# IN1010 uke 10

Gruppe 2

# Agenda

- Invarianter
  - Hva er invarianter?
  - Invarianter i lenkelister
  - Løkkeinvarianter
  - Noen begreper
- ☕ Pause 13:00-13:15

- Jobbe med oppgaver/trix/oblig!

# Invarianter

# Invariant

- F.eks. regel for hvilke verdier en instansvariabel for ha
  - “Pris kan ikke være < 0”
- Skal gjelde før og etter du kaller på en metode
- Må tenke på dette når man skriver metodene

```
public void senkPris() {
    pris -= 100;
}
```



```
public void senkPris() {
    if (pris - 100 > 0) {
        pris -= 100;
    } else {
        pris = 0;
    }
}
```

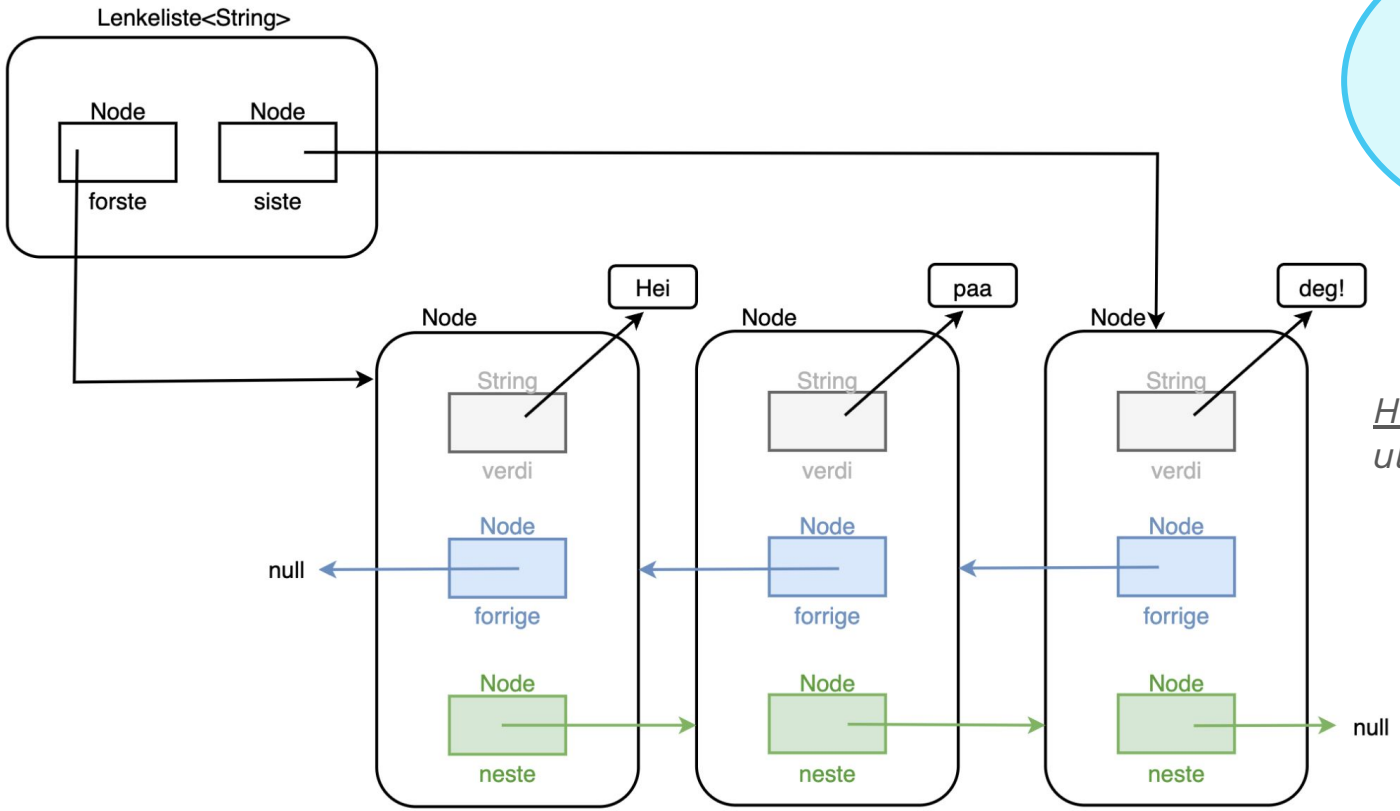
# Hva er invarianter?

- = betingelser, oftest knyttet til datastrukturen i objektene våre
- Kaller dem *datainvarianter/klasseinvarianter*
  - Betingelsene må alltid være sanne for en instans av en klasse

**Hvorfor?**

- Sikre at datastrukturen forblir gyldig og konsekvens
  - Eksempel: sikre at prisen på en vare aldri er negativ

# Invarianter i lenkelister *Dobbellenket*




**Spørsmål fra innlevering 2:** Hva er invariantene her?

*Hint: Hva gjelder for de ulike peker-variablene i Node og Lenkeliste?*

# Invarianter i lenkelister *Dobbellenket*

**Spørsmål fra innlevering 2:** Hva er invariantene her?

Invariant for *neste*

- Peker til neste node i listen (eller er null hvis noden er den siste)
- 'neste' i 'forste' peker alltid på en node hvis listen har >1 elementer

Invariant for *forrige*

- Peker til forrige node i listen (eller er null hvis noden er forste)
- 'forrige' i 'siste' peker alltid på en node hvis listen har >1 elementer

# Invarianter i lenkelister *Dobbellenket*

**Spørsmål fra innlevering 2:** Hva er invariantene her?

Invariant for *forste*

- Er null hvis listen er tom. Ellers peker på en node (den første i listen).

Invariant for *siste*

- Er null hvis listen er tom. Ellers peker på en node (den siste i listen).

# Invarianter i lenkelister *Dobbellenket*

**Spørsmål fra innlevering 2:** Hva er invariantene her?

- Kan hjelpe å kommentere invarianter i koden!

```
public class Lenkeliste {
    // Invariant: hvis listen er tom, er 'forste' null. Ellers peker 'forste' paa en node.
    Node forste;
    // Invariant: hvis listen er tom, er 'siste' null. Ellers peker 'siste' paa en node.
    Node siste;

    class Node {
      /* Invariant: 'neste' peker til neste node i listen (eller er null hvis noden er den
      siste). 'neste' i 'forste' peker alltid paa en node hvis listen har >1 elementer. */
      Node neste;

      /* 'forrige' peker til forrige node i listen (eller er null hvis noden er forste).
      'forrige' i 'siste' peker alltid paa en node hvis listen har >1 elementer. */
      Node forrige;
    }
}
```

# Løkkeinvarianter

- = betingelser som forblir sanne før og etter hver iterasjon

```java
public static void main(String[] args) {
    int sum = 0;
    for (int i = 0; i <= 10; i++) {
        sum += i;
    }
    System.out.println("Sum: " + sum);
}
```

Hva er løkkeinvarianten her?

# Løkkeinvarianter

- = betingelser som forblir sanne før og etter hver iterasjon

```java
public static void main(String[] args) {
    int sum = 0;
    for (int i = 0; i <= 10; i++) {
        sum += i;
    }
    System.out.println("Sum: " + sum);
}
```

Hva er løkkeinvarianten her?

Svar: I forkant av hver iterasjon, er 'sum' lik summen av tallene opp til og med tallet på indeks i-1

# Løkkeinvarianter

```java
public class SosialeMedier {
    String[] sosialeMedier = {"Snapchat", "TikTok", "Instagram", "Facebook"};
    public boolean finnesSosialtMedium(String medium) {
        for(int i = 0; i < sosialeMedier.length; i++) {
            if(sosialeMedier[i].equals(medium))
                return true;
        }
        return false;
    }
}
```

Hva er løkkeinvarianten her?

*Hint: Tenk deg en viss iterasjon av løkken (f.eks. i = 3). Hva gjelder her?*
*→ Hva gjelder for enhver iterasjon?*

# Løkkeinvarianter

```java
public class SosialeMedier {
    String[] sosialeMedier = {"Snapchat", "TikTok", "Instagram", "Facebook"};
    public boolean finnesSosialtMedium(String medium) {
        for(int i = 0; i < sosialeMedier.length; i++) {
            if(sosialeMedier[i].equals(medium))
                return true;
        }
        return false;
    }
}
```

Hva er løkkeinvarianten her?

Svar: For hver iterasjon av for-løkken ved indeks *i* (hvor 0 <= *i* < sosialeMedier.length), har elementene i arrayet sosialeMedier med indekser fra 0 til *i*-1 blitt sjekket, og ingen av disse elementene matcher strengen medium.

Derfor, hvis strengen medium er til stede i arrayet, må den befinne seg på en posisjon med indeks *i* eller høyere.

# Noen begreper

- “Invarianten holder” = betingelsen forblir sann
  - Bidrar til å hjelpe utviklere med å forhindre feil i programmet

- Pre- & Postconditions
  - Preconditions *(førbetingelser)* = må være oppfylte før kjøringen av en metode eller løkke

  - Postconditions *(etterbetingelser)* = vil være oppfylt etter at metoden eller løkken er fullført

# Jobb med oppgaver!

IN1010 Emnesiden → Grupper →
Gruppe 2 → Uke10

Husk å ta pause ☕