

Klubbhanteraren

Detta är inlämningsuppgift #01 för kursen TIG058.

Vi har uppdaterat årets inlämningsuppgift för att bättre passa kursens mål samt för att ge er mer relevant kunskap.

Senast uppdaterad 2015-01-30

Beskrivning

Ni skall skriva ett program till ett befintligt ramverk som hanterar en klubb. Ramverket sparar information om föreningens medlemmar - ni behöver alltså inte skriva kod för att spara till eller läsa från fil.

Er uppgift är att skriva ett menysystem vilket lämpligen använder sig av, och förmodligen bygger ut, den medföljande hjälpklassen (ClubHelper).

Programmet skall vara terminalbaserat. Det innebär att ni skall skriva och läsa text på skärmen.

Enklast är nog att utgå ifrån de medföljande exempelprogrammen.

Syfte

Syftet med inlämningsuppgiften är

- att ni skall lära er:
 - använda externa klasser (m h a import) i form av jar-filer
 - läsa dokumentation
 - skriva dokumentation
 - hantera obekanta datamängder (ni får en “databas”)
 - läsförståelse (i samband med att ni använder externa ramverk)
 - skriva testfall i kod
 - reflektera över design och implikationer av de system ni använder
 - reflektera över och testa hur menysystem lämpligen utvecklas
 - reflektera över kravens fullständighet
 - använda manualer och instruktioner för installation och användande av externa ramverk
- att förbereda er:
 - för att jobba mot en databas

- för att jobba med ett GUI-ramverk (t ex Swing)

Vi har för avsikt att arbeta vidare med denna ide i inlämningsuppgift #03 där vi byter ut nuvarande lagringssätt och det kommandoradsbaserade menysystemet mot ett fönsterbaserat. Tanken är att vi då kan diskutera de val av design vi gjort i ramverket i dag. Vår förhoppning är att detta skall bidra positivt till er förståelse för de mer övergripande programmeringsdelarna så som Interface och Arv snarare än bara ren kod.

Instruktioner

ClubRegistry

Använd endast klassen ClubRegistry när du skall kommunicera med ramverket. Övriga klasser i ramverket används av ClubRegistry. Detta är ett exempel på en så kallat lagrad arkitektur.

Kika i de medföljande exemplena efter hur man använder sig av ClubRegistry.

Tips

- Utgå ifrån det medföljande programmet ClubMain och dess hjälp-klass ClubHelper. Ni kan utöka funktionaliteten direkt i filen eller låta er inspireras (nja?) av programmet och skriva ett eget.

Examination av inlämningsuppgiften

- Ni skall lösa uppgiften enligt specifikation (se krav).
- Vi kommer eventuellt att testa koden med testkod som vi publicerar innan det så kallade systemtestet.
- Ni kommer att få svara på frågor rörande er lösning samt hur ni använt ramverken.

Hur förhåller sig uppgifterna mot tentan?

Funktionella krav på G-nivå motsvarar frågor på en tentamen på G-nivå.

Funktionella krav på VG-nivå motsvarar frågor på en tentamen på VG-nivå.

Krav

Definitioner

- List - med lista avses här verbet som skall tolkas som att skriva ut på skärmen

Funktionella krav (motsvarande G-nivå)

1. Programmet skall ha en välkomsttext där (bl a) följande skall skrivas ut
 - a. användarnamn (redan implementerat)
 - b. datum
2. Programmet skall ha ett menysystem där användaren interaktivt kan göra följande:
 - a. lista alla medlemmar i alfabetisk ordning på efternamn. Detta skall vara menyalternativ 1.
 - b. lista alla medlemmar i alfabetisk ordning på förnamn, Detta skall vara menyalternativ 2.
 - c. lista alla medlemmar i ordning efter id. Detta skall vara menyalternativ 3.
 - d. lista alla medlemmar i ordning efter lag-tillhörighet. Detta skall vara menyalternativ 4.
 - e. lista alla medlemmar i ett lag - laget anges interaktivt av användaren. Detta skall vara menyalternativ 5.
 - f. lista föräldrar i ett lag - laget anges interaktivt av användaren. Detta skall vara menyalternativ 6.
 - g. lista epost-adress(er) för en medlem. Om medlemmen är 18 år inestående år skall medlemmens epost-adress användas, annars skall föräldrars epost-adress användas). *Obs, detta är en nedbantad variant av att kunna skicka epost* Detta skall vara menyalternativ 7.
 - h. Er meny skall *ALLTID* skriva ut alternativen samt avvakta inptut från användaren genom att skriva > på skärmen. (*Detta för att Rikard och Henrik vill skriva automatiska tester*).

T ex kan en meny se ut så här:

1. lista all members (ordered by family name)
osv osv osv osv osv
 6. List parents:
 7. List email addresses
- >

3. Programmet skall också kunna startas upp i icke-interaktivt läge där följande s k command line options skall finnas:
 - a. `--members` - skall skriva ut samtliga medlemmar
 - b. `--teams` - skall skriva ut samtliga lag

Efter programmet har utfört den av användaren givna instruktionen skall programmet termineras (avslutas).

Krav på er kod (motsvarande G-nivå)

1. Menyhanteringen skall skötas i en separat klass. Detta innebär att klassen som innehåller main-metoden blir ganska lite, vilket är bra.
2. Er kod skall vara kommenterad där så behövs
3. Det skall finnas en manual (använd Javadoc) till era klasser. Observera att det inte räcker med att kompilera era klasser med javadoc - ni måste också ha relevanta Javadoc-kommentarer.

Funktionella krav (motsvarande VG-nivå)

1. Ert menysystem skall implementeras med hjälp av externt menysystem, ConsoleMenu (se nedan)

OBS: vi tror att detta är enklare än att skriva ett eget system, men tröskel till att använda det är högre. Vi tror även att detta förbereder er bra inför GUI-delen av kursen

2. Gör om koden, inklusive ramverket, så att Team inte representeras av en String utan är en separat klass (t ex Team) Team not as a string

Databas med Member-objekt

Du får med en fil (members.db) som innehåller en bunte med Member-objekt. Lägg dessa i den katalog där du kör programmen - den ligger redan på rätt plats i zip-filen.

Du kan också skapa en egen databas. Se nedan om Testkod.

Externa bibliotek

tig058

Detta är ramverket ni skall använda för hantering av medlemmar. Det finns en del begränsningar, en del på grund av tidsbrist och en del ditlagda med vilje.

Ramverket är dokumenerat med hjälp av Javadoc. Genererad manual (API) finns bifogad i inlämningsuppgiftens zip-fil där du fick denna fil. Filen är döpt efter dagens datum, för att hålla reda på versionerna, så följande exempel är bara ett exempel:

```
'tig058-2015-01-30-001842.jar'
```

*OBS: Det går att döpa om eller kopiera jar-filen så att namnet inte innehåller datumet. Låt oss anta att vi döpt om filen till **tig058.jar***

För att kompilera klasser som använder detta ramverk gör något i stil med detta:

```
javac -cp .:tig058-2015-01-30-001842.jar DinJavaKlass.java
```

Nu instrueras javakompilatorn (**javac**) att leta efter klasser även i jar-filen.

Källkoden till ramverket följer med så det är bara att kika - och gärna rätta - på det. Källkoden finns också i zip-filen. Ex:

```
extra/source-2015-01-30-001842.zip
```

ConsoleMenu

Ramverket medföljer som jar-fil. Dokumentaion (Javadoc) med exempel medföljer.

Vi har skrivit ett alternativt, och (våldigt litet) exempelprogram (ClubCM) som använder sig av detta ramverk. Detta program finns i (nästan) samma katalog som ClubMain.

```
sandklef-edu-2015-01-29-131042.jar
```

*OBS: Det går att döpa om eller kopiera jar-filen så att namnet inte innehåller datumet. Låt oss anta att vi döpt om filen till **sandklef-edu.jar***

För att använda ramverket med din kod får du instruera java att leta efter klasser i jar-filen. Gör något i stil med följande:

```
javac -cp tig058.jar:3pp/sandklef-edu.jar DinJavaKlass.java
```

```
java -cp tig058.jar:3pp/sandklef-edu.jar myprogram.hmi.console.CMClub
```

I exemplet ovan startas medföljande CMClub-program och java instrueras att leta efter klasser i jar-filen.

Utgå gärna ifrån exempelprogrammet när du utvecklar ditt program.

Exempelprogram

Till uppgiften finns ett par exempelprogram. Dessa finns i zip-filen enligt följande:

```
program-example-2015-01-30-001842.zip
```

Notera att datumet återigen kan variera beroende på vilken version som laddas upp, så se namnet ovan som ett exempel

För att köra dessa program, gör följande:

- Packa upp zip-filen (program-example-2015-01-30-001842.zip)
- Kompilera alla filer:

```
javac -cp sandklef-edu.jar:tig058.jar myprogram/hmi/cli/*.java
javac -cp .:sandklef-edu.jar:tig058.jar myprogram/hmi/console/*.java
```
- Kör programmen så här:

```
java -cp .:sandklef-edu.jar:tig058.jar myprogram.hmi.cli.ClubMain
java -cp .:sandklef-edu.jar:tig058.jar myprogram.hmi.console.CMClub
```

Testkod

Med i zip-filen följer lite test-kod som på ett ytterst enkelt sätt testar delar av ramverket tig058.

Obs, vi har i nedanstående exempel tagit bort datumet från zip-filen

För att kompilera dessa filer:

```
javac -cp .:tig058.jar test/*.java
```

För att köra dem:

```
java -cp .:test:tig058.jar TestClubStore
```

```
java -cp .:test:tig058.jar TestLogger
```

För att skapa en ny medlemsdatabas:

```
java -cp .:test:tig058.jar GenerateMembers
```

Manual

Med i zip-filen manualer/API:er för ramverket tig058. Dessa finns i zip-filen som heter:

```
tig058-manual.zip
```

Zip-filens innehåll

- description.pdf - denna pdf
- extra - källkoden till tig058-ramverket
- program-example-2015-01-30-114430.zip - exempelprogram
- sandklef-edu-2015-01-29-131042.jar - consolemenu-ramverket
- sandklef-edu-javadoc-2015-01-29-131042.zip - manualer till consolemenu
- test-2015-01-30-114430.zip - testkod
- tig058-2015-01-30-114430.jar - tig058-ramverket
- tig058-manual.zip - manual för tig058-ramverket