DAT108 Oblig2 h20 - Lambda og Web

Sist oppdatert av Lars-Petter Helland 16.09.2020

Øvingen skal gjennomføres i grupper på **2-4** studenter. (Studenter som har fått tillatelse til å levere alene registrerer seg også som en gruppe!)

Dere danner selv grupper i Canvas ved å registrere dere i en av de 70 forhåndsdefinerte gruppene «DAT108 Oblig2 gruppe x». Legg helst til alle studentene i gruppen samtidig. (Gå inn på «Personer» | «DAT108 Oblig2 gruppe», og legg til dere selv i en tom gruppe).

NB! Gruppemedlemmer må tilhøre samme kohort. Ellers er litt av vitsen med kohorter borte!

NB! Det er nytt gruppesett for hver oblig. Dvs. at dere må registrere gruppen på nytt i gruppesettet for ny oblig selv om dere er samme gruppe!

Øvingen har veiledning torsdagene 10., 17. og 24. september.

Innleveringsfrist er søndag 27. september. Vi har som mål å rette innleveringene og godkjenne innen 2 uker.

Innleveringen er en zip i Canvas. NB! Zip-en skal hete **Oblig2_gr17.zip** for gruppe 17 osv (dere forstår) ... Denne skal inneholde:

1. Et pdf-dokument som inneholder:
    a) en liste av hvem som er med i gruppen (for å unngå gratispassasjerer som melder seg inn i gruppe uten at det er avtalt).
    b) Skjermutskrift fra kjøringer av programmene, slik at vi kan se at de virker.
    c) Svar på et par "teorispørsmål" i Oppgave4.
2. Løsning på oppgavene: **Ett** Eclipse-Java-prosjekt for lambda- og streams-oppgavene (Oppgave1, 2 og 3), og **ett** Eclipse-JEE-prosjekt for weboppgaven (Oppgave4).

Oppgave 1 - Lambda-uttrykk

a)

Klasse med main()-metode: **Oppg1a**

Anta at vi har en liste av heltallsstrenger:

```
List<String> listen = Arrays.asList("10", "1", "20", "110", "21", "12");
```

Oppgaven din er å skrive disse ut på skjermen sortert etter tallverdi, dvs. 1, 10, 12, 20, 21, 110. Du skal bruke en av de innebyggede sort()-metodene i Collections til å gjøre sorteringsjobben. Bruk et lambdauttrykk til å representere en Comparator som sort() kan benytte seg av.

b)

Klasse med main()-metode: **Oppg1b**

Oppgaven din er å lage en metode som utfører en heltalltsberegning på to heltall den mottar som parametre. Hvilken beregning som utføres angis med en tredje parameter. Noe slikt:

```
public static int beregn(int a, int b, ???) {
    ...
}
```

Deretter skal du bruke denne metoden i et par eksempler.

Vis (i main) eksempler på bruk av denne metoden til å beregne:

i. Summen av 12 og 13
ii. Den største av -5 og 3
iii. Avstanden (absoluttverdien av differansen) mellom 54 og 45

Den tredje parameteren ved kall til beregn skal alltid angis som en variabel, f.eks:

```
int sum = beregn(12, 13, summerFunksjon);
```

Tips: BiFunction

Oppgave 2 - Lambda-uttrykk

Klasse med main()-metode: **Oppg2**
Andre: **Ansatt** (klasse), **Kjonn** (enum)

Ansatte har fornavn (String), etternavn (String), kjonn (Kjonn), stilling (String) og aarslonn (int).

Vi tenker at vi har en liste av ansatte og skal lage en metode som utfører lønnsoppgjør (oppdaterer de ansattes lønn etter en viss algoritme).

Denne metoden ligger i Oppg2, og ser slik ut:

```
private static void lonnsoppgjor(List<Ansatt> ansatte, ???) ...
```

Hvordan lønnen endres kan variere, f.eks. :

i. Et fast kronetillegg
ii. Et fast prosenttillegg
iii. Et fast kronetillegg hvis du har lav lønn
iv. Et fast prosenttillegg hvis du er mann

osv...

Vi ønsker at de ulike måtene **den nye lønnen skal beregnes** oppgitt som en funksjonsparameter til metoden lonnsoppgjor(...) slik at lonnsoppgjor(...) blir generell og uavhengig av type beregning.

Skriv en løsning med klassene Oppg2 (med main)), Ansatt og Kjonn.

Bruk lambda-uttrykk (i main) for de ulike typene lønnsøkning, og lag en kjørbar løsning med en liste av 5 ansatte der du tester ut de fire ulike typene lønnsøkning i listen over (og evt. flere hvis du vil). Du må gjerne lage en hjelpemetode i Oppg2 som skriver ut hele listen, f.eks:

```
private static void skrivUtAlle(List<Ansatt> ansatte) ...
```

Tips: La parameteren til lonnsoppgjor(...) være av typen Function<Ansatt, Integer> (tilsv. en f(ansatt) := beregning av ny lønn). Dette tilsvarer om du selv hadde definert en funksjonell kontrakt med metoden Integer beregnNyLonn(Ansatt a). I lønnsoppgjøret settes da den nye lønnen med a.setLonn(<beregning av ny lønn>) for hver av de ansatte.

Oppgave 3 - Streams

Klasse med main()-metode: **Oppg3**

Dere kan bruke Ansatt-klassen og listen av ansatte (kopier over til Oppg3.java) fra forrige oppgave. Ansatt-listen blir input til alle delspørsmålene.

Skriv en klasse Oppg3 med en main()-metode som inneholder løsningen på delspørsmålene.

Løsningene skal være funksjonelle, dvs. bruk stream(), map(), filter(), reduce(), osv. Alle løsningene/svarene skal lagres i variabler som deretter skrives ut på skjermen.

a) Lag en ny liste som kun inneholder etternavnene til de ansatte.

b) Finn ut antall kvinner blant de ansatte.

c) Regn ut gjennomsnittslønnen til kvinnene.

d) Gi alle sjefer (stilling inneholder noe med "sjef") en lønnsøkning på 7% ved å bruke streams direkte i stedet for metoden du laget i Oppg2. Skriv ut listen av ansatte etter lønnsøkningen.

e) Finn ut (true|false) om det er noen ansatte som tjener mer enn 800.000,-

f) Skriv ut alle de ansatte med System.out.println() uten å bruke løkke.

g) Finn den/de ansatte som har lavest lønn.

h) Finn ut summen av alle heltall i [1, 1000> som er delelig med 3 eller 5.

Oppgave 4 - Komme i gang med Java webapplikasjoner

Før dere begynner på denne oppgaven må dere ha gjort følgende:

- Lastet ned (og pakket ut) en webtjener på egen PC. Denne webtjeneren skal være enten en **TomEE plus 8.0.4** (http://tomee.apache.org/download-ng.html) eller en **Tomcat 9.0.xx** (https://tomcat.apache.org/download-90.cgi).

- Lastet ned (og pakket ut) **Eclipse IDE for Enterprise Java Developers** 2020-06 (https://www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-enterprise-java-developers).

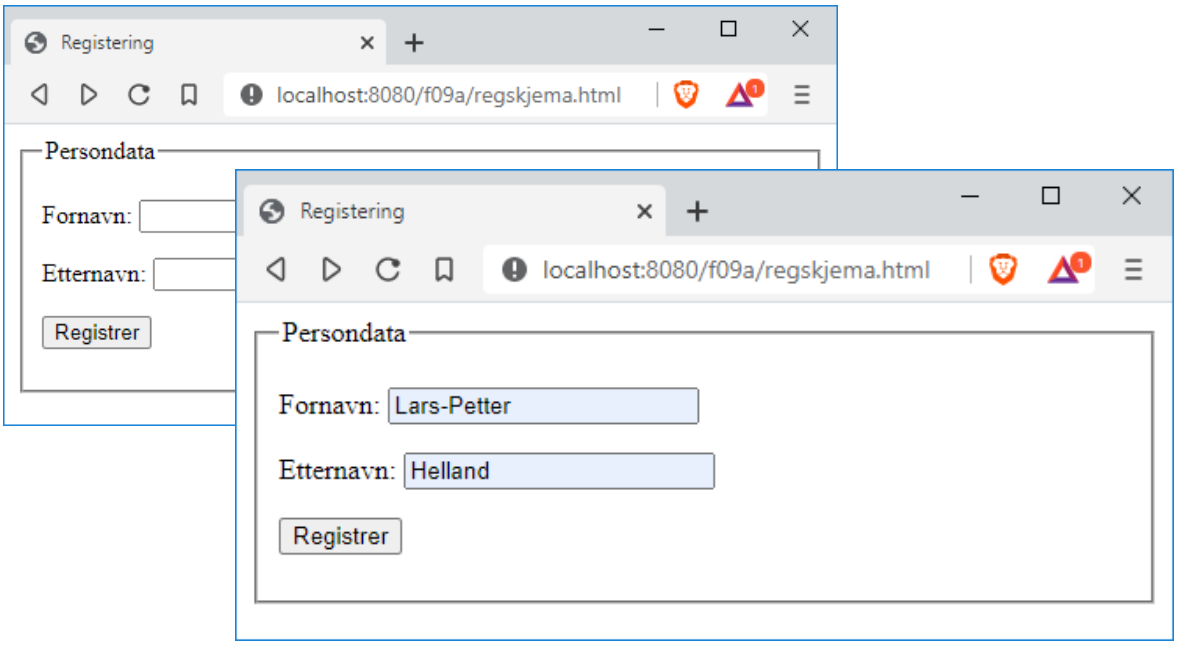
- **Registrert webtjeneren** i Eclipse (Window | Preferences | Server | Runtime Environments | Add... | Apache Tomcat v9.0 | Finish.

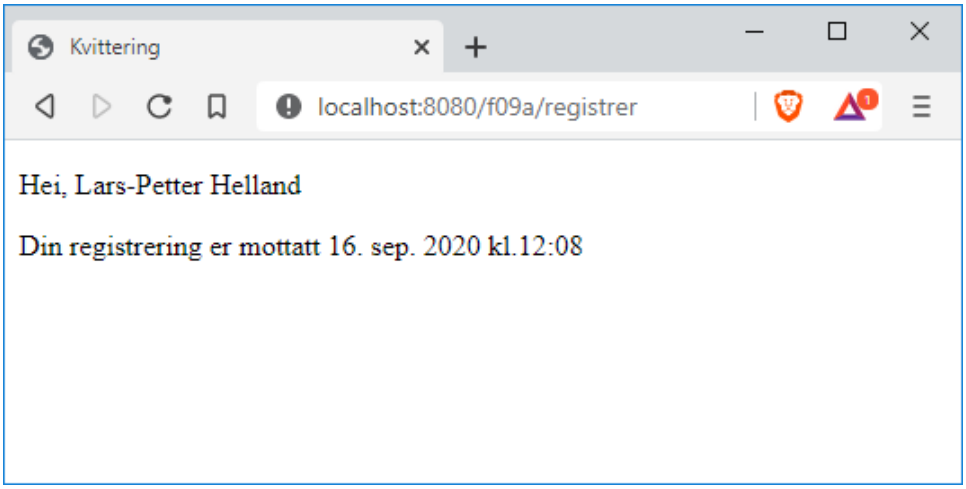
- Hvis dere kjører TomEE (og ikke kun Tomcat) må dere også **"skru ned" Java-versjonen** for webprosjekter. Dette kan gjøres en gang for alle med (Window | Preferences | Java | Compiler | Compiler compliance level: **settes til 11**.

...

Så til oppgaven:

Dere skal lage et lite web-prosjekt som har én (statisk html-) side for inntasting av noe data, og én (dynamisk generert) side som gir en tilbakemelding på det du har tastet inn. F.eks. slik:





...

Forslag til fremgangsmåte:

- Opprett et "**Dynamic Web Project**" og kall dette f.eks. "**Obl2Oppg4**".
- Default context root blir da «Obl2Oppg4». Endre denne til «**obl2**». (Properties | Web Project Settings).
- Opprett og lag den statiske html-siden og legg denne under **WebContent** i prosjektet. Kall den gjerne index.html.
- Tid for prøvekjøring. Kjør prosjektet (Run As | Run on Server). Test det fra en skikkelig nettleser (Brave, Safari, Opera, etc...)
- Opprett en Servlet som tar imot dataene fra registreringsskjemaet (html-siden), og lag doPost()-metoden. Forslag til innhold (fyll ut det som mangler):

```
String fornavn = ??? Fornavnet som brukeren tastet inn.
String etternavn = ??? Etternavnet som brukeren tastet inn.
String timestamp = ??? Dato og klokken akkurat NÅ. Fint formatert.

response.setContentType("text/html; charset=ISO-8859-1");

PrintWriter out = response.getWriter();

out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<meta charset=\"ISO-8859-1\">");
out.println("<title>Kvittering</title>");
out.println("</head>");
out.println("<body>");
out.println("<p>Hei, " + fornavn + " " + etternavn +"</p>");
out.println("<p>Din registrering er mottatt " + timestamp + "</p>");
out.println("</body>");
out.println("</html>");
```

- Tid for prøvekjøring igjen. Kjør prosjektet (Run As | Run on Server). Test det fra en skikkelig nettleser (Brave, Safari, Opera, etc...)
- Legg merke til hva URL-ene er for de to sidene. Hvordan kan disse endres til det du ønsker? (tips: context root og servlet mapping)
- Kunne vi endret applikasjonen til å bruke GET/doGet() i stedet for POST/doPost()? Hvordan?

Svaret på de siste to spørsmålene kan skrives i pdf-en dere skal levere.