

Innlevering 1 – PGR200

Frist: Søndag 2. oktober

Bakgrunn

Ved Westerdals Oslo ACT bruker vi betydelige ressurser på å sette opp en timeplan for semesteret. Arbeidet består av mye manuelt arbeid. Kan vi klare å automatisere noe av dette ved hjelp av et java-program?

Administrasjonen har informasjon om hvilke brikker som eksisterer i puslespillet som skal legges. Info om:

- Rom
- Forelesere (noen har visse krav/ønsker til undervisingstidspunkt)
- Emner (med mulige krav til fasiliteter, samt antall påmeldte, hvem skal undervise?)
- Start- og sluttidspunkt for semesteret.

Oppgaven

(Oppgaven vil bli ytterligere presisert når vi har fått jobbet med den. Det blir satt av tid til å jobbe med den i flere øvinger.)

Lag et program som kan:

- lese inn bakgrunnsinformasjon om minst tre emner ved Westerdals fra en fil. Du bestemmer selv formatet på informasjonen på filen.
- populære tabellen «Emne» med data fra innlesingen. Tabellen må du ha laget selv.
- hente ut informasjon om emner som ligger i Emnetabellen. Du skal kunne hente ut informasjon om alle emner, eller finne ett emne basert på emnekode.

Mulig utvidelser:

- Lese inn mer bakgrunnsinformasjon (rom, forelesere etc) som populeres i flere tabeller. Tabellene bør ha riktige koblinger seg imellom.
- Lese inn metadata om alle tabeller slik at tabellene kan opprettes dynamisk under kjøring.
- Sørge for en god validering av inn-dataene.
- Lage forslag til optimal timeplan. Dette forutsetter at regler for ugunstige situasjoner eksisterer og behandles. Timeplanen beskrives på egnet format.

Krav til innleveringen

1. Maven-prosjekt.
2. Java 8.
3. Eventuelle referanser til hvilke kilder som er brukt.
4. Et kortfattet notat der du beskriver:
 - a. Eventuelle forutsetninger for å kunne teste løsningen.
 - b. Kommentarer til eget resultat. Kunne noe vært gjort bedre?
 - c. Eventuelle spørsmål du trenger svar på for å komme videre.
5. Valg: **Enten** a eller b:
 - a. En video-capture av presentasjon av løsningen (koden og testene); ca 5-10 minutters varighet. Videoen bør inneholde en kjøring av testene der alle tester blir grønne.

Deretter forandres kode for å vise at minst en test feiler. Vis også dekningsgraden. Screencast-o-matic anbefales som verktøy for video-opptaket, men andre verktøy kan benyttes. En lenke til videoen skal leveres – ikke videoen selv. Husk å åpne for tilgang til videoen («unlisted» i Youtube).

- b. Hvis video ikke leveres skal man levere et mer utfyllende notat der man gjør rede for løsningen sin, gjerne med skjermdump av testresultat, testdekning etc.

Vurdering

Formålet med innleveringen er å gi studenten mulighet til å vise sin måloppnåelse, særlig relatert til Maven, JDBC og testing. Andre konsepter som kodekvalitet og unntakshåndtering vil også naturlig spille inn når kode skal vurderes. PGR200 vurderes 100% basert på mappevurdering. Innlevering 1 vil være en del av denne mappen. Studenten vil få en vurdering av innleveringen og vil deretter få mulighet til å utbedre løsningen inntil endelig levering av mappen i slutten av semesteret.

Vurderingskriteriene kommer i ITL.

Relevans ift emnebeskrivelsen – innlevering 1

Fra emnebeskrivelsen - PGR200.

Læringsutbytte

Relevante kunnskaper

Etter å ha fullført emnet skal studenten

- **forklare begrepet "SQL injection" og hvordan man kan unngå dette**
- **vite hva JDBC (Java Database Connectivity) er (...) og hvordan vi kan benytte disse for å kommunisere med en database**

Ferdigheter

Etter å ha fullført emnet skal studenten kunne

- **teste deler av applikasjonen ved hjelp av enhetstester og integrasjonstester**
- **bruke byggverktøy til å spesifisere avhengigheter, kjøre automatiserte tester og bygge et Javaprojekt**
- **utvikle programmer som kommuniserer med en database og utfører CRUD (create, read, update, delete) operasjoner**

Generell kompetanse

Etter å ha fullført emnet skal studenten kunne

- **begrunne og presentere sine egne løsninger for en mottaker som har bakgrunn innenfor fagområdet programmering**