

# CDIO del 4



02324 Advanced programming

Deltagere:

Emil Eriksen	s120561
Khan Mohammad Noori	s122997
Mathias Larsen	s113734

Jens Nielsen	s123115
Kim Rylund	s123667

Lærer:

Mads Nyborg & Stig Høg

### Timeregnskab:

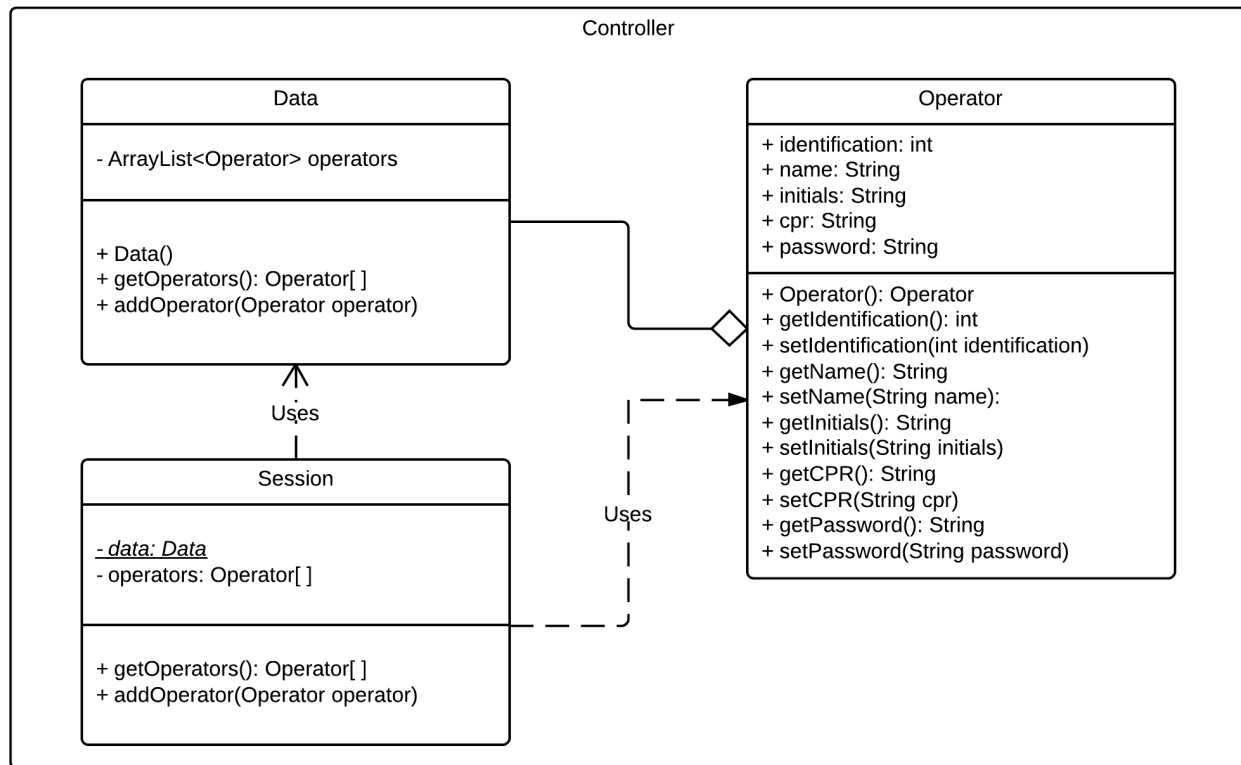
Time-regnskab	Ver. 2008-09-03												
Dato	Deltager	Design	Impl.	Test	Dok.	Andet	Ialt			Sum af Ialt	Total		
29-04-2013	Kim		2		1	2	5			Deltager			
-	Mathias	1	1	1	1	1	5			Emil	15		
-	Emil	2	2	1			5			Jens	15		
-	Jens	1	4				5			Khan	15		
-	Khan	2	2			1	5			Kim	15		
										Mathias	15		
06-05-2013	Kim	1	1	0	1	2	5			(Tom)			
-	Mathias	1	0		4		5			Hovedtotal	75		
-	Emil		5				5						
-	Jens		5				5						
-	Khan		2	2	1		5						
13-05-2013	Kim				5		5						
-	Mathias				5		5						
-	Emil		3		2		5						
-	Jens		4		1		5						
-	Khan		2		3		5						

### Indledning:

Vi skal skrive en webapplikation efter principperne i 3-lags modellen, som gør det muligt for operatør-administratoren at oprette, slette og vise operatørene i systemet. Vi har fået forud defineret de forskellige kriterier der gælder for hver af de enkelte felter. Dette skal forhåbentlig give et mere realistisk program, som giver mere "rigtige" værdier.

### Design:

Designet er opbygget ud fra 3-lags-modellen. Herunder ses et klasse diagram over de java-klasser som er en del af vores program.



Herefter er opstillet en sitemap, som gerne skulle give overblik over hvordan de forskellige jsp sider hænger sammen. De to diagrammer skal selvfølgelig ses i en sammenhæng, men for overskuelighedens skyld er de her delt op.

http://localhost:8080/16\_02324-CDIO4/OperatorAdministration.jsp

## Operatører

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
2	Test 2	bb	2222222222	2222222	Opdater	Fjern
3	Test 3	cc	3333333333	3333333	Opdater	Fjern
4	Test 4	dd	4444444444	4444444	Opdater	Fjern
					Tilføj	

When "tilbage" button is pushed

When wrong input is given

http://localhost:8080/16\_02324-CDIO4/AddOperator.jsp

Identifikation er ikke et tal.  
Navn er ugyldig.  
Initialer er ugyldig.  
CPR er ugyldig.  
Adgangskode er ugyldig.

[Tilbage](#)

## Implementation

Programmet er opdelt i to JSP sider og tre java filer.

### Operator.java

Denne klasse beskriver en operatør. Klassen har fem felter med informationer omkring operatøren samt get og set funktioner for dem alle.

```
public class Operator {

    int identification; // Operatør id (1-99999999)
    String name;        // Operatør navn (2-20 karakterer)
    String initials;    // Operatør initialer (2-3 karakterer)
    String cpr;         // Operatør CPR (10 karakterer, kun tal)
    String password;    // Operatør adgangskode (7-8 karakterer)

    // Standard konstruktør som sætter alle variablene...

    // Standard get og set metoder for alle variablene...

}
```

### Data.java

Denne klasse indeholder en liste over alle operatørene. Klassen tilføjer tre test operatøre automatisk, disse er kun til test formål. Der er mulighed for at returnere en liste over operatørene samt tilføje nye og fjerne eksisterende.

```
public class Data {

    private ArrayList<Operator> operators = new ArrayList<Operator>();

    public Data() {
        operators.add(new Operator(1, "Test 1", "aa", "1111111111", "111111"));
        operators.add(new Operator(2, "Test 2", "bb", "2222222222", "222222"));
        operators.add(new Operator(3, "Test 3", "cc", "3333333333", "333333"));
    }

}
```

```

public Operator[] getOperators() {
    return operators.toArray(new Operator[operators.size()]);
}

public void addOperator(Operator operator) {
    operators.add(operator);
}

public void removeOperator(Operator operator) {
    operators.remove(operator);
}

}

```

## Session.java

Denne klasse beskriver en session. Den indeholder en statisk instans af data klassen som deles mellem alle sessionerne. Der er mulighed for at få en liste over alle operatører, samt tilføje, opdatere og fjerne dem. Klassen gemmer en lokal kopi af listen over operatørerne, da hver operatør identificeres via deres placering i listen. Dette skal bruges når en operatør skal opdateres. Det er nødvendigt at lave det på denne måde, da man ellers ikke kan være sikker på at operatøren stadig har samme placering. Alternativt kunne man have tilføjet et unikt id som ikke kan ændres til hver operatør, og bruge det i stedet for deres placering.

```

public class Session {

    private static final Data data = new Data();
    private Operator[] operators;

    public Operator[] getOperators() {
        operators = data.getOperators();
        return operators;
    }

    public void addOperator(int identification, String name, String initials, String cpr,
        String password) {
        data.addOperator(new Operator(identification, name, initials, cpr, password));
    }
}

```

```

}

public void updateOperator(int index, int identification, String name, String initials,
                           String cpr, String password) {
    operators[index].setIdentification(identification);
    operators[index].setName(name);
    operators[index].setInitials(initials);
    operators[index].setCPR(cpr);
    operators[index].setPassword(password);
}

public void removeOperator(int index) {
    data.removeOperator(operators[index]);
}

}

```

## OperatorAdministration.jsp

Dette er websiden der vises når programmet startes. Den har en instans af et session objekt, som bruges til at udfylde en tabel over alle de eksisterende operatører. Hver linje i tabellen består af en række tekstbokse der viser de forskellige felter fra operator klassen. I enden af hver række er der en opdater og fjern knap, begge knapper sender brugeren videre til den anden JSP side, med den valgte række som argument. I bunden af siden er der altid en ekstra række som kan udfyldes, hvorefter man kan trykke på tilføj knappen. Brugeren vil også i dette tilfælde blive videresendt til den anden JSP side.

```

<jsp:useBean id="s" class="controller.Session" scope="session" />
<jsp:setProperty name="s" property="*" />
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Operatøradministration</title>
</head>
<body>
    <h1>Operatører</h1>
    <table>
        <tr>

```

```

<th>Identifikation</th>
<th>Navn</th>
<th>Initialer</th>
<th>CPR</th>
<th>Adgangskode</th>
<th></th>
<th></th>
</tr>
<%
    controller.Operator[] operators = s.getOperators();
    for (int i = 0; i < operators.length; i++) {
%>
<form action="AddOperator.jsp" method="post" style="display:inline">
    <input type="hidden" value="<%= i %>" name="Index">
    <tr>
        <td><input type="text" value="<%= operators[i].getIdentification() %>"
            name="Identification"></td>
        <td><input type="text" value="<%= operators[i].getName() %>" name="Name"></td>
        <td><input type="text" value="<%= operators[i].getInitials() %>"
            name="Initials"></td>
        <td><input type="text" value="<%= operators[i].getCPR() %>" name="CPR"></td>
        <td><input type="text" value="<%= operators[i].getPassword() %>"
            name="Password"></td>
        <td><input type="submit" value="Opdater" name="button"></td>
        <td><input type="submit" value="Fjern" name="button"></td>
    </tr>
</form>
<%
    }
%>
<form action="AddOperator.jsp" method="post" style="display:inline">
    <input type="hidden" value="-1" name="Index">
    <tr>
        <td><input type="text" name="Identification"></td>
        <td><input type="text" name="Name"></td>
        <td><input type="text" name="Initials"></td>

```



```

        <td><input type="text" name="CPR"></td>
        <td><input type="text" name="Password"></td>
        <td><input type="submit" value="Tilføj"></td>
    <td></td>
</tr>
</form>
</table>
</body>
</html>

```

## AddOperator.jsp

Denne webside håndtere oprettelse, opdatering og fjernelse af operatører. Alle felterne tjekkes for at se om værdierne er gyldige. Hvis de er gyldige, udføres operationen via et kaldt til session klassen, og brugeren videresendes tilbage til den første webside. Hvis mindst en af værdierne er ugyldig, vises en liste med fejlmeddelelserne i stedet for. Brugeren skal selv trykke tilbage for at komme retur.

```

<jsp:useBean id="s" class="controller.Session" scope="session" />
<jsp:setProperty name="s" property="*" />
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Operatøradministration</title>
</head>
<body>
    <%
        java.util.ArrayList<String> errors = new java.util.ArrayList<String>();
        Integer index = null;
        Integer identification = null;
        try {
            index = Integer.parseInt(request.getParameter("Index"));
        } catch (Exception e) {
            errors.add("Indeks er ikke et tal.");
        }
        try {

```

```

identification = Integer.parseInt(request.getParameter("Identification"));
if (identification < 1) {
    errors.add("Identifikation er mindre end 1.");
}
if (identification > 99999999) {
    errors.add("Identifikation er større end 99999999.");
}
} catch (Exception e) {
    errors.add("Identifikation er ikke et tal.");
}

String name = request.getParameter("Name");
String initials = request.getParameter("Initials");
String cpr = request.getParameter("CPR");
String password = request.getParameter("Password");
if (!name.matches("^.{2,20}$")) {
    errors.add("Navn er ugyldig.");
}
if (!initials.matches("^.{2,3}$")) {
    errors.add("Initialer er ugyldig.");
}
if (!cpr.matches("^[0-9]{10}$")) {
    errors.add("CPR er ugyldig.");
}
if (!password.matches("^.{7,8}$")) {
    errors.add("Adgangskode er ugyldig.");
}
if (errors.size() == 0) {
    if (index >= 0) {
        if (request.getParameter("button").equals("Fjern")) {
            s.removeOperator(index);
        } else {
            s.updateOperator(index, identification, name, initials, cpr, password);
        }
    } else {
        s.addOperator(identification, name, initials, cpr, password);
    }
}

```

```

        response.sendRedirect("OperatorAdministration.jsp");
    }
    for (String line : errors) {
%>
<p><%= line %></p>
<%
    }
%>

<p><a href="javascript:history.go(-1)">Tilbage</a></p>
</body>
</html>

```

### Test:

Som en test af programmet er der tildels blevet brugt black-box testing, da vi til dels er nye i jsp. Derfor er programmet testet ud fra alle de scenarier vi selv har kunnet tænke på. Programmet er endt med at fungere uden fejl, da vi har rettet alle dem vi fandt.

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
2	Test 2	bb	2222222222	2222222	Opdater	Fjern
3	Test 3	cc	3333333333	3333333	Opdater	Fjern
123456988	ismushansenmogensens	rhms	123456789	12tre4567	×	Tilføj

Identifikation er større end 99999999.

Navn er ugyldig.

Initialer er ugyldig.

CPR er ugyldig.

Adgangskode er ugyldig.

[Tilbage](#)

Ovenfor ses et tilfælde hvor ingen af de indtastede værdier er gyldige, og man får derfor at vide, hvor fejlen(e) ligger.

## Operatorer

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
2	Test 2	bb	2222222222	2222222	Opdater	Fjern
3	Test 3	cc	3333333333	3333333	Opdater	Fjern
12	rasmushansen	rh	1234567890	12tre45	x	Tilføj

## Operatorer

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
2	Test 2	bb	2222222222	2222222	Opdater	Fjern
3	Test 3	cc	3333333333	3333333	Opdater	Fjern
12	rasmushansen	rh	1234567890	12tre45	Opdater	Fjern
					Tilføj	

Ovenfor ses et tilfælde hvor alle de indtastede værdier er gyldige, og man succesfuldt får tilføjet en operator.

## Operatorer

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
2	Test 2	bb	2222222222	2222222	Opdater	Fjern
3	Test 3	cc	3333333333	3333333	Opdater	Fjern
12	rasmushansen	rh	1234567890	12tre45	Opdater	Fjern
					Tilføj	

## Operatorer

Identifikation	Navn	Initialer	CPR	Adgangskode		
1	Test 1	aa	1111111111	1111111	Opdater	Fjern
					Tilføj	

Ovenfor ses blot hvordan "Fjern" funktionen virker. Her bliver operatørene helt slettet.

**konklusion:**

Programmet gør en administrator i stand til at oprette, fjerne og redigere forskellige operatører. Både hvis man opdatere og opretter en operatør, så vil man få fejl hvis man ikke følger de regler der er inden for dataen i de forskellige felter, med undtagelse af identifikationsnummeret, hvor flere personer kan have samme identifikationsnummer.

Hvis der endelig manglede noget, så var det at vi når en administrator får en fejl, så burde systemet oplyse ham om hvilke kriterier der er for det datafelt hvor der er fejl i.