

Oppgave 1.

a)

```
public Logg()
{
    rapporter = new String[ARRAYLENGDE];
    antall = 0; // unødvendig
}

public String toString()
{
    String s = "";

    for( int i = antall-1; i >= 0; i-- )
        if(rapporter[i] != null ) // ikke nødvendig
            s += rapporter[i] + "\n";

    return s;
}
```

b)

```
public void utvidArray()
{
    String[] ny = new String[rapporter.length + UTVIDELSE];

    for( int i = 0; i < rapporter.length; i++ )
        ny[i] = rapporter[i];

    rapporter = ny;
}
```

c)

```
public void nyRapport( String s )
{
    Calendar dato = Calendar.getInstance();
    DateFormat df = DateFormat.getDateInstance();

    String rapport = "\nDato: " + df.format(dato.getTime()) + s;

    if( antall >= rapporter.length )
        utvidArray();

    rapporter[antall++] = rapport;
}
```

Oppgave 2

a)

```
public String getNavn()
{
    return navn;
}

public long getPnr()
{
    return pNr;
}

public Rom getRom()
{
    return rom;
}

public void setGjest( String g )
{
    gjest = g;
}
```

```
public void slettGjest()
{
    gjest = null;
}
```

```
b)
public void nyRapport( String rapport )
{
    loggbok.nyRapport(rapport + (gjest != null ? "\nGjest: " + gjest :
    ""));
}
```

```
public String toString()
{
    return navn + ", " + pNr;
}
```

```
public String loggbok()
{
    return toString() + ", rom " + rom.getRomNr() + "\n" +
    loggbok.toString();
}
```

Oppgave 3

```
a)
public String status()
{
    switch( status )
    {
        case INNE: return "INNE";
        case UTE:  return "UTE";
        case BESØK: return "BESØK";
        case LEDIG: return "LEDIG";
        default:   return "UKJENT";
    }
}

public boolean ledig()
{
    return beboer == null; // status == LEDIG
}
```

```
b)
public void oppdaterRomStatus( String melding, int st )
{
    status = st;
    if( beboer != null )
        beboer.nyRapport( "\nStatus: " + status() + "\n" + melding );
    display.setText( ledig() ? "LEDIG" : beboer.toString() + "\nStatus: "
    + status() );
}
```

```
c)
public boolean flyttInn( Beboer ny)
{
    if( ledig() ) // egentlig ikke nødvendig
    {
        beboer = ny;
        oppdaterRomStatus(beboer.getNavn() + " flytter inn på rom " + romNr,
        INNE);
        return true;
    }
    else
        return false;
}
```

```

public void flyttUt()
{
    arkiv += "\n" + beboer.loggbok();
    beboer = null;
    status = LEDIG;
    display.setText(status()); // display.setText("LEDIG");
}

```

```

d)
public String fårBesøk( String melding, String g )
{
    if( status == UTE )
        return beboer.getNavn() + " er ikke inne og kan ikke ta imot besøk";
    else if( status == BESØK )
        return beboer.getNavn() + " har allerede besøk.";

    beboer.setGjest( g );
    oppdaterRomStatus(melding, BESØK);
    return "Besøk registrert.";
}

public String besøkSlutt( String melding )
{
    if( status == UTE )
        return beboer.getNavn() + " er ute.";
    else if( status == INNE )
        return beboer.getNavn() + " har ikke besøk.";

    beboer.slettGjest();
    oppdaterRomStatus( melding, INNE );
    return "\nBesøk avsluttet: ";
}

```

Oppgave 4

```

a)
public String beboerLogg( long pnr )
{
    Beboer b = finnBeboer( pnr );
    if( b!= null)
        return b.loggbok();
    else
        return "Ukjent personnummer";
}

public Rom finnRom( long pnr )
{
    Beboer b = finnBeboer( pnr );
    if( b!= null) // if( b!= null && b.getRom()!=null)
        return b.getRom();
    else
        return null;
}

```

//Alternativ:

```

public Rom finnRom( long pnr )
{
    for( int i = 0; i < rommene.length; i++)
    {
        if( rommene[i] != null) // ikke nødvendig her, se Vaktvindu
        {
            Beboer b = rommene[i].getBeboer();
            if( b != null && b.getPNr() == pnr)
                return rommene[i];
        }
    }
    return null;
}

```

b)

```

public String nyBeboer(String navn, long nr )
{
    if( finnBeboer( nr ) != null )
        return navn + " er allerede innlosjert på rom " + finnRom(
            nr).getRomNr();

    for( int i = 0; i < rommene.length; i++)
    {
        if( rommene[i] != null && rommene[i].ledig() )
        {
            Beboer ny = new Beboer( navn, nr, rommene[i]);
            rommene[i].flyttInn( ny );
            return navn + " er innlosjert på rom nr. " + rommene[i].getRomNr();
        }
    }
    return "Ingen ledige rom";
}

```

// Alternativ:

```

public String nyBeboer(String navn, long nr )
{
    if( finnBeboer( nr ) != null )
        return navn + " er allerede innlosjert på rom " + finnRom(
            nr).getRomNr();

    for( int i = 0; i < rommene.length; i++)
    {
        Beboer ny = new Beboer( navn, nr, rommene[i]);

        if(rommene[i] != null && rommene[i].flyttInn( ny ) )
            return navn + " er innlosjert på rom nr. " + rommene[i].getRomNr();
    }
    return "Ingen ledige rom";
}

```

Oppgave 5

a)

```

public void innskriving()
{
    String navn = navneFelt.getText();
    long nr = Long.parseLong( nrFelt.getText() );
    String tekst = institusjonen.nyBeboer( navn, nr );
    utskrift.setText( tekst );
}

```

b)

```

private class Lytter implements ActionListener
{
    public void actionPerformed((ActionEvent e) )
    {
        if( e.getSource() == flyttInn )
            innskriving();
        else if( e.getSource() == flyttUt )
            utskrivning();
        else if( e.getSource() == ut )
            ut();
        else if( e.getSource() == inn )
            inn();
        else if( e.getSource() == besøkInn )
            besøk();
        else if( e.getSource() == besøkUt )
            besøkAvsluttet();
        else if( e.getSource() == logg )
            beboerLogg();
        else
    }
}

```

```
        visArkiv();
    }
}

c)

public class Lytter implements ActionListener
{
    private VaktVindu vindu;

    public Lytter( VaktVindu v) // Konstruktøren mottar en referanse til vinduet
    {
        vindu = v;
    }

    public void actionPerformed((ActionEvent e)
    {
        if( e.getSource() == vindu.getFlyttInn() )
            vindu.innskriving();
        else if( e.getSource() == vindu.getFlyttUt() )
            vindu.utskriving();
        else if( e.getSource() == vindu.getUt() )
            vindu.ut();
        else if( e.getSource() == vindu.getInn() )
            vindu.inn();
        else if( e.getSource() == vindu.getBesøkInn() )
            vindu.besøk();
        else if( e.getSource() == vindu.getBesøkUt() )
            vindu.besøkAvsluttet();
        else if( e.getSource() == vindu.getLogg() )
            vindu.beboerLogg();
        else
            vindu.visArkiv();
    }
}

// Nødvendige forandringer i Vaktvindu :

// -get-metoder for knappene

// i konstruktør til Vaktvindu:

Lytter lytter = new Lytter(this); // referanse til vinduet sendes med som parameter
```