

1 Kodegenerering fra Tau Suiten

For å generere Javakode eller en annen form for programmeringskode ut i fra Tau suiten, er det visse ting som må være utført.

1. En UML modell må eksistere og være korrekt.
2. Tau må være klargjort for generering av kode.
3. Du må angi PATH til hvor du skal lagre filene som genereres.
4. Du må angi hvilken editor du skal bruke.

Hver av disse stegene vil bli gått gjennom i dette dokumentet.

1.1 En uml modell må eksistere og være korrekt

Om du fremdeles modellerer i “Analysis” eller “SystemDesign” fasen, er det nå tid for å opprette “ObjectDesign” fasen. “ObjectDesign” fasen må inneholde alle klassediagrammer du skal generere kode ut i fra. For å flytte diagrammene fra en fase til en annen gjør da slik:

- Opprett “ObjectDesign” fasen ved å markere konfigurasjon versjon, den ligger mellom prosjektnavnet ditt og den fasen du nå er i (se etter V1). Velg File | New | Phase Version(s)..
- I vinduet som kommer opp, marker “ObjectDesign”, og trykk OK.
- Lag en ny pakke i den fasen du skal flytte diagrammer til, dvs. “ObjectDesign” fasen. Den skal ha samme navn som den pakken i den fasen du flytter diagrammer fra.
- Velg så den fasen du flytter til, f.eks. ObjectDesign, gå så på menyen, velg Utilities | Merge From Previous Phase, et nytt vindu dukker da opp. I det vinduet, velg så fasen du ønsker.
- Når du har gjort det, så dukker det opp et nytt vindu. I dette vinduet navigere du deg frem dit du ønsker, og marker de klassediagrammene du vil ha med videre til “ObjectDesign” fasen. **MERK!** Husk å ta med alle diagrammer som har CDM foran seg, da disse inneholder informasjon om attributter og metoder.
- Når de er markerte velg, Version | Merge på menyen, og diagrammene vil bli flyttet over.

Det første du gjør nå er å sjekke diagrammet ditt. Marker den pakken der du har de diagrammene du ønsker å generere kode ut i fra, gå på menyen Check | Global Model. Alternativt, marker klassediagrammet ditt og velg Check | Local Model

Her er det **STOR** mulighet for at du får mange “Warnings” og “Errors”. Alle “Errors” må være borte før du kan generere kode. Om du ser på figur 1, så ser du hvordan et korrekt klassediagram, klar for å generere kode, skal se ut. I klassediagrammet ditt er det noen viktige ting å merke seg for å unngå “Errors”:

1. Alle klasser **skal** ha et navn!

2. Alle attributter **skal** ha:

- en **synlighet** (- private, # protected, + public, \$ static).
- en gyldig **datatype**, f.eks. String, int.
- Slik kan du deklare et attributt: +attributtnavn:datatype = verdi, der “verdi” er en initialisering av attributtet.

3. Alle metoder **skal** ha:

- en **synlighet** (- private, # protected, + public, \$ static).
 - Metoder kan ha:
 - * parametre: +metodeNavn(**navn: String**)
 - * returverdi: +metodeNavn():**String**
- En metode kan da se slik ut: +metodeNavn(navn: String):String

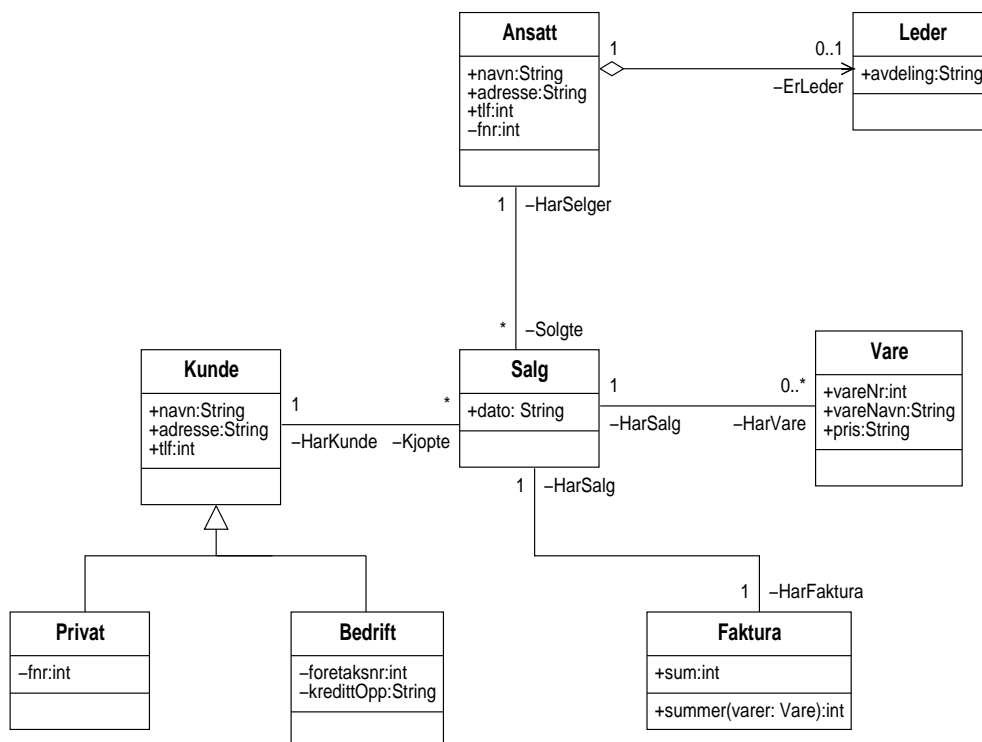
4. Assosiasjoner **skal** ha følgende:

- Rollenavn i begge ender, se f.eks. assosiasjon mellom klassen Kunde og klassen Salg i figur 1. Denne assosiasjonen vil føre til at en “Kunde” vil få en vektor med navn “kjopte”, der vektoren vil inneholde pekere til objekter av typen “Salg”. “Salg” vil få en peker med navn “Kunde”. Om det blir generert en peker eller vektor avhenger av kardinaliteten.
- Kardinalitet som angir en begrensning på hvordan det kvantitative forholdet mellom objektene skal være.

Ta for deg hver enkelt feilmelding du får ved å utføre “Check” og forandre den til slik som beskrevet over for de ulike delene. Når du ikke har flere “Errors” igjen, er du klar for neste steg.....

En ting som dere bør **merke** dere er at Tau har en lei tendens til å sette sine egne synligheter på attributter og metoder. Dvs. man kan få attributter eller metoder med to synlighetsangivelser (f.eks -+**navn:String**).

Figur 1 Klassediagramm



Når dere sjekker diagrammene for feil vil ikke dette bli fanget opp, men ved kodegenereringen så vil det dukke opp en uforståelig feilmelding. Gå da gjennom klassediagrammet og fjern de synlighetsangivelsene som ikke skal være der.

1.2 Tau må være klargjort for generering av kode.

For å få generert java kode, eller en annen type programmeringskode, må man gjøre Tau klar for dette. Det betyr at man må si til Tau at nå skal jeg generere f.eks Javakode ut i fra mitt klassediagram. Gjør følgende:

1. I hoved browseren, marker prosjektet ditt.
2. Velg Utilities | Show Active Modules

Du får nå opp et nytt vindu, om “Java Generation” modulen ikke er listet opp der, så må du aktivere den. Dette gjøres slik:

1.2.1 Java Code Generator Module

1. Marker prosjektet ditt.
2. Velg Utilities | Customize...
3. I “Category” dialog vinduet, velg:
 - (a) Module Customization, trykk “Next”
 - (b) Nå velger du “Project:dittProsjektnavn”, trykk “Next”
 - (c) Trykk “Finish” og et nytt vindu dukker opp.
4. I “Module Availability” dialog vinduet, velg;
 - (a) Edit | New, et nytt vindu dukker opp.
 - (b) Velg “Java Code Generation”, trykk Ok.
 - (c) Velg File | Save
 - (d) Lukk “Module Availability” vinduet.
5. I hoved browseren, marker “Corporate” prosjektet og marker så ditt prosjekt igjen.
6. Velg Utilities | Show Active Modules, dette for å se at den ble aktivisert.

1.2.2 Legg til JDK klassene

1. Marker prosjektet deres.
2. Velg View | Customization
3. Dobbelklikk på filen modules
4. I “Module Availability” editoren, velg Edit | New
5. Velg JDK 1.2 (Small) modulen og trykk Ok. **MERK!** Velger du full isteden for small her, vil du bli sittende i noen timer å vente på at alle java pakkene blir hentet frem.
6. Velg File | Save
7. Avslutt “Module Availability” editoren.
8. I browseren, velg View | Default
9. Beveg deg så opp til Corporate prosjektet, og så ned til “ObjectDesign” fasen i prosjektet ditt. (Dette aktiverer modulen)
10. Velg Java | Import JDK 1.2 Java (small). **MERK!** small.

Sitt ned å slapp av, i de neste minuttene vil UML suiten legge mange pakker til "ObjectDesign" fasen din. Når dette er gjort er du nære, veldig nære å få generert kode.

1.3 Du må angi PATH til hvor du skal lagre filene som genereres

Når man skal gå fra Objektanalyse fasen til Implementasjon fasen, vil det bli generert filer av modellen som er laget. For at disse filene skal bli tilgjengelige for å lese/editere må man gjøre følgende. Velg prosjektnavnet ditt i browseren. Velg så File | Properties | Edit, et nytt vindu kommer da opp. I det vinduet er det et felt med navnet "File System Path Part". Skriv da inn følgende "/hom/brukernavn/prosjektnavn". Dette for at editoren skal finne filene. Brukernavn er her ditt personlige brukernavn og ikke prosjektkontoens brukernavn.

1.4 Du må angi hvilken editor du skal bruke

Når man skal begynne å editere på filene som man har fått generert, må man velge en editoren, f.eks emacs. Velg prosjektnavn under corporate. Gå på browseren Option | editor, man får da opp et nytt vindu, skriv så inn hvilken editor man skal bruke. Når dere skifter på drop down menyen "Context", så har begge alternativene den editoren dere ønsker.

2 Selve genereringen

Nå er øyeblikket kommet. Marker konfigurasjon versjonen i prosjektet ditt (den med (V1) bak).

1. Velg File | New | Phase Version(s).
2. I det nye vinduet, velg Implementation og trykk OK.
3. Marker Implementation fasen
4. Velg Java | Generate | Specific New Packages
5. I det nye vinduet finn pakken din på listen og trykk OK.

Et nytt vindu kommer da opp. Dette vinduet forteller deg om fremgangen i genereringen. Om du har sluntret unna med designet av modellen og ikke har utført sjekker på modellen din er dette stedet du får svi.

Om du (som har gjort dette rett) har lyst til å se på filene dine, er det bare å dobbelklikke på de.

3 Mer informasjon

Mer informasjon vedrørende verktøyet Tau og hvordan man bruker det finner dere under *Tau Suiten* som dere finner link til fra in219 siden.

LYKKE TIL