

Inlämningsuppgift 3

Programmeringsteknik och databaser TIG058 VT16

1. Syftet med laborationen

Att tillägna sig kunskaper om konstruktion av ett program som använder sig av

- JDBC för att sköta all lagring av data
- Java Swing för att presentera ett enkelt grafiskt gränssnitt

2. Förutsättningar och krav

Denna inlämningsuppgift eller laboration bygger till en del på den föregående java-laborationen #1, med en del nya och/eller förändrade förutsättningar.

2.1. Databas

Databasen består av följande relationer:

medlem

Attribut	Datatyp	Ex på värden
id	int	345
givenName	varchar(50)	Oskar
familyName	varchar(50)	Oskarsson
email	varchar(50)	xyz@myclub.com
gender	varchar(6)	kvinna
birth	char(10)	1978-05-20
memberSince	char(10)	2010-05-23
active	int	1

Kommentar: Information om medlemmar. Kön (gender) kan vara kvinna eller man. active kan ha värdena 1 (true) eller 0 (false).

funktion

Attribut	Datatyp	Ex på värden
id	int	567
role	int	1
team	varchar(10)	P00

Kommentarer: Vad man gör i klubben och för vilket lag. 0 är spelare, 1 är coach, 2 är förälder.

Children

Attribut	Datatyp	Ex på värden
id (parent)	int	234
id (child)	int	99

Kommentar: Föräldrar och deras barn. Kan också användas för att se vilka föräldrar ett barn har.

2.2. Programförutsättningar

Programmet skall ha ett grafiskt gränssnitt baserat på Java Swing och all hantering av filer som gjordes under den tidigare laborationen skall nu istället skötas av en databas som vi kommunicerar med via JDBC.

Indata till databasen ligger i en textfil som innehåller alla sql-satser som behövs för att skapa databasens tabeller och fylla dem med innehåll. Denna finns i Gul under kursens sida, filen ligger i katalogen Inlämningsuppgift #3 och med namnet `init_databas.sql`. Denna fil kan sedan läsas in till den databashanterare ni använder. Detta kan göras varje gång man vill ha ett fräscht innehåll i databasen. T ex, kommandot `.read init_databas.sql` i `sqlite3` exekverar alla sql-satserna som finns i `init_databas.sql`.

Programmet skall kunna utföra två huvuduppgifter, Registervård resp. Sökning.

Vid ändringar av något i databasen skall denna uppdateras ögonblickligen.

Programmet skall vid uppstart presentera någon form av meny där man kan välja vad man vill göra. Det skall också vara möjligt att navigera fram och tillbaka i programmet, t ex. efter det att en sökning efter ledare genomförts skall man kunna ta sig tillbaka en nivå i programmet.

2.2.1. Registervård

Via detta alternativ skall man kunna lägga till *Nya Medlemmar* resp. *Uppdatera medlemsinfo*. `JOptionPane` får inte användas för att mata in information utan programdelen skall mer likna ifyllandet av ett formulär.

För *Nya Medlemmar* gäller följande (OBS! Det är inte enbart relationen `member` som behöver uppdateras.):

- Inga fält får vara tomma.
- Id skall genereras av programmet (eller databasen).
- För funktion skall det vara möjligt att *role* kan anta fler än ett värde för en given medlem, det skall t ex vara möjligt att både vara coach och förälder.
- Om man är förälder skall också aktuella barns id läggas in.
- I övrigt där det så går att göra skall felkontroller finnas.

För *Uppdatera medlemsinfo* gäller följande, använd id eller namn för att välja rätt medlem för att sedan kunna:

- Aktivera/inaktivera medlemmen
- Ändra mailadress
- Ändra medlemmens funktion
- Ta bort en medlem

2.2.2. Sökningar

Via detta alternativ skall det vara möjligt söka i medlemsregistret på följande sätt:

- Söka efter en specifik medlem, använd efternamn eller id som sökbegrepp. **All** information om denne medlem skall presenteras, dvs. föräldrar till barn skall också visas etc.
- Söka efter vem/vilka som är ledare för ett visst lag.
- En sorterad listning av alla medlemmar, det skall vara möjligt att välja sorteringsordning antingen på id eller efternamn.
- Information om ett specifikt lag med angivande av spelare, hur många de är och vem/vilka som är coach.

2.2.3. Icke funktionella krav

- Alla java filer skall vara välindenterade och ha beskrivande namn på alla variabler, klass och metoder. Vid namngivning följ standarden beskriven i: <http://www.oracle.com/technetwork/java/codeconv-138413.html>.
- Använd inte svenska tecken i filnamn och i namn på variabler, klasser etc.
- Hårdkoda **INTE** sökvägar till t ex databasen (som C:\Users\nisse\Mina Filer\Java\test.db), fråga istället efter var databasen finns eller anta att den finns i aktuell katalog.

3. Examination

Om laborationsdokumentationen inkl. program ej godkänns direkt kan retur ges (max 2) på sådan dokumentation som med moderata ändringar eller kompletteringar är möjliga att godkänna. I annat fall underkänns laborationsdokumentationen utan retur.

De som ej fått laborationsdokumentationen godkänd "senaste godkännande-datum" kommer att underkännas på laborationen. Senaste godkännandedatum är tisdagen den 13/5 2015.

Nästa tillfälle att få laborationen godkänd blir nästa gång kursen går och man måste då utföra vid det tillfället aktuell laboration.

Laborationen skall genomföras med en gruppstorlek om maximalt 2 personer.

Dokumentationen skall innehålla följande:

- En Readme-fil innehållande namn och personnummer på de i gruppen medverkande. Denna fil skall också innehålla alla eventuellt gjorda antaganden. Skriv gärna också något om hur ni upplevt denna laboration, vad som varit svårt, vad som var för lätt och vad vi borde ändra till nästa år.
- Alla java-filer skall medfölja. Inga class filer skall inkluderas.

Ingående filer skall sedan läggas i ett zip-arkiv och laddas upp till Gul (Inlämningsuppgift #3) senast måndagen den 28/3 2015 kl. 17.00, dvs. efter systemtestet som sker fredagen den 26 resp måndagen den 28/3 enligt bokningsschema som kommer att sättas upp utanför mitt rum. Laboration inlämnad för sent underkänns utan pardon.