

Projekt Delfinklub - Gruppe 10

Administrativt system til svømmeklubben Delfinen

Dette program er tilegnet 3 forskellige 'roller' i svømmeklubben, som angivet i use-case-diagrammet; formanden, der står for administration af selve medlemslisten; kassereren, der står for registreringen af betalinger fra medlemmer; og træneren, der står for al administration af kompetitive hold, inkl registrering af resultater fra træning og stævner. Hele vores use-case er blevet implementeret i programmet.

Funktionalitet

I starten af programmet ser man en skærm med 3 login muligheder. Man kan logge ind som formand, kasserer eller træner ved at klikke på rubrikken hvor dette står på. Der er også muligheden for at lukke programmet.

I alle vinduer i programmet har vi implementeret en **tilbage** knap der giver muligheden for at gå tilbage til det tidligere vindue i programmet.

Funktionen **Vis medlemmer** viser de forskellige medlemmer i klubben. Dette er muligt ligegyldigt hvem man logger ind som. Denne funktion viser alle medlemmer i klubben. Man kan sortere medlemmerne efter navn, fødselsår og hvorvidt de er aktive eller passive medlemmer.

Sorter		Navn ▼					
Name	Fødselsdato	Adresse	Postnr	By	Telefon	Mail	Medlemskab
AK Ak	25-03-2000	adds 12	1245	Byen	12345678	lk@gmail.com	Aktiv
Aa Aa	28-06-1990	lkajf 12	6345	Byen	23456789	AAAAAA@gmail...	Passiv
Bk Bk	02-02-2002	asdas 20	2002	Byen	12345678	bk@gmail.com	Aktiv
Ck Ck	11-03-2010	asd 5	3265	Byen	87654321	Ck@gmail.com	Aktiv
Dk Dk	11-11-2000	asd 12	5412	Byen	12345678	Dk@gmail.com	Aktiv
Kkk Kkk	13-11-1996	gdf 21	5498	Byen	12345678	kkk@gmail.com	Passiv
Pk Pk	02-04-2009	asdas 25	2356	Byen	98765432	Pk@gmail.com	Aktiv

Formanden: Som formand kan man tilføje, redigere eller slette medlemsoplysninger.

Under **Tilføj Medlemmer** kan man indtaste og vælge de forskellige parametre der kræves for at oprette et Member objekt og til sidst registrere det. Knappen "Clear" rydder tekstfelterne der er skrevet i, uden at personen registreres.

☐ Passiv
 ☒ Aktiv

☒ Motion
 ☐ Konkurrence

Navn:

Fødselsdato:

Adresse:

Postnr:

By:

Telefon:

Mail:

Hvis det lykkes at registrere en person, ryddes tekstfelterne, og man får en bekræftelse i det gule felt. Tekstfelterne kræver specifik information og hvis man indtaster noget forkert eller udelader noget information og forsøger at registrere medlemmet får man en fejlbesked.

Navn: Morten

Fødselsdato: 01-13-2001

Adresse: 444 4

Postnr:

Fejl: Skriv venligst det fulde navn.
Skriv venligst vej og nummer.
Skriv venligst et dansk postnummer.
Skriv venligst en gyldig by.
Skriv venligst et telefonnummer på 8 cifre.
Skriv venligst en gyldig email.

Under **Rediger Medlem** kan man vælge alle medlemmer ud fra en dropdown menu, som det ses her til højre:

Navn: Bk Bk

Alder: 02-02-2002

Adresse: asdas 20

Postnr: 2002

By: Byen

Telefon: 12345678

Mail: bk@gmail.com

☐ Passiv ☒ Aktiv

Tilbage

Slet Rediger

Abraham Sapien

Vælg

Så kan man redigere specifik information for dette medlem eller helt **slette** medlemmet. Her får man på samme måde fejlbeskeder, hvis noget er forkert, eller en bekræftelse, hvis det går godt. Dette ses her til venstre.

Kassereren: Som kasserer kan man enten tilføje betalinger eller se medlemmer, der er i restance.

Under **Tilføj Betaling** er der en dropdown menu med alle medlemmer som ikke har betalt kontingent.

Ved at klikke på "Betal alt" betales hele kontingentet for det valgte medlem og denne fjernes fra dropdown menuen. Man kan også selv indtaste beløbet der betales i vinduet til højre og trykke på "Betal". Så mindskes det beløb som medlemmet skal betale.

AK Ak

At Betale

1600.0

, - DKK

Betal alt Betal

400 , - DKK

Betal

At Betale

700.0

Betal alt

Betalingen er registreret.

Under **Vis Restance** får man en liste over personer der ikke har betalt kontingent, og man kan ligesom i “Vis Medlemmer” sortere dem efter navn, fødselsår og medlemskab.

Tilbage

Sorter:

Navn

Name	Fødsels-dato	Adresse	Postnr	By	Telefon	Mail	Kontingent	Restance
Aa Aa	28-06-1990	lkajf 12	6377	Byen	23456789	AAAAAA@gmail...	500.0	500.0
Abraham Sapien	02-02-1950	BPRDHQ 1	4321	Gettysburg	98765432	AbeSapien@out...	500.0	500.0
AK Ak	25-03-2000	adds 12	1245	Byen	12345678	lk@gmail.com	1600.0	700.0
Bk Bk	02-02-2002	asdas 20	2002	Byen	12345678	bk@gmail.com	1000.0	1000.0
Dk Dk	11-11-2000	asd 12	5412	Byen	12345678	Dk@gmail.com	1600.0	1600.0
Kkk Kkk	13-11-1996	gdf 21	5498	Byen	12345678	kkk@gmail.com	500.0	500.0
Pk Pk	02-04-2009	asdas 25	2356	Byen	98765432	Pk@gmail.com	1000.0	1000.0

Træneren: Som træner kan man få vist de to konkurrencehold eller få vist en top 5 af resultater i en bestemt disciplin. Man kan også registrere et medlem som konkurrencesvømmer, eller tilføje resultater fra hhv træningssessioner og stævner.

Registrer K åbner et vindue hvor man enten kan tilføje et medlem til et konkurrencehold, og dermed gøre medlemmet aktivt - eller fjerne et medlem, der allerede er konkurrencesvømmer, fra sit hold og dermed gøre medlemmet passivt. Dette ses her til højre.

Tilbage
ak ak

ak ak er nu aktiv på et konkurrencehold.

Fjern
Tilføj

I **Vis K-Hold** kan man via en dropdown liste se en oversigt over enten junior eller senior konkurrencesvømmere.

Tilbage

Hold

Senior

Name	Fødselsdato	Adresse	Postnr	By	Telefon	Mail
DK Dk	29-01-2000	hjk 1	5645	Byen	23456789	DK@gmail.com

Tilføj T-Resultat og **Tilføj K-Resultat** kan man tilføje tider i forskellige discipliner til en konkurrencesvømmer. Træningsresultatet bliver kun gemt hvis det er hurtigere end svømmerens personlige rekord i den respektive disciplin. Alle stævnerresultater bliver derimod gemt. Hvis man indtaster noget forkert eller udelader noget information, får man en fejlbesked. På samme måde får man en bekræftelse, hvis det lykkes at tilføje resultatet.

Egon Olsen

Disciplin:
BUTTERFLY

Tid:

Dato:

Tilføj

Tilbage
Simon Norup

Disciplin:
BUTTERFLY

Tid:

Stævne:

Placering:

Dato:

Tilføj

Vis Top 5 viser de 5 hurtigste konkurrencesvømmere i en given disciplin, inkl det tilhørende resultat:

Sorter	BUTTERFLY	
Navn	Hold	Tid
gsjd fhg ashdkas	Senior	(time=09:12:60, date=03-05-2006)
Simon Norup	Senior	(time=10:10:10, date=11-03-2005)
jkhakd asdkjl	Senior	(time=10:12:60, date=03-05-2007)
Egon Olsen	Junior	(time=12:10:10, date=11-03-2005)

Robusthed

Exception Handling:

Vores primære exception er `IllegalArgumentException`, som bl.a. udnyttes i konstruktørerne i `Member` og de to `Result`-klasser. Her tjekker de, om de givne informationer er gyldige. Hvis ikke, kastes en exception tilbage til GUI-klassen, som håndterer dem ved at printe en informativ besked på den relevante `JFrame`. Fx skal datoer formateres "DD-MM-ÅÅÅÅ", mens navne og emails, der allerede eksisterer i medlemslisten også afvises. Derudover kastes `RuntimeException` med en relevant besked, hver gang noget går galt i `FileHandler`-klassen, fx hvis en linje i tekstfilen ikke er formateret korrekt, eller hvis programmet ikke kan finde den angivne `Path`. Vi burde selvfølgelig have brugt mere specifikke typer exceptions, men dette nåede vi ikke at gennemføre.

Unit Tests:

Vi har nået at lave unit tests til alle klasser undtagen `CompetitiveResult`. I `MemberTest` testes både korrekte og ugyldige argumenter i konstruktørerne, idet en stor del af den interessante funktionalitet går ud på at kaste exceptions fx ved forkert formaterede Strings. I `MasterSystemTest` har vi brugt `@Fix - MethodOrder` for at fremtvinge den tiltænkte rækkefølge på testene, da resultaterne tager højde for, hvad der er sket tidligere.

Vedligeholdelsesvenlighed

Navngivning:

Vi har navngivet metoder og forskellige funktioner på engelsk, så engelsktalende programmører også kan læse koden. De funktioner der er navngivet har relevante navne som "getMembersInArrears", hvor `arrears` er det engelske ord for restance, eller "addToCompetitiveTeam" som gør præcis som dens navn beskriver. Det samme gælder klassenavne; "CompetitiveSwimmer" blev valgt netop for forståelsen, selvom navne som `Member2` eller `CompMember` ville have været nemmere at arbejde med i koden.

Javadoc:

Vi har skrevet kommentarer for de vigtigste klasser, inkl. `MasterSystem`, `FileHandler` og `Member`, og derefter autogenereret javadoc. Her har vi dokumenteret hvad både klasserne generelt og de individuelle metoder implementerer af funktionalitet, dvs. info om (1) hvad de gør overordnet set, (2) parametre, (3) returtype, (4) exceptions. Dermed gøres det lidt lettere at forstå hvad det hele går ud på, når man skal vedligeholde eller udvide systemet. Forfatter og version er også angivet øverst.

Interfaces:

`MasterSystem` og `FileHandler` har hver deres interface, som ikke har nogen genbrugelig funktion som sådan, men som har fungeret som kontrakter. Disse interfaces sørger blot for, at man aldrig bryder med kravspecifikationen (fx `MasterInterface` = use case).

Genbrug

Vi har genbrugt vores kode ved både nedrivning af felter og funktioner, og ved private metoder inden for klasserne hver især. Også ved at kalde én public metode i mange af samme klassers andre public metoder, som det ses med `readMembersFromFile` i `FileHandler`-klassen.

Nedrivning:

`CompetitiveSwimmer` arver fra `Member`, da denne blot er en udvidelse af `Member` med alle de samme stamoplysninger, men samtidig med nye informationer i form af resultater.

`CompetitiveResult` arver fra `TrainingResult`, da denne også er en udvidelse af det almindelige resultat blot med ekstra info om stævne og placering.

Interne metoder:

- `readFile` fra `FileHandler` læser fra filerne "competitives" og "members" som så bliver konverteret til arraylists i andre metoder.
- `tableSet (GUI)` fylder det givne skema med information fra "members" filen den har forskellige funktioner baseret på hvilken "TableEnum" (enum) man sender igennem. Det også den der holder sorterings koden hvor den tager information fra den combobox der hører til det table man er på. `compTable(GUI)` gør det samme som `tableSet` men tager fra `competitives`. har heller ikke sorteringsfunktion, informationen er dog opdelt efter junior og senior holdene.
- `isFaster` fra `MasterSystem` tjekker bare om en tid fra træning eller konkurrence er hurtigere end en anden. Brugt af `addTrainingResult` som også bliver brugt af `addCompetitiveResult`

Modulært design:

Det modulære design gør, at andre bedre kan genbruge de forskellige klasser. `FileHandler`-klassen står for al kommunikation og konvertering til og fra filer (eksterne data). Derfor ville man uden problemer kunne bytte den ud med en hvilken som helst anden `dataAccessor`. GUI'en har vi på samme måde forsøgt at få til at stå for al brugergrænseflade og intet andet, så den ville kunne byttes ud med nye UI-klasser. Dermed kan `MasterSystem` såvel som de basale klasser genbruges.

Eventuelle forbedringer

Når man sorterer medlemslisten efter alder gør den det kun efter årstallet de er født i uden at tage højde for måned og dato. Sorterings algoritmen kunne i det hele tage forbedres med flere sorteringsmuligheder.

En anden forbedring kunne være at hvert medlem havde deres egen side som man kunne se på i forhold til deres information, både i forhold til personlig information og svømmeresultater. Dette kunne eventuelt gøres ved at man kunne klikke på en bestemt person i tabellen.

GUI koden kunne også forbedres ved at hver `JFrame` blev delt op i klasser så hele koden bliver mere overskuelig. Desuden kunne brugergrænsefladen være flottere.