Om uppgiften

Så nu bör ni ha läst igenom uppgiften - annars gör det!! - och kanske är redo att sätta igång. Men var skall man börja?

Ofta är det så att det är svårt att veta var man skall börja och vi (lärare) kommer att lägga tid på detta kommande tisdag under föreläsningen. Men ju mer förberedda ni är desto mer får vi ut av föreläsningen och därmed av uppgiften och därmed höjer vi er kunskap och tentaresultatet blir förhppningsviss bättre. Lång mening den där sista!

Två huvudsakliga uppgifter har vi framför oss:

- Sätt upp en utvecklinsmiljö
- Lös uppgifterna (implementera kraven)

Vi kommer lösa dessa två huvudproblem i separata kapitel. Dessa problem kan, och kommer, delas upp i mindre, och därmed mer enkelt lösbara, delar.

. . . så till saken, hur börja?

Få igång ett exempel

Vi kunde ju utgå från exemplena. Alltså gör vi så.

Katalogstruktur

Men vi vill ju ha en egen paktestruktur (jo, det vill ni!) så det är inte bara att hacka rakt in i exempelkoden.

Låt oss skapa lite kataloger

1. 3pp - innehåller de ramverk som vi använder oss av

   ```
   mkdir 3pp
   ```

2. edu/sysvp/program - katalog(er) för vårt program, klubbhanteraren.

   ```
   mkdir -p edu/sysvp/program
   ```

Notera att det är tre under varandra liggande kataloger i (2)

Släng in grejern i 3pp-katalogen

1. Kopiera menuconsole-jarfilen till 3pp, samt döp om den:
   cp ../tmp-dist/sandklef-edu-2015-01-29-131042.jar 3pp/sandklef-edu.jar
2. Kopiera tig058-jarfilen till 3pp, samt döp om den
   cp ../tmp-dist/tig058-2015-01-30-114744.jar 3pp/tig058.jar
3. 3pp-katalogen ser nu ut enligt följande:

   ```
   3pp
   |---- sandklef-edu.jar
   |---- tig058.jar
   ```

4. Kopiera ett av exemplena till er program-katalog. Vi tar här exemplet utan ConsoleMenu. Ni rekommenderas dock att utgå ifrån ConsoleMenu-exemplet.

   a. Packa upp zip-filen med exempelkod, t ex så här:

   ```
   unzip ../tmp-dist/program-example-2015-02-02-102724.zip
   ```

   b. Kopiera exempel-programmet.

   ```
   cp myprogram/hmi/cli/*.java edu/sysvp/program/
   ```

   Obs, paketstrukturen i Java-filerna i edu/sysvp/program/ måste nu fixas till så att de hänger ihop med vår katalogstruktur! Se föreläsning om paketstruktur

   c. edu-katalogen ser nu ut enligt följande:

   ```
   edu/
   |----sysvp/

      |--- program/

           |--- ClubHelper.java

           |--- ClubMain.class
   ```

OBS: Rikard (aka besserwisser) har rätteligen påpekat att en klass som heter xxHelper egentligen brukar impliciera att den har ett annat syfte vår klass har. Så det blir plus i kanten (även Immanuels kant) om ni döper om klassen till något annat

Kompilera och kör

1. Kompilera Java-filerna

   ```
   javac javac -cp .:3pp/sandklef-edu.jar:3pp/tig058.jar edu/sysvp/program/*.java
   ```

   *Sh%#t, vad många fel! varför... jo vi måste säga till javac att det finns klasser på andra ställen än där den normalt letar. Sätt CLASSPATH vid kompilering, t ex

   ```
   -cp 3pp/ett-ramverk.jar:3pp/ett-annat-ramverk.jar
   ```

   OBS, det är upp till er att få till rätt CLASSPATH

2. Kör programmet

   ```
   java -cp .:3pp/sandklef-edu.jar:3pp/tig058.jar edu.sysvp.program.ClubMain
   ```

   What?

   ```
   Exception in thread main java.lang.NoClassDefFoundError: tig058/handin01/log/Logger
   ```

   But why? ... jo vi måste säga till Java att det finns klasser på andra ställen. Sätt classpath vid körning/exekvering, t ex:

   ```
   -cp 3pp/ett-ramverk.jar:3pp/ett-annat-ramverk.jar
   ```

   OBS, det är upp till er att få till rätt CLASSPATH

Om att börja lösa ett problem

I uppgiften står det:

- "Er uppgift är att skriva ett menysystem"
- "Programmet skall vara terminalbaserat"
- "Enklast är nog att utgå ifrån de medföljande exempelprogrammen."

Dela in/upp problemet i mindre delar

Ok, vi skall göra ett menysystem som i sin tur skall köra annan kod utifrån användarens önskemål. Detta är ju egentligen två problem:

1. menysystem

2. det som skall göras när användaren väljer ett menyalternativ

Uppdelat i två delar har vi två mindre problem att lösa. Ok ok ok, sedan skall vi sammanföra lösningarna, men ändå det är ju nu två mindre problem .. eller kanske tre om man räknar med sammanföringen.

Utöver att man kan fokusera på en sak i taget, så är det dessutom bra att dela upp gränssnitt och "logik" för att få en mer flexibel och lättunderhållen programvara.

Menysystemet - eller *Varför bry sig?*

1. Om man gör menysystemet att till en början endast skriva ut något i stil med "du valde menyalternativ 1" istället för att faktiskt implementera det som skall hända så kan vi i lugn och ro utveckla och testa menysystemet.

I kort, *"varför bry sig om funktionaliteten när vi kan strunta i den just nu?"*

Funktionaliteten

1. Skapa nu en klass som innehåller metoder som implementerar en lösning för t ex menyalternativ 1.

Klassen kan t ex heta Controller eller KlubbHanterare och Metoden skulle kunna heta listMemberAlpha.

Fråga? Hur skall ni i instanser av denna klass få tillgång till ert ClubRegistry-objekt som är basen för er kommunikation med det underliggande ramverket.

3. INNAN, ja innan!, ni kopllar ihop menyalternativ 1 med metoden i den nyss skapade klassen kan ni testa den separat.

- Skriv en liten klass, t ex ListTester (eller kanske ListEsther?).
- Klassen kan innehålla en main-metod som skapar en instans av klassen (t ex Controller eller KlubbHanterare) ni nyss skapade.
- Anropa metoden, t ex listMemberAlpha, och kolla att resultatet är ok.

Slå ihop dellösningarna - *One ring to bind them all*

Ok, nu har vi ett menysystem och en implementation (en metod i en klass). Låt oss nu sammanföra dessa två.

1. Skapa ett objekt av den nya klassen på lämpligt ställe i er kod.
2. Istället för den utskrift ni gjorde i menysystemet ("du valde menyalternativ 1") kan ni nu istället anropa metoden i er nya klass

Gå vidare

Vi har nu gått igenom hur man sätter upp utvecklingsmiljön i *Få igång ett exempel* och hur man kan dela upp problemet i lite mindre delar *Om att börja lösa ett problem.*

Återstår ***bara*** att implementera alla krav. . . .

Mvh Rikard och Henrik

. . . . . **Don't panic** (Douglas Adams)