1 CLASSPATH

Dette er kun beskrevet for utviklingsmiljøet på IfI og ikke for windows plattformer.

For at java kompilator og java run time environment skal kunne finne alle filene i Hospital Scheduling System (HSS) er det nødvendig å sette hjemme-CLASSPATH i en fil som heter .envir. Denne filen ligger under roten i området ditt;

/hom/brukernavn/.envir

der brukernavn er **ditt** brukernavn. Åpne denne filen i en editor og legg til følgende linje:

setenv CLASSPATH /hom/brukernavn/inf3120\: $CLASSPATH

under linjen:

. /local/lib/setupfiles/envir # Setter standard variable ved IfI

Bytt ut brukernavn med ditt **eget** brukernavn, samt endre katalogen inf3120 om du valgte å ikke opprette denne når du lagret og pakket ut filene. Sett i så fall inn PATH'en til den katalogen du valgte å lagre filene i.
Logg ut av linux og logg på igjen for at endringene skal tre i kraft. Du er nå klar for å kompilere og kjøre programmet. Dette gjøres ved å gå inn i katalogen inf3120 der filen **HSS.java** er og skrive følgende i terminalvinduet:

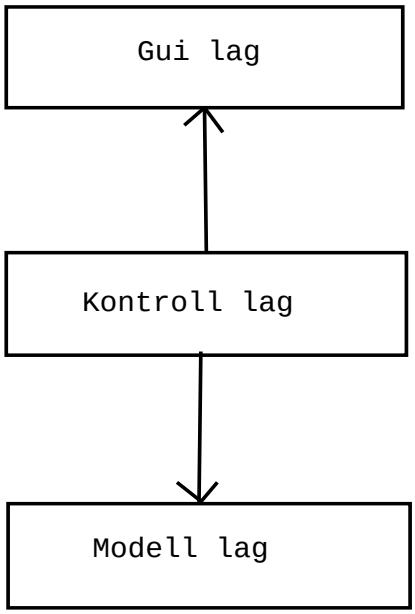
javac HSS.java

2 Struktur

Programmet HSS er byggd opp i tråd med 3-lags arkitektur prinsippet. Dvs. at programmet er delt opp i tre ulike lag som kommuniserer sammen. Se figur 1 for en grafisk fremstilling av arkitekturen. De tre lagene er internt bygd opp av Java filer. Det er disse filene som dere skal jobbe på.

Ved å konstruere programmet på en slik måte blir det enklere for utviklere (dere) å videreutvikle, vedlikeholde og gjenbruke deler eller hele HSS. Som vist i figuren er det piler som viser hvilket lag som kan initiere kommunikasjon med hvem, og bare disse kan initiere kommunikasjon.

Figur 1: 3-lags arkitekturen til HSS



- **GUI-laget**
  Gui-lagets ansvar i HSS er å bygge opp grensesnittet som brukerne kommuniserer med. Kommunikasjonen mellom GUI og bruker skjer gjennom "drop down boxes", tekstfelter og knapper. Gui-laget kjenner kun til seg selv i dette systemet.

- **Kontroll-laget**
  Kontroll-lagets ansvar i HSS er å gjøre grensesnittet og forretningsmodellen uavhengig av hverandre. Klassene i kontroll-laget fanger opp når det trykkes på en knapp i grensesnittet og utfører så den valgte handling. Kontroll-laget kjenner til både Gui-laget og Modell-laget.

- **Modell-laget**
  I dette laget er klassene som representerer entitetene til sykehuset implementert. Dvs. dette er business-logikken. Modell-laget innholder klasser som er entitetene i problemområdet til HSS. Ved å gjøre modell-laget uavhengig av resten av systemet kan det enkelt utvides for å tilby mer eller endret funksjonalitet. Modell-laget kjenner kun til seg selv.

Rent teknisk er disse tre lagene fordelt på hver sin katalog (Gui, control, model). Alle filene i en spesifikk katalog tilhører samme pakke. Pakke

(package) er et uttrykk i Java som kan benyttes til dette formålet. Ved å benytte denne muligheten gjør det oss i stand til å implementere 3-lags arkitekturen gjennom å importere (import) våre egne pakker.

3 Programmets gang

Ved programmets start lages det første grensesnittet (Gui-laget) mot brukeren (HssMain). Når brukeren trykker på en av knappene (JButtons) i dette vinduet trigges det en metode i kontroll-laget som styrer alt som skjer i systemet. Dette er gjentagende for alle hendelser som skjer i systemet som er styrt av knappene. Når brukeren trykker på hvilken som helst knapp i HSS er det konsekvent en metode i kontroll laget som kjøres. Denne metoden fordeler så ansvar til alle de tre lagene i systemet.

Skiving og lesing til fil skjer gjennom bruk av "Serializable" begrepet Java tilbyr. Alle klassene i model-laget implementerer "Serializable". Dette fører til at filene er svært lite lesbare. Filen **register.dta** er der all informasjon blir skrevet til. Denne må derfor ikke slettes.