



Avdeling for informatikk og e-læring, Høgskolen i Sør-Trøndelag

Øving 2: Java-interface og RMI

Else Lervik

31.01.2008

Lærestoffet er utviklet for faget LO346D Java EE og distribuerte systemer

Dette er vanskelig stoff, og gjennomgangen i leksjonen er relativt overfladisk. En mer grundig gjennomgang finner du i læreboka.

Jeg anbefaler at du leser boka og kjører eksemplene. Merk at du uansett utgave av boka bør bruke eksemplene som hører til den nyeste utgaven, se <http://www2.tisip.no/boker/java/>.

Ang. unntakshåndtering: Du skal ikke gjøre noe spesielt ut av dette i disse oppgavene. Håndter de unntakene som måtte oppstå ved å sende dem videre.

Oppgave 1 - Programmeringsoppgave

I leksjonens eksempelsamling finner du en fil som heter Register.java. Den inneholder klassene Utstyr og Register. Studer kildekoden slik at du skjønner sammenhengen mellom de to klassene. Disse klassene er ikke tilrettelagt for RMI.

Du skal nå lage et distribuert system basert på RMI, der et objekt av klassen Register kjører på tjenersiden. Objektet skal lages på tjenersiden. Klienter skal kunne endre datainnholdet i objektet ved å sende meldinger til det.

Du må gjøre nødvendige tilpasninger i klassene Register og Utstyr. Tenk over hva du behøver å gjøre: Serialisering, eget interface og subklasse til UnicastRemoteObject og/eller synkronisering av metoder. Vær bevisst på de valgene du gjør!

Lag et tjenerprogram der du lager Register-objektet og registrerer dette i navnetjenesten. Tjenerprogrammet skal videre legge inn fem utstyr objekter i dette Register-objektet.

Lag et klientprogram som lar brukeren endre lagerbeholdningen. La programmet gå i løkke, slik at endringer kan gjøres flere ganger. For hver gang skal fullstendig lagerbeholdning for alle varene (utstyr objektene) skrives ut på skjermen hos klienten. (Tenk ikke for vanskelig – du skal ikke lage et menystyrt klientprogram.)

Til slutt skal bestillingslisten skrives ut.

Prøv å kjøre flere klientprogram mot det samme tjenerobjektet. La en klient endre varebeholdningen, og en annen sjekke dette.

Oppgave 2 - Teorispørsmål

Teorispørsmålene er i tilknytning til oppgave 1.

a) Sett opp de filene som en klient kan greie seg med (så få som mulig).

Innledning til oppgave b) og c)

Disse oppgavene handler om argumentoverføring. Gå først gjennom programliste 19.4 i "Programmering i Java" og kontroller at du skjønner hvorfor utskriften blir som den blir. Se også oppgavene i slutten av kap. 19.3.

b) Du har forhåpentligvis funnet at klassen Register må være subklasse til UnicastRemoteObject. Hva må du gjøre med klassen Utstyr? Begrunn svaret.

c) Anta at følgende tre metoder skal legges inn i klassen Register:

```
int finnAntallRegistrert() throws RemoteException;
Utstyr finnUtstyr(int indeks) throws RemoteException;
void regNyttUtstyr(Utstyr u) throws RemoteException;
```

Hvilke konsekvenser, om noen, får dette for klassen Utstyr? Gå gjennom de aktuelle alternativene (ingen endring i Utstyr, serialisering, subklasse til UnicastRemoteObject, synkronisering) og gjør rede for hvorfor, eventuelt hvorfor ikke, de kan brukes.

d) Hvorfor er det viktig at metodene i klassen Register er synkronisert? Klassen ArrayList som brukes her, er imidlertid ikke synkronisert. Bruk API-dokumentasjonen til å finne et alternativ (slå opp på ArrayList). Er det nødvendig å bruke et synkronisert alternativ til ArrayList? Begrunn svaret.
