UiT

NORGES
ARKTISKE
UNIVERSITET

Hin Bank og Fallittkasse

2017

Narvik Høgskole
ITE1810 Internettapplikasjoner, Høst -17
Allan Arnesen, aar029

Innholdsfortegnelse

Innledning

Jeg har møtt mine problemer underveis, og mine redesigninger. Selve oppstarten av dette prosjektet gikk egentlig relativt greit, og var ganske lett å forstå i forhold til de tidligere oppgavene. Spesielt grunnet gode eksempler som var lagt ut på Canvas.

Jeg startet med å bygge Hin Bank med kun en sessionBean BankServices, med da tilsvarende interface BankServicesRemote.

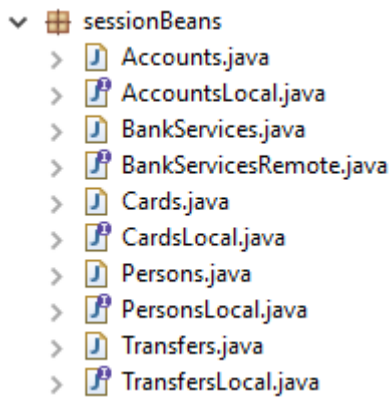
Gjennom denne kunne klientene få tilgang til å opprette nye entiteter hos banken, eller hente ut entiteter som skulle behandles (kort, konto).

Jeg startet så ombyggingen til at jeg beholdt sessionBean BankServices og BankServicesRemote, men at klientene fikk kun tilgang på metoder i BankServices som har egne metodekall til lokale sessionbeans som styrer tilgang til entitetene. Det gikk en god del tid bort i å i praksis lage prosjektet 2 ganger, og det ble kanskje litt vel lite tid til ferdigstilling av det endelige prosjektet. Men jeg føler uansett at det var verdt det da det var veldig mye læring å hente.

Kan nevne at jeg har i realiteten bygd prosjektet 2.5 ganger da det av en eller annen grunn plutselig ikke gikk å åpne workspace til prosjektet lengre. Etter litt undersøkelse fant jeg gjennom loggene til Eclipse at et save-kall(saveAll()) hadde feilet i løpet av natten grunnet ingen «top windows». Det var rundt om kl 03.00 loggingen var gjort. Et tidspunkt pc'en var avslått og jeg sov. Jeg kom i hvert fall frem til at det fort kunne være enklere å starte på nytt enn å skulle finne ut av feilen. Bla var noen GLOBALS ramlet ut osv. Ikke verdens største katastrofe da ingen data eller kode var gått bort. Men litt smårart uansett. Samt at jeg måtte manuelt skrive en del kode på nytt. Ved copy/paste av enkelte filer så fikk jeg feilmeldinger om at klasser jeg aldri har hatt i prosjektet ikke ble funnet.

Struktur

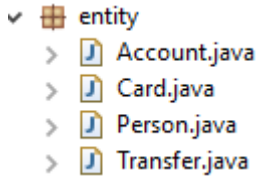
Strukturen vår er som vist på bildene. Jeg har bygd det slik at hver entitet har sin egen sessionBean. Men bare lokal. Så ingen klienter skal ha direkte tilgang til disse.



Selve banken Hin Bank og Fallittkasse har fått sine tjenester som er tilgjengelig for klienter, da her som minibanker, gjennom sessionBean BankServices og BankServicesRemote.

BankServicesRemote gir klientene tilgang til ønsket funksjonalitet, mens BankServices i tillegg benytter seg av alle lokale sessionBeans for å få utført spørringer, oppdateringer, endringer, registreringer eller slettinger mot databasen. Hvor da de forskjellige sessionBeans'ene representere sin entity klasse.

På så måte er klienten her frakoblet POJO's og man kan endre utføring, teknologi osv for de forskjellige bank-tjenestene uten at det skal påvirke klienten.



Entitetene er bygget på den veldig enkle måten at du har en Person. En Person kan ha mange Accounts. Hver Account har et Card. Og hver endring av saldo på Account får en oppføring i Transfer. Slik at overføring mellom 2 kontoer vil gi 2 linjer i Transfer. En for hver konto. Et uttak på den ene, og ett innskudd på den andre.

Transfers logging

I eksemplet på neste side er en overføring mellom 2 kontoer. Der vi først sjekker saldoen til avgivende konto. Om avgivende konto har dekning, trekker vi penger fra den kontoen, setter det inn på mottakende konto og logger begge overføringene.

Samme prosess benyttes i alle transaksjoner som vedrører saldoen til en konto.

```
try{
    Context context = new InitialContext();
    AccountsLocal accountSessionBean = (AccountsLocal)context.lookup("java:comp/env/Accounts");

    Account fromAccount = accountSessionBean.getAccount(fraKonto);
    Account toAccount = accountSessionBean.getAccount(tilKonto);
    Date date = new Date();
    double balance = fromAccount.getSaldo();

    if(balance >= belop){

        fromAccount.setSaldo(balance-belop);
        toAccount.setSaldo(toAccount.getSaldo() + belop);

        fromAccount = accountSessionBean.getAccount(fraKonto);
        toAccount = accountSessionBean.getAccount(tilKonto);

        Transfer transfer = new Transfer("Uttak", fromAccount.getKontonummer(), belop,
                "Overføring til " + toAccount.getKontonummer(), fromAccount.getSaldo(), date);
        addTransfer(transfer);
        transfer = new Transfer("Innskudd", toAccount.getKontonummer(), belop,
                "Overføring fra " + fromAccount.getKontonummer(), toAccount.getSaldo(), date);
        addTransfer(transfer);
```

Koden generelt

Jeg går ikke i noe særlig stor grad gjennom koden her, da den er rimelig godt dokumenter i selve koden.

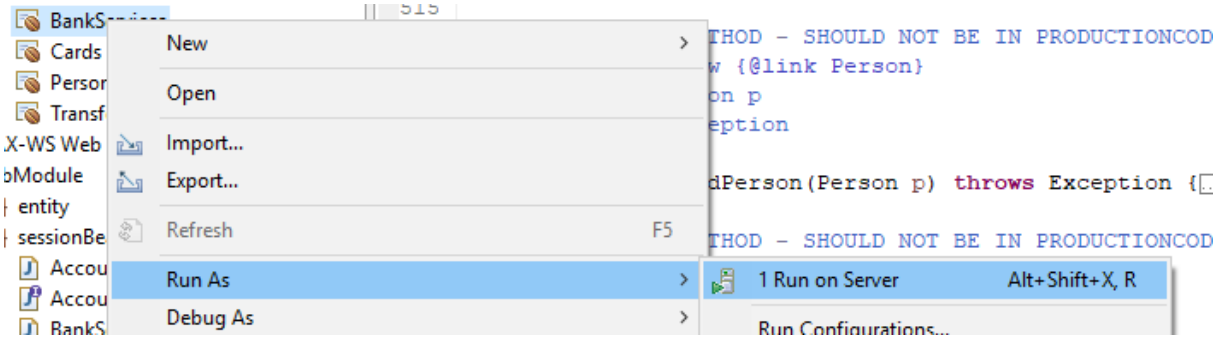
Det som kan nevnes er at det er et par like metoder som benyttes litt ulikt og noe kode som kun er for test mened. Disse er også dokumentert i koden. Men det er noen metoder som gir tilgang til å få ut hele lister med kort, kontoer og personer. Disse ville jeg selvfølgelig ikke latt ligge om dette var produksjonskode. Det er også et par forskjellige metoder for å opprette kontoer ol. Det er på grunn av at enkelte metoder er brukt til å generere test-data, som for eksempel en saldo på kontoen og PIN til kortet. Men det er en annen metode som er designet og implementert for klientene.

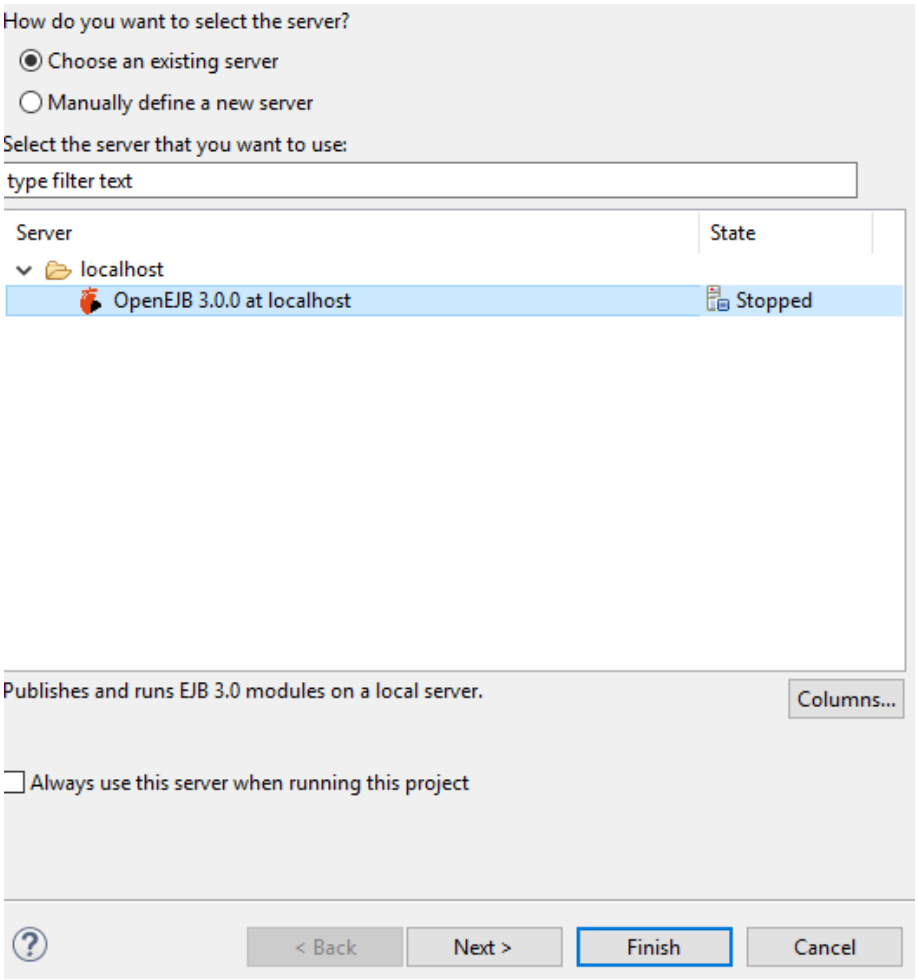
Alle metoder som er tilgjengelig gjennom TestEJB's minibank test benytter kun metoder som er tiltenkt klienter. De andre metodene brukes i selve TestEJB.

I de metodene som har vært laget 2 ganger siden jeg brukte 2 forskjellige design er også med i innleveringen, i sine representative metoder. Bare kommentert ut. Årsaken til det er at jeg var litt usikker til å begynne med om det var nødvendig å strukturere det slik som jeg har gjort, eller om det holder å benytte entityManager direkte fra BankServices. Jeg er rimelig sikker på at jeg har skjønt det riktig, men om det er slik at det hadde holdt med det opprinnelige oppsettet tar jeg gjerne en tilbakemelding på det.

Kjøre prosjektet

For å kjøre prosjektet er det ikke verre så å starte sessionBean BankServices med Run on Server.





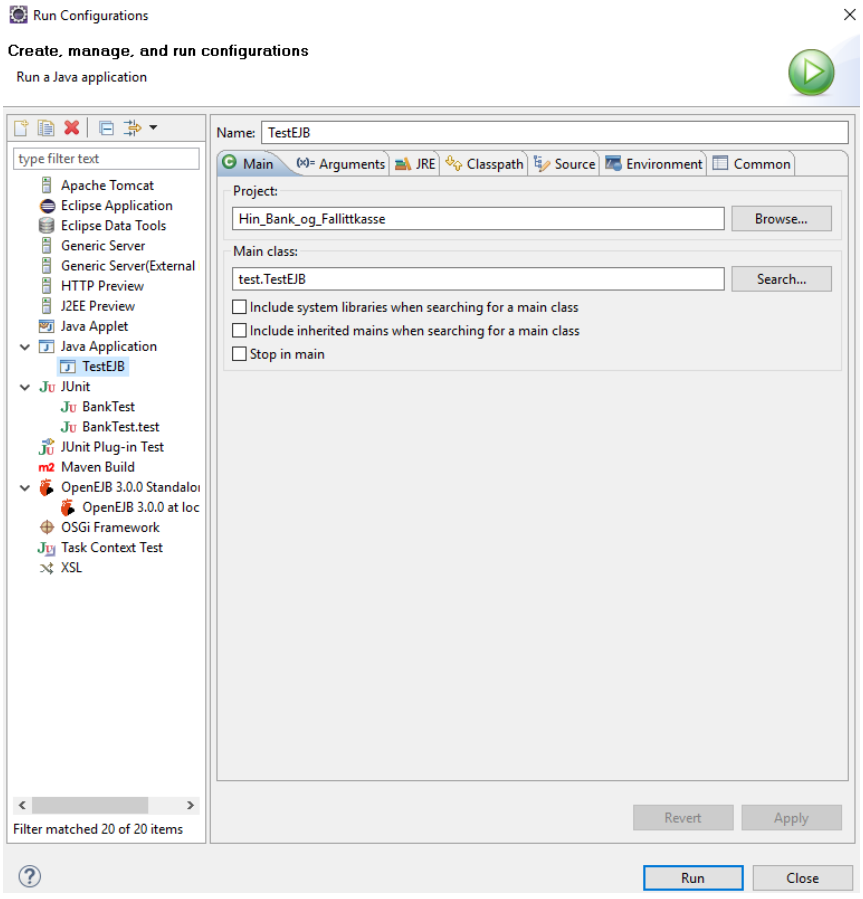
Trykk finish. Eventuelt opprett deg en server på samme måte som fremgangsmåte gitt på Canvas.

Resultat:

```
OpenEJB 3.0.0 at localhost [OpenEJB 3.0.0 Standalone server] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (25. nov. 2017, 2
INFO:    NAME                 IP               PORT
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    httpejbd             127.0.0.1        4204
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    ejbd                 127.0.0.1        4201
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    admin                127.0.0.1        4200
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    multipulse           239.255.2.3      6142
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    ejbds                127.0.0.1        4203
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    multicast            239.255.2.3      6142
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager printRow
INFO:    multipoint           127.0.0.1        4212
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager start
INFO: -------
nov 25, 2017 11:50:09 PM org.apache.openejb.server.SimpleServiceManager start
INFO: Ready!
```

Start så klienten ved å høyre klikke på TestEJB. Velg Run As og Run Configurations. IKKE Run on Server.

Velg Java Application. Fyll ut som vist og trykk Run.



Du får nå opp testprogrammet i Console.

```
Adding testpersons..
Adding testkonto's and card's..


**********************
Velkommen til Bank Test
**********************
Options
1. List Person's
2. List Konto's
3. List Kort's
4. List Transfer's
5. Test Minibank
6. Exit
Skriv inn ditt valg:
```

Her kan du velge å få listet opp aller Personer, Kontoer, Kort eller Transfers som er registrert totalt. Kun brukt for testing og oversikt.

PRO-TIP. Kjør nummer 3 om du skal teste minibank slik at du kan få tak i et par kortnummer og pin koder.

Om du kjører nummer 5 får du opp minibanken og kan på en veldig enkel måte teste de forskjellige tjenestene minibanken tilbyr.

```
Skriv inn kortnummeret:
510000005
Skriv inn PIN:
3208


**********************************************************
Velkommen til minibank Test for kort 510000005
**********************************************************
Options
1. Sjekk saldo
2. Opprett konto
3. Se Transaksjoner
4. Innskudd
5. Uttak
6. Overfør mellom konti
7. Exit
Skriv inn ditt valg:
```

Får å kjøre JUnit testene (de få som er der) høyre klikker du på filen BankTest og velger Run as -> JUnit Test. Da kjøres testene.