

Info233- Avansert Programmering, våren 2019

Obligatorisk oppgave: 3 av 4

Innleveringsfrist: Mandag 8. april kl. 14.00
Format: .zip arkivfil
Filnavn: brukernavn_oblig3.zip
Pakkenavn: no.uib.info233.oblig3

Om obligatoriske oppgaver

I de obligatoriske oppgavene stilles det en rekke krav. Disse må tilfredsstilles, ellers godkjennes ikke oppgaven. Det er veldig viktig at du leser oppgaven nøye.

Hvis teksten er uklar eller du mener det finnes flere tolkninger av et krav, vennligst la undervisningsassistentene få vite dette og etterlys en presisering.

Om man får «ikke godkjent», får du anledning til å levere en egenretting kort tid etter.

Du vil uansett få tilbakemelding på innleveringene dine.

Når du skal levere må hele prosjektmappen arkiveres i en .zip mappe. Oppgavene leveres på Mitt UiB i de relevante mappene.

Det er ikke nødvendig å gjennomføre deloppgavene i den samme rekkefølge de er oppført. Les gjennom alle oppgavene før du begynner. Legg ved det du blir bedt om å forklare, gjøre rede for, eller analysere i en egen fil.

For å bestå oppgaven så må du fullføre minst 80 av 100 poeng og du må ha minst 4 poeng på delspørsmål h.

For å oppnå rett til egenretting, må du oppnå minst 40 poeng.

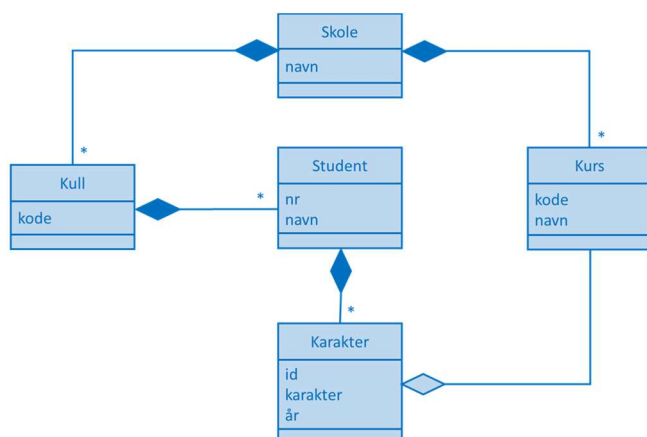
Krav

I denne tredje obligatoriske oppgaven skal du jobbe med grafiske brukergrensesnitt og databaser fra en Java-applikasjon. Fremdeles gjelder noen generelle regler om innleveringene: Du skal

- Utføre alle deloppgavene som oppgitt i oppgavebeskrivelsen under.
- Lage et enkelt kjørbart program i Java.
- Organisere klasser i pakker.
- Skrive kommentarer til alle klasser og metoder i henhold til JavaDoc (se vedlegg A i læreboken).
- Ha meningsfulle navn på metoder og variabler.

Oppgave

I oppgaven kommer dere til å jobbe med Java, Design Patterns, JavaFX og SQLite. Oppgaven er å implementere et studentregisteringssystem. Systemet skal kunne registrere følgende på en skole for hvert kull; studenter, kurs, og karakter på kurs en student har tatt. I denne oppgaven kan du bruke alle deler av klassebiblioteket som følger med Java.



UML-modell for problemstillingen og danner grunnlag for databaseskjema. **SQL for å opprette databaseskjema for dette skjemaet vil bli gitt snarlig**, men hvis du ønsker, kan du lage det selv på grunnlag av modellen.

Legg ved det du blir bedt om å forklare, gjøre rede for, eller analysere i en egen fil. Les gjennom alle oppgavene før du begynner. Legg spesielt merke til deloppgave h som må løses til minst 4 poeng. Det er ikke nødvendig å gjennomføre deloppgavene i den samme rekkefølge de er oppført.

Deloppgaver

Hver deloppgave beskriver hva som skal gjøres og hvor mange poeng som kan oppnås. En delvis implementasjon gir ikke full pott, men kan gi en redusert poengmengde hvis implementasjonen er i riktig retning.

Deloppgave a) – 15 poeng

Lag en main-metode som bruker gitt modell (skjema) til å opprette en ny database med SQLite.

Hvis du ønsker å benytte et annet databasesystem enn SQLite, kan du gjøre det, men SQLite er kanskje det enkleste som finnes. Spør på lab-en om du lurer på noen tekniske ting. Informasjon er å finne her: <https://www.sqlite.org/index.html>

Det er ikke nødvendig å bruke den gitte modellen eller skjemaet, du kan også implementere ditt eget, eller endre det om du skulle ønske det. Legg da ved det du har gjort. Skjemaet du lager må inneholde minst den informasjonen som er gitt.

Deloppgave b) – 5 poeng

Utvid main fra deloppgave a til å befolke databasen (legg inn data). **Det kommer et forslag som er beskrevet ved SQL-setninger, men du kan befolke den med dine data hvis du ønsker det.**

Deloppgave c) – 10 poeng

Lag en objekt-modell i Java for å implementere modellen som er gitt. Dette betyr at du skal oversette UML-klassene til Java-klasser.

Deloppgave d) – 5 poeng

Lag en ny main-metode (du kan ha mange main-metoder i et Java-prosjekt, men de må eksistere i ulike klasser og da kan du velge hvilken som skal utføres). Denne skal opprette en ny skole (UiB), et kull med kode 2019V, et kurs med kode INFO233 og navnet "Avansert programmering", studentene Per og Kari som har nr 1 og 2 i kullet 2019V. Per har karakteren A i kurset INFO233 i 2019, Kari har karakteren A i kurset INFO233 i 2019.

Deloppgave e) – 15+5 poeng

Lag et vindu som kan vise karakterene til en student. Den skal vise studentens navn, og alle kurs (kode + navn) med karakter og årstall som studenten har registrert. Det er et pluss hvis vinduet også inneholder skole og kull for studenten. Det gis inntil 5 bonuspoeng for et godt utformet vindu! Hva som er godt er en subjektiv vurdering fra den som gir poeng.

Oppgaven er to-delt. Du kan lage en av to versjoner, den ene viser data tilsvarende de som er produsert i deloppgave d (dummy-data), den andre skal hente data lagret i databasen og vise dem. Husk å vurder utformingsmønstre. Det gis maks 15 poeng for en versjon som henter fra database og maks 10 for en som benytter «dummy»-data.

Deloppgave f) – 20+5 poeng

Lag et vindu hvor man kan registrere en ny student eller endre data på en eksisterende student på et kull på en skole. De registrerte dataene skal lagres i databasen. Hvis du ikke får det til, lagre dataene i en «dummy»-modell, men da får du et trekk på minst 5 poeng.

Deloppgave g) – 20+5 poeng

Lag et vindu som viser alle studentene på et kull på en skole i en liste. Når du beveger deg i listen skal du i et annet vindu se detaljer om studenten som til enhver tid er valgt. Detaljene kan være kun navn, men hvis du viser mer, er det bonuspoeng å hente på inntil 5 poeng for et godt resultat.

I utgangspunktet skal dataene hentes fra databasen, men du kan benytte «dummy»-data hvis du ikke har database. Dette gir et trekk på 10 poeng.

Deloppgave h) – 10 poeng

Reflekter over utformingsmønstrene (design patterns) du har brukt i oppgaven. Hvilken utformingsmønster har du brukt? Hvordan har du brukt dem? Hvorfor har du brukt dem? Hvordan er de nyttig? Det er viktig at du svarer på alle disse spørsmålene. Og hvis du ikke har brukt noen er det kanskje på tide å gjøre det. Uansett må du svare på følgende spørsmål: Er det utformingsmønstre du kunne ha brukt som kunne gjort utviklingen enklere?