

Treningsdagbok dokumentasjon

Adrian Flatner, Joakim Hantho Qvale, Halvor Horge

Klasser:

- **ActiveDomainObject**
Abstrakt klasse med load og save.
- **Apparat**
Håndterer apparatId og lager beskrivelse, samt lagrer disse.
- **Bruker**
Håndterer brukerId og brukernavn, samt save av disse. Begge må være unike.
- **DBConn**
Abstrakt klasse som oppretter databasetilknytting.
- **Ovelse**
Håndterer øvelsesnavn og sett, samt save av disse.
- **OvelseGruppe**
Lager gruppeld og beskrivelse for øvelsesgruppe, samt save av disse.
- **Treningsdagbok**
Main klasse som kjører hele programmet til konsoll.
- **TreningsdagbokCtrl**
Inneholder spørringene til krav 2, 3, 4 og 5, i tillegg til alle funksjoner som lagrer innhold i databasen. I denne ligger mesteparten av programmets funksjonalitet, og det er gjennom denne alle use cases blir løst. Videre blir også relasjoner laget i databasen i det ulike versjoner av data blir lagret.
- **Treningsokt**
Lager øktId, dato, tidspunkt og varighet, samt lagrer disse.

Usecases:

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

Beskrivelse:

Hver bruker av applikasjonen har en unik ID. Hver bruker kan registrere ubegrenset antall treningsøkter lenket til brukeren med en fremmednøkkel. Hver økt kan om ønskelig tillegges et notat. Hver økt kan også registrere en eller flere øvelser som alle har en primærnøkkel i form av ØvelseID. Øvelser er også registrert i øvelsesgrupper som inneholder grupperingen av en eller flere øvelser. Videre spesifisering av øvelser skjer ved hjelp av to disjunkte subklasser: Apparatøvelse og Frivekt. En apparatøvelse kan utføres på ett eller flere apparater og er linket ved hjelp av ApparatID. Apparater tilbyr utførelse av en eller flere apparatøvelser.

Gjennomføring:

Alle klasser og tilhørende informasjon er laget i Java og MySQL. Main-programmet (Treningsdagbok), er lagd for å implementere dette i konsollen. Dette blir gjort gjennom å dele opp programmet i 3 hvor man enten kan registrere treningsøkt, apparat eller øvelse. Hver gang brukeren skriver inn vil et kall til kontrolleren sørge for at objektene blir lagret i databasen, i tillegg til å lage nødvendige relasjoner.

2. Få opp informasjon om et antall n sist gjennomførte treningsøkter med notater, der n spesifiseres av brukeren.

Beskrivelse:

Hver økt har en ØktID som også kan fungere som en teller. På den måten kan man hente ut de n siste øktene. Tilhørende hver økt finnes et notat som kan hentes ut samtidig

Gjennomføring:

Spørringer i TreningsdagbokCtrl er utviklet og testet til å fungere. Main-funksjonen realiserer spørringen gjennom case 2. Her skriver brukeren

inn hvor mange økter den vil se. Deretter vil “brukerokt” fra kontrolleren bli kalt, som gjør en spørring for å vise use casen.

3. For hver enkelt øvelse skal det være mulig å se en resultatlogg i et gitt tidsintervall spesifisert av brukeren.

Beskrivelse:

Joine notat og treningsøkter for så å sette en begrensning på treningsøktentiteten som spesifiserer dato- og tidspunktattributtene.

Gjennomføring:

Use casen blir gjennomført i case 3 av programmet. Brukeren skriver inn hvilken øvelse den vil ha, i tillegg til et tidsintervall. Deretter blir resultatlogg i kontrolleren kalt.

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

Beskrivelse:

Den blir oppfylt med relasjonen mellom Øvelse og Øvelsesgruppe der man kan linke øvelse til forskjellige grupper.

Gjennomføring:

Programmet spør om opprettelse av ny muskelgruppe. Her kan man legge til en muskelgruppe med tilhørende øvelser. I kontrolleren blir nye muskelgrupper opprettet, og man kan også legge til øvelser i disse. Her blir det opprettet en relasjon for å koble disse. Programmet spør så om hvilken muskelgruppe man ønsker å se øvelser fra. Her kan man velge den nylig lagde gruppen.

5. For hver treningsøkt skal det være mulig å se hvilke øvelser man kan gjøre på et apparat.

Beskrivelse:

Det finnes relasjon mellom treningsøkt og øvelse, og mellom øvelse og apparat.

Gjennomføring:

Denne casen blir gjennomført i case 5 av programmet. Her skal brukeren skrive inn navnet på et apparat. `ovelsePaaApparat` blir kalt i kontrolleren. Denne utfører en spørring i databasen som returnerer øvelsene som kan gjøres på apparatet.