList();
    public Adressebok (){
    }//slutt konstruktør
    public void add(Kontakt k){//ikke
ferdig
        al.add(k);
    }//slutt add
    public Kontakt getKontakt (String
telefonnr){
        //ikke ferdig
        return null;
    }//slutt getKontakt
}//slutt klasse
```

Klassen ArrayList inneholder 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.

```
public class Kontakt{
    String
    telefonnr,fornavn,etternavn,vei,poststed,kommentar,fda
    to;
    int postnr;
    public Kontakt(String telefonnr,String
fornavn,String etternavn,
        String vei,int postnr,String poststed,String
fdato,String kommentar){
        this.telefonnr= telefonnr;
        this.fornavn = fornavn;
        this.etternavn = etternavn;
        this.vei = vei;
        this.postnr = postnr;
        this.poststed = poststed;
        this.fdato=f.dato;
        this.kommentar=kommentar;
    }//slutt konstruktør
    public String getTelefonnr() {return telefonnr ;}
    public String getFornavn () {return fornavn;}
    public String getEtternavn () {return etternavn;}
    public String getVei () {return vei;}
    public int getPostnr() {return postnr;}
    public String getPoststed () {return poststed;}
    public String getFdato () {return fdato;}
    public String getKommentar(){return kommentar;}
    public String toString(){//ikke ferdig
        String svar="";
        return svar;
    }//slutt toString
} //slutt klasse
```

Oppgave 4)

Klassen Adressebok har en metode

```
public Kontakt getKontakt(String telefonnr)
```

Metoden returnere null, dersom det ikke finnes noen eksisterende kontakt i adresseboka med et telefonnr lik inn-parameteren

telefonnr. Ellers skal metoden returnere den aktuelle kontakten.

Det skal tas høyde for at med IP-telefoni, så kan telefonnummer være en tekst med navnet på personen. Det må derfor ikke skilles mellom små og store bokstaver.

Det skal tas høyde for eventuelle feilsituasjoner som kan oppstå.

Oppgave 5)

Klassen Adressebok har en metode

```
public String add(Kontakt k)
```

Dersom inn-parameteren k er null, så skal det ikke legges noe til i arkivet. Det samme gjelder dersom objektet som ønskes satt inn,

ikke har telefonnr dvs telefonnr er null eller har lengde 0. I alle andre tilfeller skal kontakten lagres i adresseboka. Metoden skal returnere en passende tekst avhengig av hva som skjer i metoden.

Skriv koden for metoden add.

Hvor i koden vil du opprette en instans av klassen Adressebok?

Oppgave 6)

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

Skriv koden for å få fram menysystemet som vist i figur 2 på side 2.

Skriv koden som er nødvendig for å få lagret en ny instans av klassen Kontakt i adresseboka ved å velge Kontakt/Lag fra menyen. Eventuelle meldinger skal skrives i labelen i sør i ramma.

Du kan forutsette at metoden add i klassen Adressebok (se oppgave 5), er skrevet og virker.

Oppgave 7)

Hver kontakt inneholder informasjon om telefonnr.

Skriv en metode i klassen Adressebok:

```
public String getKontakter (String søk)
```

Denne metoden skal finne om det finnes kontakter med et telefonnr som inneholder teksten *søk*. Dersom ingen kontakter oppfyller kravet, så skal metoden returnerer "Ingen kontakter med telefonnr som inneholder xxx " der xxx er teksten *søk*. Ellers skal metoden returnere en linje per kontakt og på hver linje skal det stå <telefonnr>:<fornavn>:<etternavn>.

Dersom det finnes mer enn 40 kontakter som oppfyller søket, så skal metoden returnere en liste over de 40 første kontaktene og deretter en egen linje med meldingen: "Det er flere kontakter, innsnevr søket".

Oppgave 8)

Du har fått et java-bibliotek som kan brukes til å ringe fra pc-en. Disse forutsetter at pc-en har lydkort, mikrofon og høyttalere og eventuelt annet utstyr som er nødvendig for å brukes datamaskinen som telefon.

For hver telefonsamtale må det opprettes en instans av klassen under, se klassediagrammet.

com.yyy.Samtale
public Samtale(String telefonnr)
public String KoblOpp()
public String TaNed()
public String getTid()

Konstruktøren krever at telefonnr det skal ringes til er innparameter. Metoden KoblOpp har ingen inn-parameter, men foretar en faktisk oppkobling mot det ønskede telefonnr. Når samtalen avsluttes eller mislykkes, så returnerer metoden en aktuell melding om samtalen slik som feilmelding, mottaker ikke tilgjengelig, mottaker avviste eller lengden og kostnaden til samtalen når samtalen er ferdig.

Metoden TaNed() avslutter samtalen og fører til at metoden KoblOpp avsluttes.

Metoden getTid() returnerer samtalens lengde i millisekunder.

Skriv koden for at knappene "Ring" og "Legg På" virker med dette biblioteker. Telefonnr skal hentes fra tekstfeltet i høyre del av ramma. Aktuelle meldinger skal skrives i labelen i sør i ramma.

Oppgave 9)

Denne oppgaven tar for seg situasjonen at brukeren velger Kontakt/Søk/Deler. Skisser en løsning der brukeren får opp en dialogboks og kan gi deler av telefonnr. Det skal da komme en liste over kontakter i tekstområdet i senter. Bare dersom det finnes kun en kontakt, skal det legges aktuell informasjon i tekstområdet slik som beskrevet i metoden toString i klassen Kontakt og telefonnr i tekstfeltet til høyre.

Dersom brukeren selv skriver et telefonnr i tekstfeltet i høyre del av ramma, og velger "Ring", så skal det komme informasjon om kontakten (hvis det finnes) i tekstområdet i høyre del av ramma.

Lykke til!!

God sommer !!!