Mappe-prosjekt IDATx1001 - H2022 - Del 3

Fra Del 1 - Systembeskrivelse

Det skal utvikles en programvare som skal brukes av et varehus – Smarthus As. Programvaren skal brukes til å håndtere varelageret til Smarthus AS (så kalt "Warehouse management system (WMS) " på engelsk).

Smarthus AS leverer hovedsakelig varer til bygg-industrien, så typiske varer er laminatgulv, dører og vinduer, lister og annet trevirke.

Løsningen skal til slutt bestå av et tekstbasert brukergrensesnitt og et register som lagrer informasjon.

Løsningen skal utvikles igjennom 3 iterasjoner; Del 1, 2 og 3. Denne oppgaveteksten beskriver del 3 av oppgaven og bygger videre på del 1 og 2!

I tillegg til selve programmet, skal du også skrive en rapport. Denne startet du på i del 1 av prosjektet og du jobber videre med den i del 3 (se detaljer under).

Den endelige løsningen skal leveres i Mappen for emnet, for vurdering sammen med rapporten i slutten av semesteret.

Tilbakemelding/samtale på del 3 - "innlevering"

Du skal ikke levere inn noe for hver av delene i prosjektet (heller ikke del 3), men du vil bli tilbudt mulighet til å få muntlig tilbakemelding på arbeidet ditt så langt. Denne tilbakemeldingen gis på lab samme uke som ny del publiseres, og gis av faglærer eller læringsassistent.

Smart Hus AS - Del 3:

Basert på tilbakemeldingene du fikk på del 2: Gjør nødvendige forbedringer på entitetsklassen, og utvid applikasjonen iht oppgavebeskrivelsen under.

Oppgavebeskrivelse

I **del 3** av mappen skal du ferdigstille applikasjonen din ved å ta i bruk teori gjennomgått mot slutten av emnet:

- Refaktorering, copupling, cohesion, dokumentert kode, kode stil osv.
- Enum for å håndtere **kategorier**.
- Refaktorering for å forbedre løsningen/designet ditt
- Hvordan vil du **teste** løsningen din for sikre at den er robust og godt designet? (Husk å beskriv dette i rapporten!)

I tillegg skal du nå **sette ditt eget preg** på løsningen:

- Med utgangspunkt i opprinnelig kravspesifikasjon, hvilke endringer/forbedringer ville du ha gjort for at applikasjonen skal bli enda mer nyttig for brukeren? Her har du lov til å gå bort fra opprinnelig kravspesifikasjon og tilføre dine tanker og ideer. Løsningen din må selvsagt fortsatt oppfylle de grunnleggende funksjonelle brukerkrav, men du står fritt til å endre på designet av klasser, og legge til ny funksjonalitet.
- Brukergrensesnitt: du må gjerne tenke alternativt her. Feks, er en meny beste løsning (tatt i betraktning at dere ikke har lært å programmere grafisk brukergrensesnitt enda)?
- Dersom SmartHus ønsker å legge til en ny kategori av varer (feks, Hagemøbler) hvor mye av koden din må du endre? Hva tenker du dette forteller om ditt valgte design?
- Hvordan møter din løsning prinsippet om **lagdelt arkitektur**?
- Beskriv i rapporten hva du har foreslått av endringer og hvordan du har valgt å implementere disse.

Hvilke læringsmål vi ønsker å teste i del 3

I tillegg til at du nå skal kunne demonstrere samtlige læringsmål i emnebeskrivelsen, vil vi i del 3 ha spesielt fokus på:

- evnen til å levere et selvstendig arbeid
- evnen til å re-designe (refaktorere) egen løsning uten å tilføre nye feil
- even til å skrive fail-safe kode (også referert som guard-conditions)
- robuste design prinsipper som *modularisering*, *coupling*, *cohesion*, *responsibility driven design*
- brukervennlighet/god brukerinteraksjon

Krav til del 3

Krav til Programmet/koden

Følgende krav gjelder (fortsatt) til denne delen av oppgaven:

- Koden skal følge en bestemt ***kodestil*** (enten **Google** sin stil, eller "BlueJ"-stilen gitt i regelfilen "IDATx1001" for Chekstyle. For dere som ikke bruker BlueJ, er det naturlig å velge Google sin stil.
- Kodestilen **skal** verifiseres med **CheckStyle**-plugin (for BlueJ, IntelliJ osv) og vise ingen/minimalt med regelbrudd ved levering.
- **Klassen** og alle **metoder**, **variabler** (felt, parametere, lokale variabler) **skal** ha gode, beskrivende navn som tydelig gjenspeiler hvilken tjeneste en metode tilbyr, eller hvilken verdi variablene representerer/holder på.
- Alle navn på klasser, metoder og variabler **skal** være på **engelsk**.
- Koden skal være dokumentert iht standarden for JavaDoc (se hvordan JDK'en er dokumentert, for inspirasjon og som referanse).

Du velger selv hvilken IDE (utviklingsverktøy) du vil bruke på prosjektet (BlueJ, IntelliJ, VisualStudio Code etc).

Krav til rapport

Rapporten fra del 2 videreføres med tilbakemeldingene du fikk fra LA/Faglærer:

- I del 3 bør rapporten i størst mulig grad ferdigstilles. Kapitlene resultat og drøfting skal nærme seg ferdigstillelse, tilstrekkelig til at det er mulig å gi tilbakemelding.
- Det er viktig at dere dokumenterer opprinnelige funksjoner/klasser contra refaktorerte funksjoner/klasser, og ikke minst hvorfor du valgte å refaktorere.

Lykke til