

# Mappe-prosjekt IDATx1001 - H2022 - Del 1

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 1 av oppgaven!

I tillegg til selve programmet, skal du også skrive en rapport. Denne skal du starte på allerede i denne delen (del 1) av prosjektet (se detaljer under).

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

## Tilbakemelding/samtale på del 1 - "innlevering"

Du skal ikke levere inn noe for hver av delene i prosjektet (heller ikke del 1), 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 1:

### Oppgavebeskrivelse

I del 1 av oppgaven skal vi implementere **entitetsklassen** i løsningen. M.a.o. klassen som skal representere **en vare (engelsk: item)**.

Smarthus As ønsker at en vare skal være registrert med følgende informasjon:

1. Varenummer – består av bokstaver og tall
  2. Beskrivelse – en tekst som beskriver kort om varen
  3. Pris – Heltall
  4. Merkenavn – en tekst som inneholder merke (Hunton, Pergo, Egger osv)
  5. Vekt – i kilogram, som et desimaltall
  6. Lengde - som et desimaltall
  7. Høyde - som et desimaltall
  8. Farge – beskrevet som tekst
  9. Antall på lager - antall varer på lager. Skal aldri være mindre enn 0.
  10. Kategori - et tall som representerer kategori av varen. Bruk følgende: (1) Gulv laminater, (2) Vinduer (3) Dører og (4) Trelast
-

## Krav til del 1

### Krav til Programmet/koden

Følgende krav gjelder til denne delen av oppgaven (del 1):

- Koden skal følge en bestemt **kodestil** (enten Google eller "BlueJ"-stilen gitt i regelfilen "IDATx1001" for Checkstyle)
- Kodestilen **skal** verifiseres med **CheckStyle**-plugin (for BlueJ, IntelliJ osv) og vise ingen regelbrudd ved levering.
- **Klassen** og alle **metoder**, **variabler** (felt, parametre, 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**.

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

For å teste klassen (spesielt til dere som ikke bruker BlueJ): Lag en klasse som representerer selve applikasjonen, og implementert **klassemetoden** *public static void main(String[] args){}* i denne klassen. Skriv test-kode i main-metoden for å teste at Vare-klassen din fungerer som forventet.

### Krav til rapport

Det vil bli gjort tilgjengelig en mal (i Word) for rapporten som du skal benytte.

Følgende krav gjelder for rapporten i Del 1:

Rapporten skal inneholde:

- En forside med kandidatnavn, og navn på emnet ("IDATA1001 Programmering 1" for Ålesund osv.)
  - En innholdsfortegnelse (automatisk generert)
  - En innledning som beskriver kort kravspesifikasjonen gitt i oppgaven (her kan du hente det meste fra oppgavebeskrivelsen).
  - En terminologi-liste, som inneholder begreper relatert til oppgaven, både på norsk og engelsk
  - Under hovedavsnittet "design/implementasjon". Beskriv med egne ord hvilke valg du har gjort når du implementerte klassen som skal representere en vare:
    - Hvorfor valgte du de datatypene du valgte (for feltene til klassen)?
    - Har du implementert **mutator-metoder** for noen av feltene? I så fall; for hvilke felt og hvorfor?
    - Reflekter/diskuter hvilke tiltak du har innført for å sikre at klassen er implementert som en **robust klasse** (en robust klasse er en klasse der det ikke er mulig å angi ugyldige verdier til feltene i klassen).
-