

JavaFX-prosjekt: Dashbord for sensorovervåking

Gruppestørrelse: 1–2 studenter | **Estimert arbeidsmengde:** 10–12 timer

Introduksjon

Smarte hjem er avhengige av sensorer – enheter som hele tiden måler temperatur, fuktighet, bevegelse eller luftkvalitet og rapporterer avlesningene sine i sanntid.

I dette prosjektet vil du bygge et **Sensor Monitor Dashboard** i JavaFX: et enkelt kontrollpanel som viser live sensordataoppdateringer, visuelle indikatorer og fargekodede statuslinjer.

Du vil øve på å bruke observerbare egenskaper, egenskapsbinding og hendelseshåndtering for å simulere et kontinuerlig kjørende overvåkingssystem.

Prosjektets mål

Utvikle et JavaFX-dashbord som:

- Viser en liste eller tabell over simulerte sensorer og deres avlesninger.
- Oppdaterer sensorverdier som svar på brukerinteraksjon (f.eks. ved å klikke på «Oppdater» eller «Start»).
- Indikerer sensorverdier visuelt (fremdriftslinjer, farger osv.).
- Lar brukeren starte eller stoppe overvåkingssimuleringen.

Krav

1. Generell struktur

- Prosjektet må være en JavaFX GUI-applikasjon (ingen konsollbruk).
- Følg en tydelig modulær struktur inkludert:
 - `MainApp` – inngangspunkt for applikasjoner.
 - Kontrollerklasse(r) – koordiner oppdateringer og håndter brukerhendelser
 - Layoutklasse(r) – bygg dashbordgrensesnittet direkte i Java ved hjelp av layoutbeholdere (f.eks. `BorderPane`, `VBox`, `HBox`, `GridPane`)

2. Simulering av sensor

- Lag minst tre sensortyper (f.eks. temperatur, fuktighet, lys).
- Hver sensor skal ha:
 - Et navn

- En numerisk verdi (ved hjelp av `DoubleProperty`)
- Verdiene kan simuleres eller endres gjennom brukerinteraksjon – for eksempel ved å bruke en knapp som randomiserer eller øker/reducerer verdier

3. Binding av eiendom

- Bind hver sensors verdi til et visuelt element:
 - Eksempel: en 'ProgressBar' eller 'Label' som viser avlesningen.
- Brukergrensesnittet skal oppdateres automatisk etter hvert som verdiene endres – ingen manuelle oppdateringskall.
- Dashbordet skal tydelig vise økende eller synkende avlesninger.

4. Brukerinteraksjon

- Inkluder minst to interaktive elementer:
 - En Start/Stopp-knapp for å kontrollere simuleringen.
 - En varslingssterskelkontroll (glidebryter eller tekstfelt) for å definere advarselsgrenser.
- Når en sensorverdi overskrider terskelen, skal displayet endres visuelt (f.eks. fargen blir rød).

5. Visuell tilbakemelding

- Dashbordet må gi tydelig visuell tilbakemelding for sensorendringer.
- Eksempel:
 - Endre fargen på en stolpe eller etikett når en verdi overskrider terskelen.
 - Bruk CSS-stiler eller -ikoner til å markere advarselbetingelser.

6. System oppførsel

- Dashboardet skal starte i midlertidig stanset tilstand.
- Når du trykker på "Oppdater" eller "Start", endres eller oppdateres sensoravlesningene.
- Når "Stopp" trykkes, fryser oppdateringer.
- Brukergrensesnittet må være interaktivt og responsivt hele veien

Milepæler for utvikling

Del	Fokus	Forventet utgang / test
-----	-------	-------------------------

1. Oppsett	Bygg et instrumentbordoppsett som viser sensorplassholdere.	Appvinduet lastes inn med en liste over sensorer og etiketter.
2. Datamodell	Implementer sensorklasse med <code>DoubleProperty</code> for verdier.	Verdier kan skrives ut eller vises via innbinding.
3. Oppdater logikk	Implementer manuelle eller knappbaserte oppdateringer for sensorverdier.	Ved å klikke på "Oppdater" endres avlesningene.
4. Brukerkontroller	Legg til Start/Stopp- og terskelinnstillinger.	Ved å klikke på "Start" kjøres oppdateringer; «Stopp» stopper dem.
5. Tilbakemelding og fargevarsler	Fremhev sensorer som overskrider terskler med dynamiske farge- eller stilendringer.	Farger eller indikatorer endres dynamisk.
6. Integrasjon	Finjuster oppsettet og test alle interaksjoner.	Dashbordet kjører problemfritt med live-oppdateringer.

Tips

- Hold logikken enkel – ingen timing eller bakgrunnstråder er nødvendig.
- Bruk `Bindings.format()` eller `.asString()` for å vise verdier pent.
- Eksperimenter med CSS for å visuelt skille mellom sensorer og varslingsnivåer.
- Fokuser på interaktivitet og klarhet i stedet for kompleksitet.
- Merk: Det forventes ingen kontinuerlige bakgrunnsoppdateringer.

Levering

Et dashbord for JavaFX-sensorovervåking som viser:

- Flere sensorer med sanntidsoppdatering av verdier
- Manuelle eller hendelsesdrevne oppdateringer av avlesninger
- Egenskapsbinding mellom sensordata og UI-visning
- Tydelige visuelle signaler for terskler eller varsler

Programmet skal føles responsivt og enkelt å observere – med fokus på klarhet, ikke kontinuerlige bakgrunnsoppdateringer.

Eksempel på brukergrensesnitt

