

# Semesteroppgave for Programutvikling, V20

## Innlevering

Besvarelsen skal inneholde et Java kodeprosjekt som kan kjøres fra en Java IDE.

Anbefalt oppsett er et modul-basert Maven Java 13 prosjekt, opprettet og utviklet i IntelliJ. Eventuelle eksterne biblioteker må legges til i Maven konfigurasjonsfilen, inkludert JavaFX 13.

Sensor vil også kunne kjøre Java 8 IntelliJ prosjekter uten Maven. Med dette oppsettet er det anbefalt å ikke bruke eksterne biblioteker utover bibliotek for testing. (Hvis det inkluderes eksterne biblioteker er det en risiko for at sensor ikke får kjørt prosjektet.)

Hvis dere har et annet oppsett, så inkluder en tekstfil med instruksjoner for hvordan å kjøre prosjektet. (Dette anbefales ikke, men er tillatt.)

Hvis prosjektet inneholder syntaks- eller kompileringsfeil (dvs. ugyldig/uferdig Javakode), så kan ikke sensor endre besvarelsen med programmering for å korrigere koden. Hvis sensor ikke får kjørt et prosjekt, kan ikke besvarelsen evalueres og dere får ikke bestått. Gjør derfor litt ekstra testing for å verifisere at prosjektet kan kjøres fra en virkårlig datamaskin med et standard oppsett med IDE (helst, IntelliJ men sensor har også Eclipse og NetBeans installert) og JDK (Java 13 eller 8).

## Tekniske krav og krav til funksjonalitet

Dere skal implementere et konfigurasjonssystem med både brukergrensesnitt for sluttbrukere og superbrukere. Vedlegget på slutten av dette dokumentet beskriver tre konkrete alternativer dere skal velge mellom. Dere skal bare implementere et av disse tre alternativene.

Teknisk sett er de tre alternativene like utfordrende. Dere har derfor samme utgangspunkt for å oppnå en god karakter uansett hvilket alternativ dere velger.

Programmet dere skal implementere skal inneholde funksjonalitetene beskrevet under. Dere blir utelukkende evaluert basert på disse funksjonalitetene. Det vil si, dere kan gjerne legge til andre funksjonaliteter til programmet, men slik ekstra funksjonalitet blir ikke evaluert.

## Opprettelse av komponenter

Superbruker (administrator) skal ha muligheten til å opprette individuelle komponenter fra et grafisk brukergrensesnitt. Hvis brukeren taster inn ugyldig data, skal brukeren få beskjed om dette. Det skal dermed ikke være mulig å legge til elementer med ugyldig data.

Det grafiske brukergrensesnittet for å legge til elementer skal være designet slik at det er enkelt for brukeren å forstå hvordan å legge til elementer. Det blir lagt noe vekt på det grafiske designet, men ikke mye. Det vil si, det skal være enkelt å bruke, men det forventes ikke noe ekstraordinært som animasjoner eller andre dynamiske GUI funksjonaliteter.

## Opplisting av eksisterende elementer

Superbruker skal kunne få listet opp elementer som allerede er lagt til. Disse elementene må kunne endres på og slettes. Programmet skal støtte filtrering og sortering av elementene.

## Konfigurering av brukers produkt

GUIet for sluttbrukeren skal støtte muligheten for å opprette et produkt der brukeren selv konfigurerer de individuelle komponentene til produktet.

Brukeren skal kunne få listet opp tidligere fullførte konfigurerte produkter med relevant informasjon (som pris).

## Filbehandling

Programmet skal støtte filbehandling slik at programmets data ikke går tapt når programmet avsluttes. Det anbefales at besvarelsen allerede inneholder eksempeldata slik at sensor har noe å gå ut ifra.

Det forventes at bruker ikke må bruke mye tid på valg av plassering av filene. Filplasseringen kan for eksempel være pre-definert eller programmet kan la bruker få velge hvilken mappe filene skal lagres i. Til sammenlikning, har noen besvarelser fra tidligere år krevd at bruker velger filplasseringen til 10+ ulike konfigurasjonsfiler med FileChooser hver gang applikasjonen starter (dette er ikke spesielt brukervennlig).

Lagring av data som kommer fra superbruker skal lagres med binære filer, mens data fra sluttbruker skal lagres med tekstfiler. Tekstfilene skal være compatible med Excel, slik at produkter kan prosesseres med andre programmer.

Sensor kommer til å teste programmet ved å injisere ugyldig data i filene, så husk å inkludere håndtering av ugyldig data fra fil.

Metoden som laster inn data på superbrukerens side skal gjennomføres i en egen tråd. Legg til noen sekunder med venting slik at tråd-løsningen kan verifiseres av sensor. Det skal ikke være mulig å legge til eller endre på elementer i brukergrensesnittet mens tråden arbeider.

## Andre elementer

Funksjonaliteten over beskriver konkrete forventninger til programmet som sensor vil evaluere besvarelsen ut ifra. Det er opp til dere å utvikle annen funksjonalitet for at programmet skal fungere som forventet. For eksempel, må konseptet bruker og superbruker inkluderes på en eller annen måte. Her kan man legge til funksjonalitet for å opprette nye brukere eller opprette standard brukere, med eventuelle brukernavn og passord som på en eller annen måte informeres til sensor i besvarelsen.

## Evaluerings

Evalueringsen av besvarelsen blir gjennomført etter følgende kriterier for hver karakter:

- A. Besvarelsen løser alle problemstillinger i oppgaven. Programmet og besvarelsen generelt er utviklet slik at sensor enkelt finner frem til de ulike delene av oppgaven, både i brukergrensesnittet og i kode. Programkoden skal være utviklet etter standard prinsipper for objekt-orientert programmering.
- B. Samme som A, der mindre feil tillates. Det vil si, besvarelsen løser alle problemstillinger i oppgaven, men med noe rom for mindre feil og mangler.
- C. De viktigste oppgavene er korrekt løst. Dette inkluderer forventet funksjonalitet på sidene for sluttbruker og superbruker, samt støtte for filbehandling.
- D. De viktigste oppgavene er korrekt løst, der større mangler i deler av løsningen tillates. Et eksempel på en variasjon som tilsvarer denne karakteren er at all funksjonalitet er korrekt implementert, men løsning for lesing fra fil mangler.
- E. Kandidatene har implementert et fungerende registreringsprogram, men ikke stort mer enn dette. Basisfunksjonalitet som registrering av nye elementer og opplisting av registrerte elementer må være løst.
- F. (Ikke bestått.) Besvarelsen tolkes som ikke bestått etter minimumskravene beskrevet for karakter E.

## Code of conduct

Kopiering av kode fra Internett eller fra ferdigstilte løsninger, samt kopiering av kode mellom grupper, tilsvarer juks. Dette vil ikke bare føre til strykkarakter, men vil også føre til utvisning fra alle norske universiteter og høyskoler.

Gruppearbeid kan være krevende, spesielt for et større prosjekt som denne semesteroppgaven. Vi tolererer imidlertid ikke uakseptabel oppførsel. Dette inkluderer mobbing eller trakassering basert på kjønn, etnisitet, hudfarge, uførhet, seksualitet, religion eller tro, eller alder.

Mobbing er en form for psykologisk trakassering. Mobbing kan være trusler som undergraver selvtillit, kompetanse, og integriteten til offeret. Psykologisk trakassering kan også inkludere kontinuerlig ufortjent kritikk, nedsettende bemerkninger, høylytt kjefting, banning og støtende ordbruk, og påtrengende oppførsel. Om du er selv et offer for trakassering eller observerer trakassering, må du rapportere dette til emneansvarlig.

## Vedlegg

Dette vedlegget beskriver tre alternativer for hvilken type konfigurasjonssystem som kan utvikles i semesteroppgaven. Dere skal bare velge et av disse alternativene.

Alternativene representerer realistiske scenarioer for konfigurasjonssystemer og hvert alternativt kan potensielt representere svært komplekse systemer. Det er for hvert alternativ gitt forslag på konkrete konsepter som kan inkluderes i programmet. Dere må ikke følge disse forslagene slavisk, men forslagene representerer den type kompleksitet som forventes av en god besvarelse. For å få en god karakter er det dermed ikke forventet at dere utvikler et program som inneholder mer kompleksitet enn det forslagene representerer.

### Alternativ 1: Datamaskin

Dette konfigurasjonssystemet skal sette opp komponenter for en datamaskin. Man kan tenke seg at dette systemet kan brukes i salg av nye datamaskiner. Dette er relativt vanlig for kjøp av nye PCer. Se blant annet komplett.no og dell.no, som støtter tilpasning av individuelle komponenter.

Eksempler på komponenter:

- Prosessor
- Skjermkort
- Minne
- Harddisk
- Tastatur
- Mus
- Skjerm

Hver konkrete komponent skal ha en pris og sluttbrukeren skal få listet opp totalprisen på datamaskinen som er konfigurert.

### Alternativ 2: Bil

Dette konfigurasjonssystemet skal sette opp komponenter for en bil. Dette systemet skal støtte pris for hver komponent/tilpasning og kalkulering av totalpris.

Eksempler på biltype:

- Elektrisk
- Bensin
- Hybrid

Eksempler på komponenter:

- Motor (antall hestekrefter etc.)
- Felg
- Setetrekk
- Ratt (sport, vanlig etc.)
- Spoiler

Andre tilpasninger:

- Farge
- Integrert GPS navigasjon
- Type klimaanlegg
- Soltak
- Hengerfeste

### Alternativ 3: Bondegård

Dette er en kombinasjon mellom et konfigurasjonssystem og spill. En bondegård bygges opp ut ifra et gitt begrenset landområde. Innenfor dette området kan det opprettes områder for dyrking av mat og planter som korn, høy, frukt, og grønnsaker, og områder for beitemark og fjøs.

Superbruker kan konfigurere de ulike parameterne for bondegårder, som for eksempel hvor mye høy ei ku spiser per dag og hvor mye høy som kan høstes per kvadratmeter land.

Sluttbruker kan opprette en konkret bondegård der området for bondegården fylles opp med dyrkeområder og dyr. Programmet skal vurdere konfigurasjonen til bondegården, for eksempel etter parametere som fortjeneste, balanse mellom fôr og dyr, dyrevelferd, og klimautslipp.

Forslag til konsepter som kan inkluderes:

- Dyrking
  - Korn
  - Høy
  - Frukt
  - Grønnsaker
- Dyr
  - Kyr
  - Geiter
  - Sauer
  - Fjærkre
  - Griser
- Andre sentrale konsepter:
  - Traktor
  - Tresker
  - Fjøs
  - Silo
  - Beitemark