

**IDATT1001 Programmering 1**

Rapport for mappevurdering, del 1

Kandidatnummer 10114

Innhold

[1 Innledning 3](#_Toc117632306)

[1 Innledning 4](#_Toc117632307)

[2 Terminologiliste 5](#_Toc117632308)

[3 Design/implementasjon 6](#_Toc117632309)

[*3.1 Valg av datatyper* 6](#_Toc117632310)

[*3.2 Mutator-metoder* 6](#_Toc117632311)

[*3.3 Refleksjon rundt robusthet* 6](#_Toc117632312)

# Innledning

På vegne av bedriften Smarthus AS, skal det innledningsvis i mappevurderingen opprettes en klasse som representerer en vare. Denne klassen skal inneholde følgende objektvariabler:

1. varenummer
2. beskrivelse
3. pris
4. merkenavn
5. vekt
6. lengde
7. høyde
8. farge
9. antall på lager
10. kategori

Det er gitt følgende krav til koden i del 1 av mappevurderingen:

1. Alle navn på klasser, metoder og variabler skal være skrevet på engelsk.
2. Kodens stil skal være i samsvar med enten Google- eller BlueJ-stilene.
3. Kodestilen skal verifiseres med en «CheckStyle»-plugin, og ikke vise noen regelbrudd.
4. Klassen, metodene og variablene osv. skal ha gode og beskrivende navn, slik at de tydelig gjenspeiler dets egenskaper, være det type tjeneste ved metode, eller verdi ved variabel.

Klassen *item* skal testes gjennom bruken av en *public void main(String[] args)*{}-metode.

# Innledning

På vegne av bedriften Smarthus AS, skal det innledningsvis i mappevurderingen opprettes en klasse som representerer en vare. Denne klassen skal inneholde følgende objektvariabler: varenummer, beskrivelse, pris, merkenavn, vekt, lengde, høyde, farge, antall på lager og kategori.

Det er gitt følgende krav til koden i del 1 av mappevurderingen:

Alle navn på klasser, metoder og variabler skal være skrevet på engelsk. Kodens stil skal være i samsvar med enten Google- eller BlueJ-stilene. Kodestilen skal verifiseres med en «CheckStyle»-plugin, og ikke vise noen regelbrudd. Klassen, metodene og variablene osv. skal ha gode og beskrivende navn, slik at de tydelig gjenspeiler dets egenskaper, være det type tjeneste ved metode, eller verdi ved variabel. Klassen *item* skal testes gjennom bruken av en *public void main(String[] args)*{}-metode.

# Terminologiliste

# Design/implementasjon

## *3.1 Valg av datatyper*

Objektvariablene sine datatyper ble valgt på bakgrunn av oppgavetekstens krav. Det ble altså ikke valgt datatyper på eget initiativ utenom for objektvariablene «antall på lager» og «kategori», hvor begge passer best som heltall.

## *3.2 Mutator-metoder*

Det ble vurdert at mutator-metoder for objektvariablene «pris» og «antall på lager» ville være naturlige. Prisen vil kunne variere på bakgrunn av for eksempel et tilbud. Antallet av en vare på lager vil variere etter hvor mange som blir kjøpt inn i forhold til antallet som selges til kunder.

## *3.3 Refleksjon rundt robusthet*

For å gjøre koden robust, er det tatt spesielt hensyn til håndtering av unntakstilfeller. Det er tatt i bruk «try-catch»-blokker for behandlingen av unntaksmeldinger og for å kunne gi brukeren en tydelig beskjed rundt hva som gikk galt. Gjennom en forespeiling av ulike tilfeller som kan oppstå og ved å tilrettelegge for disse typene unntaksbehandling, vil koden kunne fremstå mer robust og brukervennlig.