

## Opgaver uge 1

### Tirsdag 29. januar 2008

2008-01-29

#### Formål med opgaverne

Efter disse øvelser skal du kunne bruge Eclipse til at oversætte, køre og eksperimentere med Java-programmer, og til at rette i et eksisterende Java-program så det løser et nyt problem.

Du skal også kunne oprette et CVS-repository i dit hjemmekatalog på serveren `ssh.itu.dk` og gemme et Eclipse Java-projekt i det repository.

#### Gør dette først

- Installér Eclipse, for eksempel ved brug af vejledningen fra kurset Objektorienteret Softwarekonstruktion. Den findes på hjemmesiden for det kursus, <http://wiki.lundogbendsen.dk/x/KAA1>, under Noter til uge 1.

Eclipse kører på MS Windows, Linux og MacOS, og kræver at Java er installeret, helst JDK version 6, men den Java-version du installerede i efterårets kursus skulle være helt tilstrækkelig.

Hvis du arbejder på din egen laptop bør du have dit Eclipse workspace på den lokale disk, ikke på ITU's netværksdrev. Når man bruger CVS til versionsstyring har man automatisk en backup på ITU's servere, som it-afdelingen yderligere tager backup af.

#### Opgave 1.1: Lav et Java-projekt i Eclipse

Lav et Java-projekt i Eclipse ved at lave i hvert fald trin 1–7 i *Frivillig opgave 1.2* fra kurset Objektorienteret Softwarekonstruktion, uge 1.

#### Opgave 1.2: Opsætning af CVS repository

Når man arbejder på software i en projektgruppe er det ekstremt nyttigt at bruge versionsstyring, hvor hovedversionen af alle filerne ligger på en fælles server. Til det formål anvendes CVS (Concurrent Versioning System) sammen med Eclipse og ITU's filserver. Et nyere alternativ til CVS er SVN (Subversion), men det er meget omstændeligt at sætte op under Eclipse hvis det skal virke sammen med ITUs server.

**Opret et CVS repository på `ssh.itu.dk`** Lad os sige at bruger `sestoft` vil oprette et CVS repository med navn `faapcvs` i sit hjemmekatalog på ITU-serveren `ssh.itu.dk`.

- Log ind på `ssh.itu.dk`, der er en Unix-maskine hvor du kan lægge dine ITU-filer.  
Fra Windows kan dette gøres med Secure Shell, der giver krypteret adgang til serveren via internettet. Secure Shell er installeret på alle ITU's Windows-maskiner.  
Til din egen laptop kan du downloade installationsfilerne fra:  
<http://www.itu.dk/sysadm/software/public/ssh/SSHSecureShellClient-3.2.9.exe> hvor du bliver bedt om ITU's intranet login, som er user `fyn` og password `DNS&BIND`  
Se gerne vejledning på <https://www.itu.dk/sysadm/dokuwiki/doku.php?id=topics:ssh-win>
- Når du er logget ind på `ssh.itu.dk` skal du udføre følgende kommandoer:

```
# mkdir faapcvs
# chmod g+srwx faapcvs
# cvs -d /import/home/sestoft/faapcvs init
```

Ovenstående kommandoer betyder dette:

- Kommandoen `mkdir` betyder "make directory"; den opretter et katalog, det som Windows kalder en mappe, til at have alle CVS-filerne i.
- Kommandoen `chmod` betyder "change mode"; den ændrer skriverettigheder på kataloget, sådan at gruppen (g) kan læse (r), skrive (w) og eksekvere (x) filer i kataloget, og disse rettigheder arves af alle nye filer og underkataloger der oprettes (s=sticky, rettighederne "klistrer" til nye filer. Det bliver vigtigt senere når I skal dele filerne i en projektgruppe.
- Kommandoen `cvscvs -d /import/home/sestoft/faapcvscvs init` opretter et antal administrative filer i `faapcvscvs` kataloget. Brugernavnet `sestoft` skal selvfølgelig erstattes med dit eget, også i punkterne nedenfor.

Nu har du et CVS repository på serveren `ssh.itu.dk` som du kan tilgå fra hele internettet.

Log ud fra `ssh.itu.dk` igen.

### Få fat i CVS repositoret fra Eclipse

- Inde i Eclipse kan du få fat i CVS repositoret på denne måde:

```
File
-> New
-> Other
-> CVS
-> CVS Repository Location
-> Host: ssh.itu.dk
    Repository path: /import/home/sestoft/faapcvscvs
    User: sestoft
    Password: <dit ITU mail password>
    Connection type: extssh
    Save password: Ja
-> Finish
```

Se illustration på en slide fra forelæsning 1 i BFÅP.

- Nu kender Eclipse til dette CVS repository. For at tjekke dette, kan du i Eclipse gå til CVS Repository Exploring, hvor du gerne skulle kunne se det under navnet  
:extssh:sestoft@ssh.itu.dk:/import/home/sestoft/faapcvscvs

### Læg dit Eclipse Java-projekt ind i CVS repositoret

- Læg nu sit Eclipse Java-projekt fra opgave 1.1 ind i CVS repositoret. I Eclipse Resource View skal du:

```
Højreklikke på projektet
-> Team
-> Share project
-> CVS
-> Use existing ... :extssh:sestoft@ssh.itu.dk:/import/...
-> Use project name as module name
-> Finish
```

Nu ligger projektet godt og trygt på `ssh.itu.dk` hvor it-afdelingen sørger for at der bliver taget backup osv. Desuden vil CVS sørge for at alle gamle versioner af dine filer bliver gemt.

Det der ligger på din lokale pc/mac skal fremover opfattes som en *kopi* af hovedversionen, som er den der ligger i CVS repositoret på serveren `ssh.itu.dk`.

Derfor skal du hver gang synkronisere den lokale kopi med repositoret inden du begynder at redigere i den, og synkronisere igen når du er færdig.

- Senere skal vi se hvordan en projektgruppe kan dele et repository. På den måde kan alle gruppemedlemmer automatisk se de ændringer som de andre har lavet.

**Opgave 1.3 – nogle grafbegreber**

(i) En graf er plan (“planar”) hvis den kan tegnes i planen (dvs på den ene side af et stykke papir) uden at kanterne krydser hinanden.

Er grafen Hvad er det? fra forelæsning 1 plan? Læs definitionen omhyggeligt og se godt efter.

(ii) En graf farves (“is colored”) ved at man tildeler hver knude en farve, sådan at to knuder der er forbundet med en kant ikke har samme farve.

En berømt sætning i grafteori lyder: *Enhver plan graf kan farves med kun fire farver*. En konsekvens er at atlaser kan trykkes med fire farver (hvis hver stat består af ét sammenhængende landområde).

Denne tilsyneladende simple påstand blev fremsat allerede i 1852 men blev først bevist i 1976 af Appel og Haken, men deres bevis var så omfattende at det krævede brug af en computer, og derfor var det længe omstridt blandt matematikere.

Kig på grafen Hvad er det? fra forelæsning 1 og lav en farvning af den med kun fire farver.

(iii) Hvad betyder det hvis et vejnet (hvor kanter er vejstumper og knuder er vejkryds eller vejpunkter) udgør en plan graf? Tror du det danske vejnet udgør en plan graf?

(iv) Er det naturligt at et vejnet er en orienteret graf eller en ikke-orienteret graf? Hvorfor? Er det naturligt at et vejnet er acyklisk?

(v) En graf kaldes sammenhængende (“connected”) hvis der er en sti fra enhver knude til enhver anden knude. Er det danske vejnet sammenhængende? Hvorfor/hvorfor ikke?

**Opgave 1.4 – repræsentation af vejdatasættet som graf i Java**

En graf kan repræsenteres på mange måder i Java. I vejdatasættet er en knude grundlæggende identificeret ved sit løbnummer 1, 2, ... fra `kdv_node_unload.txt` filen:

```
public class Node {
    public final int index;
    ...
}
```

En orienteret kant indeholder referencer til de to knuder  $v_1$  og  $v_2$  den forbinder. Her står  $v$  for vertex, selv om knudeklassen hedder `Node`; de to engelske betegnelser for knude bruges på skift. Et separat felt angiver om kanten er orienteret fra  $v_1$  til  $v_2$  eller omvendt, eller ikke-orienteret (tovejs):

```
public class Edge<N extends Node> {
    public N v1;
    public N v2;
    protected byte direction;
    public static final byte FORWARD =1;
    public static final byte BACKWARD =2;
    public static final byte BOTH = FORWARD | BACKWARD;
    ...
}
```

Selve grafen er repræsenteret med en såkaldt kantliste: for hver knude i grafen er der en liste af de orienterede kanter der hører til knuden:

```
public class Graph<E extends Edge<N>, N extends Node> {
    public ArrayList<ArrayList<E>> edges;
    ...
}
```

Dvs. hvis `graph` er en `Graph<Edge<Node>, Node>` og `node` er en `Node`, så er `graph.edges.get(node.index)` en arrayliste af kanter der hører til knuden `node`. Bemærk at `graph.edges.get(0)` er `null` fordi der aldrig er en knude med løbenummer 0.

Programteksten til at indlæse vejdatasættet findes på kursushjemmesiden. Studér og kød denne programtekst og besvar følgende spørgsmål.

- (i) Hvad opnås der ved at bruge typeparametre `N` til `Edge<N>`, og typeparametrene `E` og `N` til `Graph<E, N>`?
- (ii) Klassen `KrakNode` er en subklasse af `Node`. Hvilke ekstra instansfelter og metoder er der i forhold til `Node`?
- (iii) Klassen `KrakEdge` er en subklasse af `Edge<KrakNode>`. Hvilke ekstra instansfelter og -metoder er der i forhold til `Edge<KrakNode>`?
- (iv) Brug `KrakLoader` til at indlæse Valby-delgrafem, og beregn den samlede længde af kanterne (vejstumperne) i Valby.
- (v) Brug `KrakLoader` til at indlæse Valby-delgrafem, og lav en liste af alle vejnavne i Valby uden gentagelser. Vink: En `collection` er nyttig her.