

# Heldagsprøve IT-2

**Tid** : 12.12.2024 kl. 8:00 – 12:00

**Hjelpe midler** : Alle ikke-kommuniserende hjelpe midler (deriblant KI) er tillatt

## Oppgave 1 Kvadratrøtter (20%)

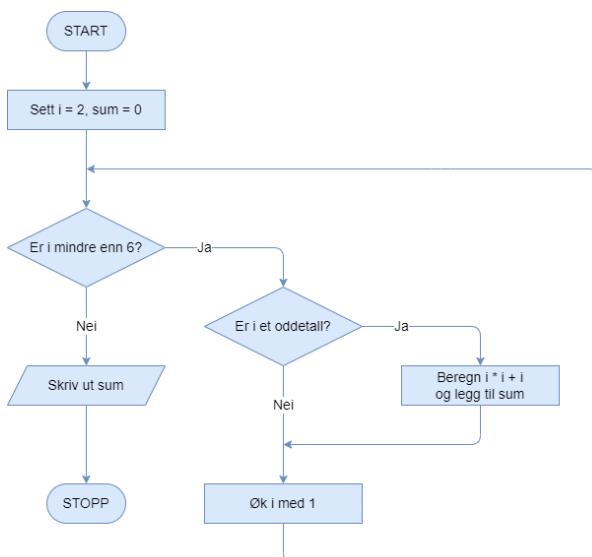
Forklaringene i denne oppgaven kan ikke være «jeg kodet opp løsningen i python...»

- a) Ta utgangspunkt i følgende pseudokode:

```
SET m TO 3
SET sum TO 0
FOR hvert tall i fra og med 2 til og med 4
    SET sum TO sum + i * m
    DECREMENT m
ENDFOR
DISPLAY sum
```

Hva blir skrevet ut? Forklar hvordan du kom fram til svaret.

- b) Ta utgangspunkt i flytskjemaet nedenfor. Hva blir skrevet ut? Forklar hvordan du kom fram til svaret.



- c) Pseudokoden nedenfor skal kategorisere tall, men den inneholder en eller flere feil. Finn ut hva som er galt og rett opp pseudokoden så tallene blir kategorisert riktig.

```
PRINT "Skriv inn et heltall:"
READ tall
IF tall GREATER THAN 0
    PRINT "Tallet er mellom 1 og 10"
ELSE IF tall GREATER THAN 10
    PRINT "Tallet er mellom 11 og 20"
ELSE IF tall GREATER THAN 20
    PRINT "Tallet er over 20"
ELSE
    PRINT "Tallet er 0 eller negativt"
ENDIF
```

- d) Ta utgangspunkt i følgende pseudokode:

```
FUNCTION beregn(n) :  
    IF n LESS THAN OR EQUAL TO 1  
        RETURN n  
    ELSE  
        RETURN n + beregn(n-1)  
ENDFUNCTION  
  
DISPLAY beregn(5)
```

Hva blir skrevet ut? Forklar hvordan du kom fram til svaret.

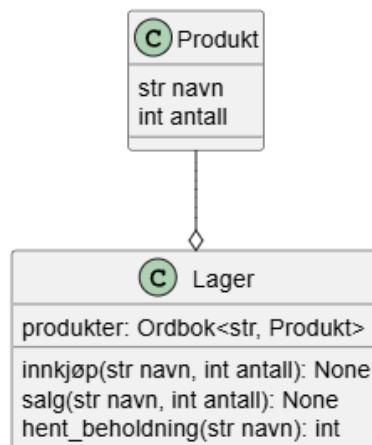
## Oppgave 2 Lagersystem (30%)

Du skal utvikle og teste et enkelt, objektorientert lagerstyringssystem for en butikk. Systemet skal bestå av to klasser: **Lager** og **Produkt**, som vist i klassediagrammet.

### Krav til programmet:

**Lager**-klassen skal inneholde

- **En ordbok, produkter**, med produktnavn som nøkkel og produktobjekt som verdi, slik at det er enkelt å slå opp produkter som er tilgjengelige for salg.
- **En metode: innkjøp** som legger til et angitt antall av et produkt i lageret.
  - Når et produkt kjøpes inn, skal det først sjekkes om det finnes i produkter-ordboken.
  - Hvis det ikke finnes, skal et nytt produkt opprettes med det angitte antallet, og produktet skal legges til i ordboken med riktig lagerbeholdning.
  - Hvis det finnes, skal lagerbeholdningen oppdateres i henhold til det angitte antallet.
  - Skriver ut "Innkjøp av X Y er gjennomført", hvor X er antall og Y er produktnavn.
- **En metode: salg** som
  - Fjerner et angitt antall av et produkt fra lageret når det selges.
  - Skriver ut "Salg av X Y er gjennomført", hvor X er antall og Y er produktnavn.
- **En metode: hent\_beholdning(str)** som returnerer antall av et produkt som er tilgjengelig på lageret.



**Produkt**-klassen skal inneholde

- Informasjon om produktets navn og antall på lager.
- Når produktet opprettes, får det et navn, og antall settes til 0 som standard (valgfritt argument).

### Eksempel på normal bruk

- 1) Opprett et lager.
- 2) Kjøp inn et antall av et produkt.
- 3) Vis lagerbeholdning.
- 4) Selg et antall av produktet.
- 5) Vis lagerbeholdning.
- 6) Kjøp inn et nytt antall av produktet.
- 7) Vis lagerbeholdning for et produkt.

### Forventet utskrift

- Lageret er opprettet  
Innkjøp av 10 Melk er gjennomført  
Antall Melk på lager: 10  
Salg av 7 Melk er gjennomført  
Antall Melk på lager: 3  
Innkjøp av 5 Melk er gjennomført  
Antall Melk på lager: 8

a)

Lag programmet i henhold til kravene og demonstrer normal bruk som ovenfor.

Merk at **innkjøp**- og **salg**-metodene skriver ut teksten, men når vi skal vise lagerbeholdningen må vi skrive ut teksten selv sammen med antallet som vi får fra **hent\_beholdning(str)**.

**NB! Snu arket for b) og c) oppgave!!**

**b)**

Lag et eget testprogram som verifiserer normal bruk. Testene skal sjekke:

- 1) At et nyt **Lager**-objekt kan opprettes og inneholder en tom ordbok med produkter.
- 2) At lagerbeholdningen oppdateres korrekt når et nyt produkt kjøpes inn.
- 3) At lagerbeholdningen økes korrekt ved kjøp av flere enheter av et produkt som allerede finnes på lageret.
- 4) At lagerbeholdningen reduseres korrekt når et produkt selges.
- 5) At **hent\_beholdning(str)** returnerer riktig antall for et gitt produkt.

**c)**

Mye kan gå galt i et program når brukere legger inn data. I denne delen av oppgaven skal du fange opp og håndtere følgende feilsituasjoner:

- 1) Innkjøp skal ikke tillate et antall som ikke er et heltall.
- 2) Innkjøp skal ikke tillate et negativt antall.
- 3) Salg av et produkt som ikke finnes i lagerets produkter-ordbok, skal ikke tillates.
- 4) Det skal ikke være mulig å selge flere enheter av et produkt enn det som er tilgjengelig på lageret.

Utvid programmet (lag en NY fil) med funksjonalitet for å fange opp disse feilene.

## Oppgave 3 Land, befolkning og areal. (25%)

Du skal behandle informasjon fra et gitt datasett om land, befolkningstall og landområder. Datasettet er tilgjengelig som [land.json](#)

- Lage et program som teller hvor mange land som er små, middels, eller store, med tanke på innbyggertallet:
  - Små: innbygger under 1000 000
  - Middels: innbygger mellom 1000 000 og 100 000 000
  - Store: innbygger over 100 000 000
- Utvid programmet med å vise en oversikt over de 10 landene med størst befolkning, sortert i synkende rekkefølge. For hvert av landene skal oversikten inkludere landets navn og befolkningstall angitt i millioner innbyggere.
- Dernest skal programmet skrive ut en sortert liste etter flest innbyggere per kvadratkilometer (befolkingstettheten) for disse 10 landene. Disse skal også være sortert synkende, etter befolkingstettheten. Her skal bare landene sammen med befolkingstettheten vises.

## Oppgave 4 Husholdningsregnskap (25%)

De fleste merker at prisene har økt kraftig i det siste. I denne oppgaven skal du lage et program som holder oversikt over hva familien Hultgren bruker på mat og strøm.

I vedlegget [husholdningsregnskap.csv](#) finner du en CSV-fil med familien Hultgrens mat og strømutgifter i årets 3 første måneder.

Oppgaven blir å hjelpe familien med å få oversikt og utgiftene.

- Skriv ut en oversikt over hva de har brukt totalt på mat og strøm i løpet av de tre månedene.
- Skriv ut en oversikt over hva de har brukt i hver av de tre månedene på mat og strøm.
- Gi familien mulighet til å legge til nye poster i regnskapet.

Lag en funksjon som tar

**dato, type, beløp og kommentar**

som parameterer og så legger til en linje med disse dataene sist i filen. Se eksempel under.

```
legg_til("30.03.2025", "mat", 405, "oda")
```

```
29.03.2025;mat;854;kiwi
```

```
30.03.2025;mat;405;oda
```

**Hint 1:** Lagre en kopi av husholdningsregnskap.csv før du starte på siste-oppgaven, hvis du klarer å slette den ☺

**Hint 2:** Ved å åpne en fil med [open\("filnavn.txt", "a"\)](#) vil du kunne skrive tekst til en fil uten at det som står der før skrives over. "a" – står for append.