Dokumentasjon TDT4145

Halvor Ødegaard Teigen
Ole-Jørgen Hannestad
Simen Krantz Knudsen

March 20, 2019

1 Antagelser gjort i prosjektet

- Man kan gjennomføre maks en treningsøkt per dato. Dette er aktuelt for use case 2: Hvis man la inn to økter i løpet av samme dato, kunne man risikere å få frem informasjon om kun en av dem.
- En øvelse må være i en øvelsesgruppe.
- Man kan ikke legge inn samme øvelse to ganger i løpet av en økt. Dette vil i så fall føre til duplikate primærnøkler.
- Attributtene `Prestasjon` og `Form` i klassen Treningsøkt gjelder for en hel økt sett under ett. En enkeltøvelse inneholder ikke denne informasjonen.
- Attributten `Treningspartner` for en treningsøkt kan være Null eller inneholde en personID. Antagelsen er dermed at man kan ha maks én partner per treningsøkt.

2 Klasser

Klassene i dette prosjektet består av følgende:

- Apparat
- DBConn
- DBOperations
- Fastmontert
- Frittstaende
- Ovelse
- Ovelsesgruppe
- Person
- TreningsCtrl
- Treningsokt

3 Beskrivelse av klassene

* Apparat: Klassens oppgave er å holde på informasjon om et apparats respektive apparatID, navn og beskrivelse.
* DBConn: Klassens oppgave er å opprette en connection med databasen.
* DBOperations: Klassens oppgave er å ta hånd om spørringer til databasen og behandlingen av disse. Dette inkluderer:
  - Å legge til apparater, øvelser, øvelsesgrupper og økter.
  - Hente ut informasjon om n siste økter med notater.
  - Vise resultatlogg i et intervall spesifisert av sluttbruker.
  - Finne alle øvelser i samme øvelsesgruppe.
  - Hente ut alle personer (med navn, tlf.nr. og favorittøvelse) som har en gitt favorittøvelse.
  - Logikk for å gå gjennom tabellene i databasen ved søk etter allerede eksisterende rader, f.eks. en spesifikk økt eller øvelse.

  Det antas i klassen at informasjonen fra konsoll har blitt håndtert riktig av klassen TreningsCtrl.
* Fastmontert: Arver fra klassen Ovelse. Klassens oppgave er å holde på informasjon om en fastmontert øvelses totale antall kg, antall gjennomførte sett og tilhørende apparat som brukes.
* Frittstaende: Arver fra klassen Ovelse. Klassens oppgave er å holde på informasjon om en frittstående øvelses beskrivelse.
* Ovelse: Superklassen til Fastmontert og Frittstaende. Klassens oppgave er å holde på informasjon om en øvelses ID og navn.
* Ovelsesgruppe: Klassens oppgave er å holde på ID og navn en øvelsesgruppe.
* Person: Klassens oppgave er å holde på informasjon om en persons ID, navn, telefonnummer og favorittøvelse. For en treningsøkt skal man kunne legge til en treningspartner, av typen Person.
* TreningsCtrl: Klassens oppgave er å håndtere input fra konsoll på riktig måte, og formatere informasjonen på riktig måte før DBOperations interagerer med selve databasen.
* Treningsokt: Klassens oppgave er å holde på informasjon om en treningsøkt. Denne informasjonen er ID, dato, form, prestasjon og starttidspunkt for den respektive økten, samt øktens varighet, notat og eventuell treningspartner.

4 Use cases – Oversikt og realisering

1. Registrere apparater, øvelser og treningsøkter med tilhørende data:

   (a) Registrering av apparater: Funksjonen `addApparat()` i klassen TreningsCtrl spør om navn og beskrivelse på et apparat. Dette blir lagt direkte inn i databasen.

   (b) Registrering av øvelser: Funksjonen `addOvelse()` i klassen TreningsCtrl spør i konsollen om navn, form og prestasjon for en øvelse, samt om den respektive øvelsen er fastmonterte eller frittstående. Hvis fastmontert, itereres det over allerede eksisterende apparatIDer i databasen for å matche oppgitt apparatID.

   (c) Registrering av treningsøkter: Funksjonen `addTreningsokt()` spør om dato, tidspunkt,varighet, evt ID på treningspartner samt notatet for økten. Dette blir lagt direkte inn i databasen av funksjonen `addTreningsokt` i klassen DBOperations, som har to implementasjoner: en med og en uten treningspartner.

2. Få informasjon om n siste økter med notater, n spesifisert av bruker:

   Funksjonen `printNSisteTreningsokter()` i klassen TreningsCtrl henter ut en liste med de n siste treningsøktene og printer informasjonen. All informasjon tilknyttet objektet printes ut. Funksjonen er i tillegg definert slik at bruker får informasjon om hvor mange økter det totalt finnes i databasen, dvs. maks n.

   I klassen DBOperations er funksjonen `getNSisteTreningsokter()` implementert, som legger til de n siste treningsøktene i databasen i en liste. Funksjonen `getTreningsokter()` legger til alle treningsøkter i en liste.

3. For hver øvelse skal det være mulig å se en resultatlogg i et gitt tidsintervall, spesifisert av bruker:

   Funksjonen `printLoggForOvelseITidsrom()` i klassen TreningsCtrl printer en oversikt over tilgjengelige øvelser i databasen og spør bruker om å oppgi ID på ønsket øvelse. Iterer gjennom alle øvelser og tilhørende treningsøkter, og printer ID på økt, dato, starttidspunkt og evt. treningspartner. Øvelser og treningsøkter er koblet sammen via tabellen "Ovelseriokt" i databasen.

4. Lage øvelsesgrupper og finne øvelser som er i samme gruppe:

   Funksjonene `addOvelsesgruppe()` og `printOvelserIOvelsesgruppe()` i klassen TreningsCtrl hhv. legger inn en oppgitt øvelsesgruppe i databasen og printer øvelsesID og -navn på alle øvelser i en øvelsesgruppe.

5. Eget use case - printe alle personer med en gitt favorittøvelse:

   Funksjonen `printPersonerMedGittFavorittovelse()` i klassen TreningsCtrl ber bruker spesifisere en favorittøvelse hun vil se. Itererer gjennom alle øvelser for å finne riktig ID, og itererer deretter over en liste med alle personer som har denne øvelsen som favorittøvelse. Printer ID, navn og telefonnummer på alle personene.