



# IN1010 uke 8

Gruppe 4

# Agenda

- Oblig 4 tips og info
- Repetisjon
  - Pekere av ulike klasser
  - Lesing/skriving til fil
  - Iterable
  - Exceptions
- Løse problemer i koden - litt tips
- Jobbe med oppgaver/trix/oblig!

# Oblig 4

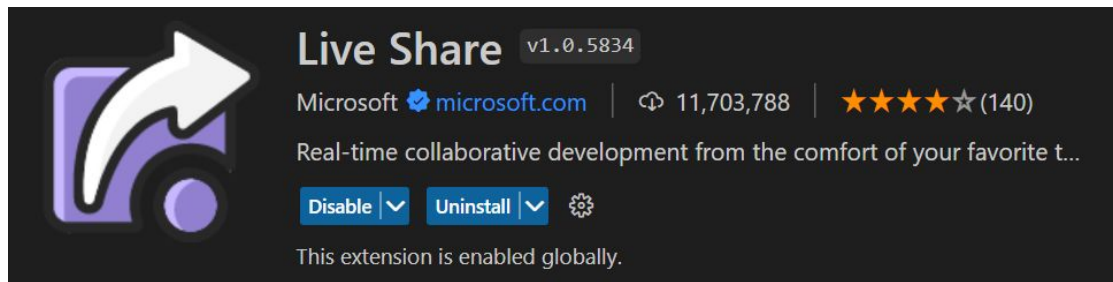


# Oblig 4 teamarbeid

- Tips og info for gruppeoppgaven:  
Under Oblig 4 på emnessiden

# Hvordan kode sammen?

- Ikke bare fordel oppgavene!
- Git/Github
- Parprogrammering (rekommenderes!)
  - Sitt sammen
  - Book grupperom med stor skjerm
- Live share (kode samtidig på ulike maskiner)



# Repetisjon



# Pekere av ulike klasser



# Pekere av ulike klasser

```
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
  
}
```

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
  
}
```



```
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
}
```

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
}
```

```
Mat maten = new Mat();  
Sjokolade sjokoladen = new Sjokolade();  
  
Sjokolade sPeker = maten;  
Sjokolade sPeker2 = sjokoladen;  
  
Mat mPeker = maten;  
Mat mPeker2 = sjokoladen;
```

*Er dette lov?*

*Hva er ikke lov?*

```
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
}
```

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
}
```

⊗ Type mismatch: cannot convert from Mat to Sjokolade

```
Mat maten = new Mat();  
Sjokolade sjokoladen = new Sjokolade();  
  
Sjokolade sPeker = maten;  
Sjokolade sPeker2 = sjokoladen;  
  
Mat mPeker = maten;  
Mat mPeker2 = sjokoladen;
```

*Er dette lov?*

*Hva er ikke lov?*

```
Sjokolade sPeker = sjokoladen;  
Mat mPeker = maten;  
Mat mPeker2 = sjokoladen;  
  
sPeker.spis();  
mPeker.spis();  
mPeker2.spis();  
  
sPeker.kast();  
mPeker.kast();  
mPeker2.kast();  
  
sPeker.smelt();  
mPeker.smelt();  
mPeker2.smelt();
```

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
}
```

```
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
}
```

*Er dette lov? Hva er ikke lov?*

```
Sjokolade sPeker = sjokoladen;
Mat mPeker = maten;
Mat mPeker2 = sjokoladen;

sPeker.spis();
mPeker.spis();
mPeker2.spis();

sPeker.kast();
mPeker.kast();
mPeker2.kast();

sPeker.smelt();
mPeker.smelt();
mPeker2.smelt();
```

```
class Sjokolade extends Mat {

    @Override
    public void spis() {
        System.out.println(x: "Spiser sjokolade");
    }

    public void smelt() {
        System.out.println(x: "Smelter sjokolade");
    }
}

class Mat {

    public void spis() {
        System.out.println(x: "Spiser mat");
    }

    public void kast() {
        System.out.println(x: "Kaster");
    }
}
```

⊗ The method smelt() is undefined for the type Mat

⊗ The method smelt() is undefined for the type Mat



```
Sjokolade sPeker = sjokoladen;  
Mat mPeker = maten;  
Mat mPeker2 = sjokoladen;  
  
sPeker.spis();  
mPeker.spis();  
mPeker2.spis();  
  
sPeker.kast();  
mPeker.kast();  
mPeker2.kast();  
  
sPeker.smelt();
```

*Dette er lov, hva printes?*

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
  
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
}
```

```
Sjokolade sPeker = sjokoladen;  
Mat mPeker = maten;  
Mat mPeker2 = sjokoladen;  
  
sPeker.spis();  
mPeker.spis();  
mPeker2.spis();  
  
sPeker.kast();  
mPeker.kast();  
mPeker2.kast();  
  
sPeker.smelt();
```

```
Spiser sjokolade  
Spiser mat  
Spiser sjokolade  
Kaster  
Kaster  
Kaster  
Smelter sjokolade
```

```
class Sjokolade extends Mat {  
  
    @Override  
    public void spis() {  
        System.out.println(x: "Spiser sjokolade");  
    }  
  
    public void smelt() {  
        System.out.println(x: "Smelter sjokolade");  
    }  
  
class Mat {  
  
    public void spis() {  
        System.out.println(x: "Spiser mat");  
    }  
  
    public void kast() {  
        System.out.println(x: "Kaster");  
    }  
}
```

# Lese/skrive til fil

*Kodeeksempel: Lese og skrive navnliste*

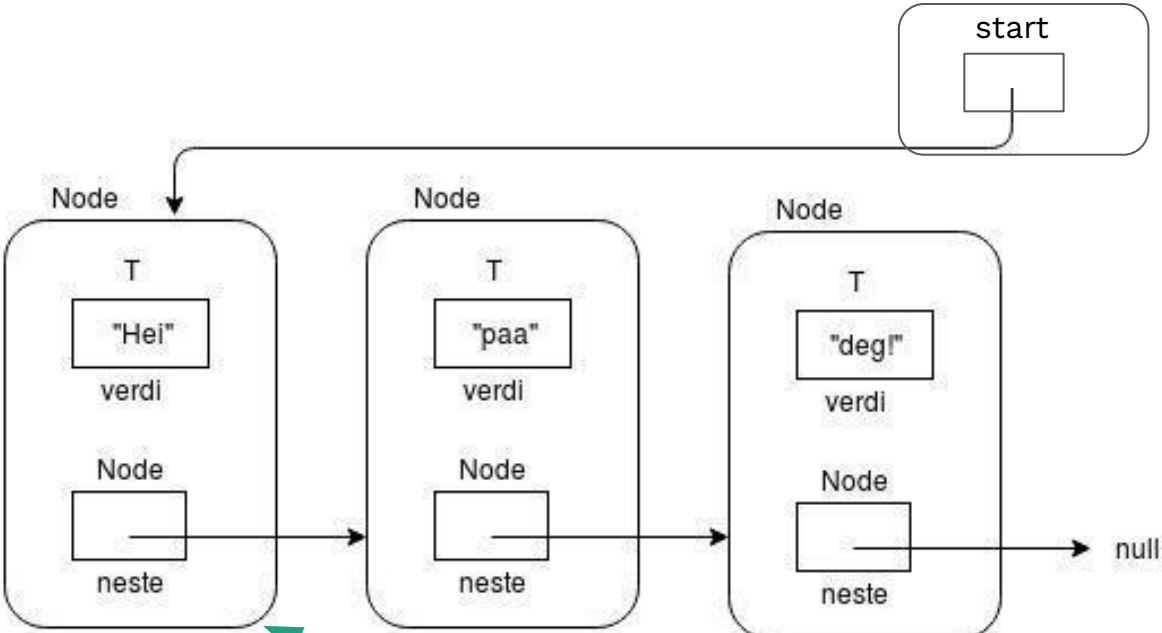


# Iterable

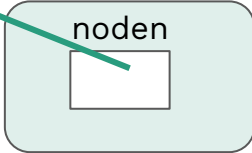




Lenkeliste<T>

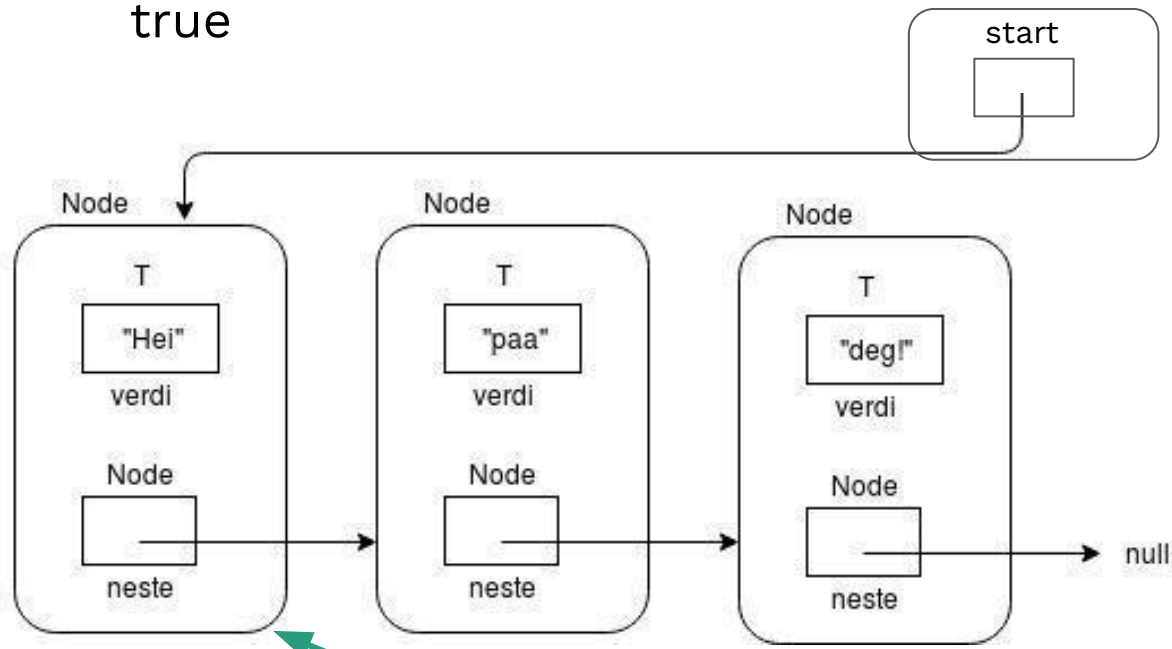


Lenkelisteterator

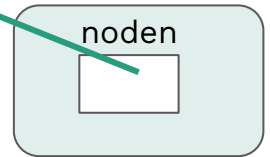


*hasNext():*  
true

Lenkeliste<T>



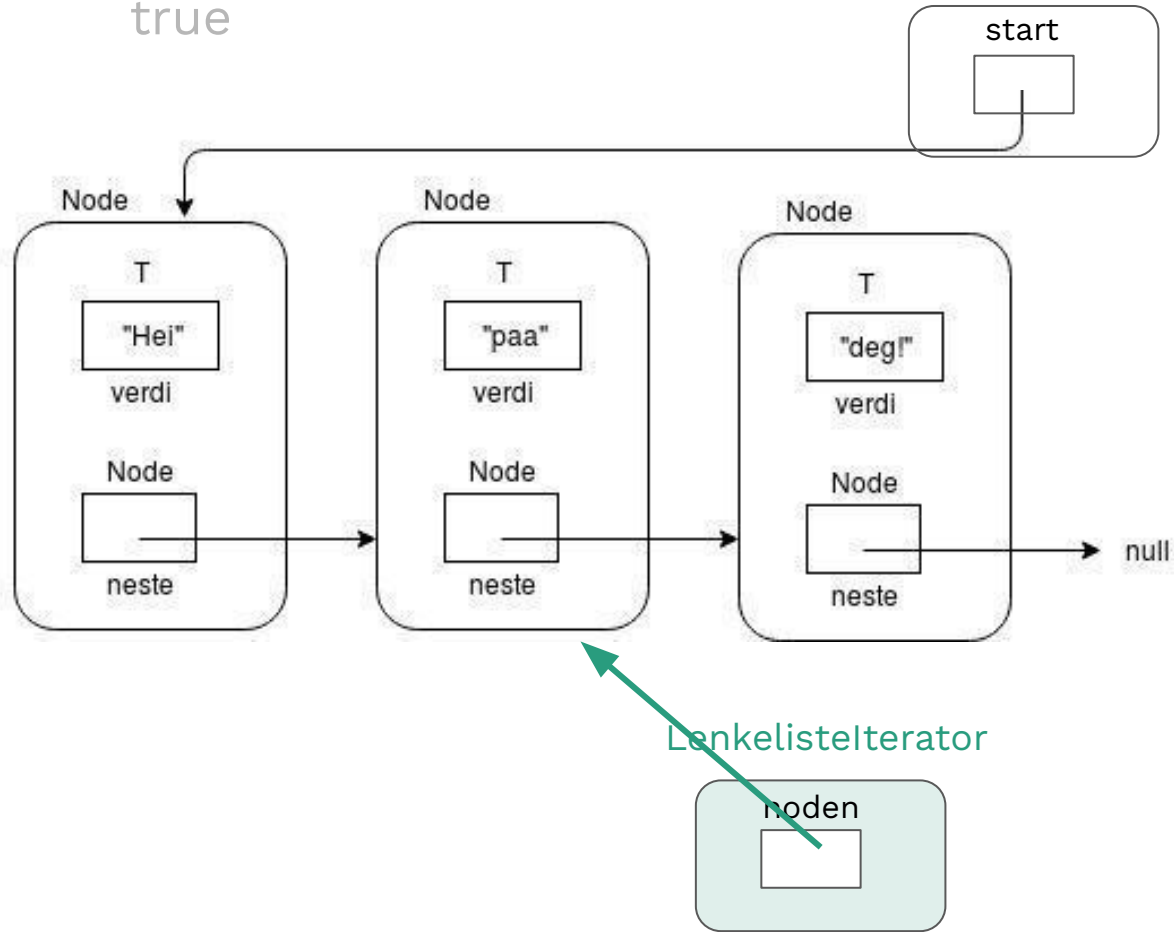
Lenkelisteliterator



*next():*  
"Hei"

*hasNext():*  
true

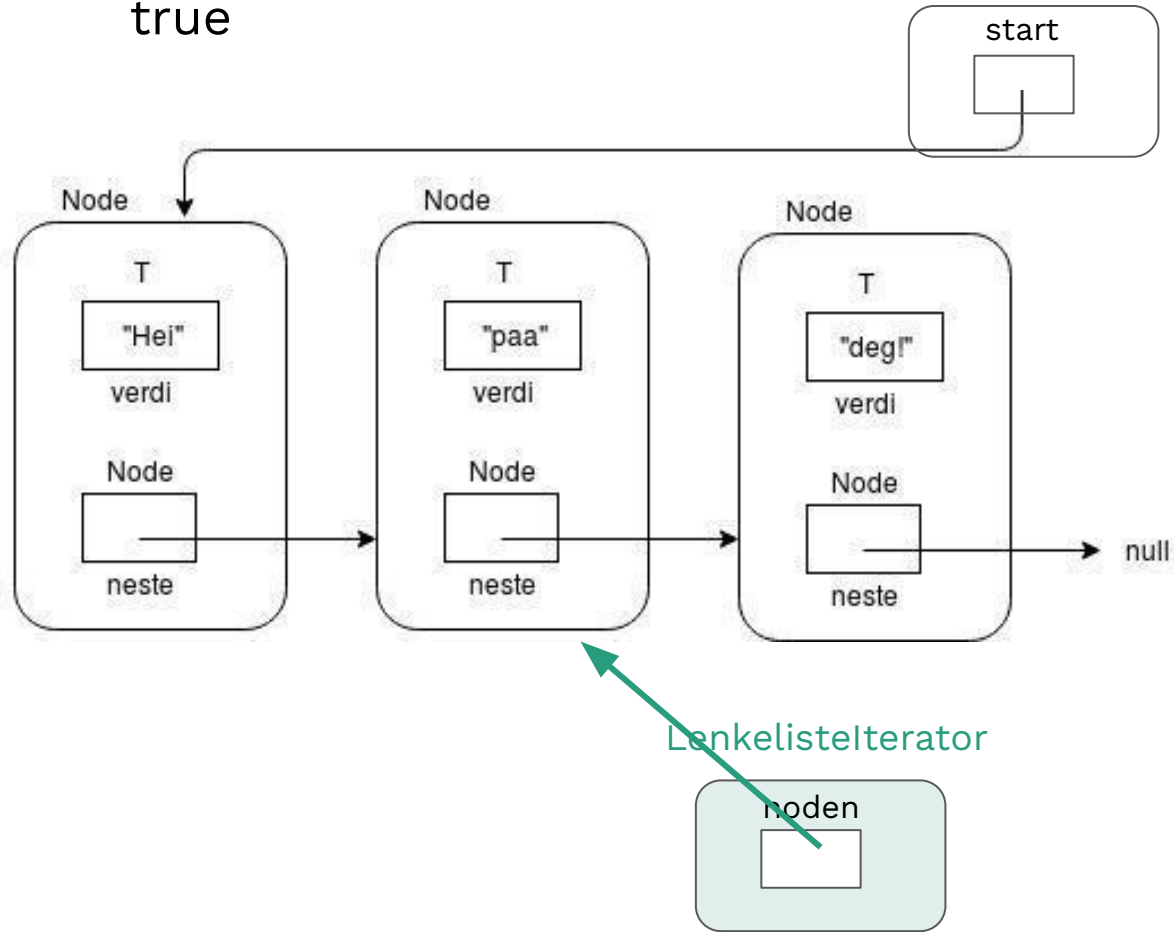
Lenkeliste<T>



*next():*  
"Hei"

*hasNext():*  
true

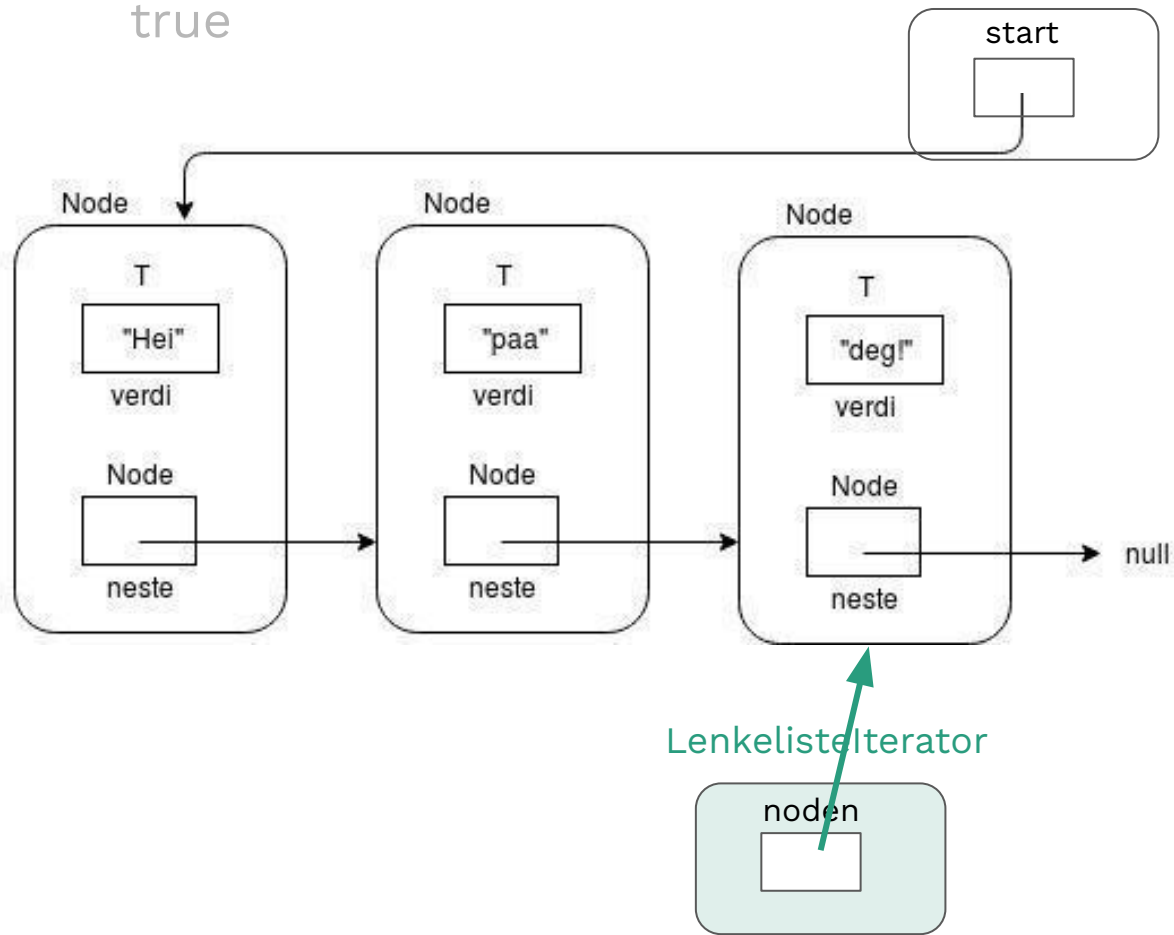
Lenkeliste<T>



*next():*  
"Hei"  
"paa"

*hasNext():*  
true

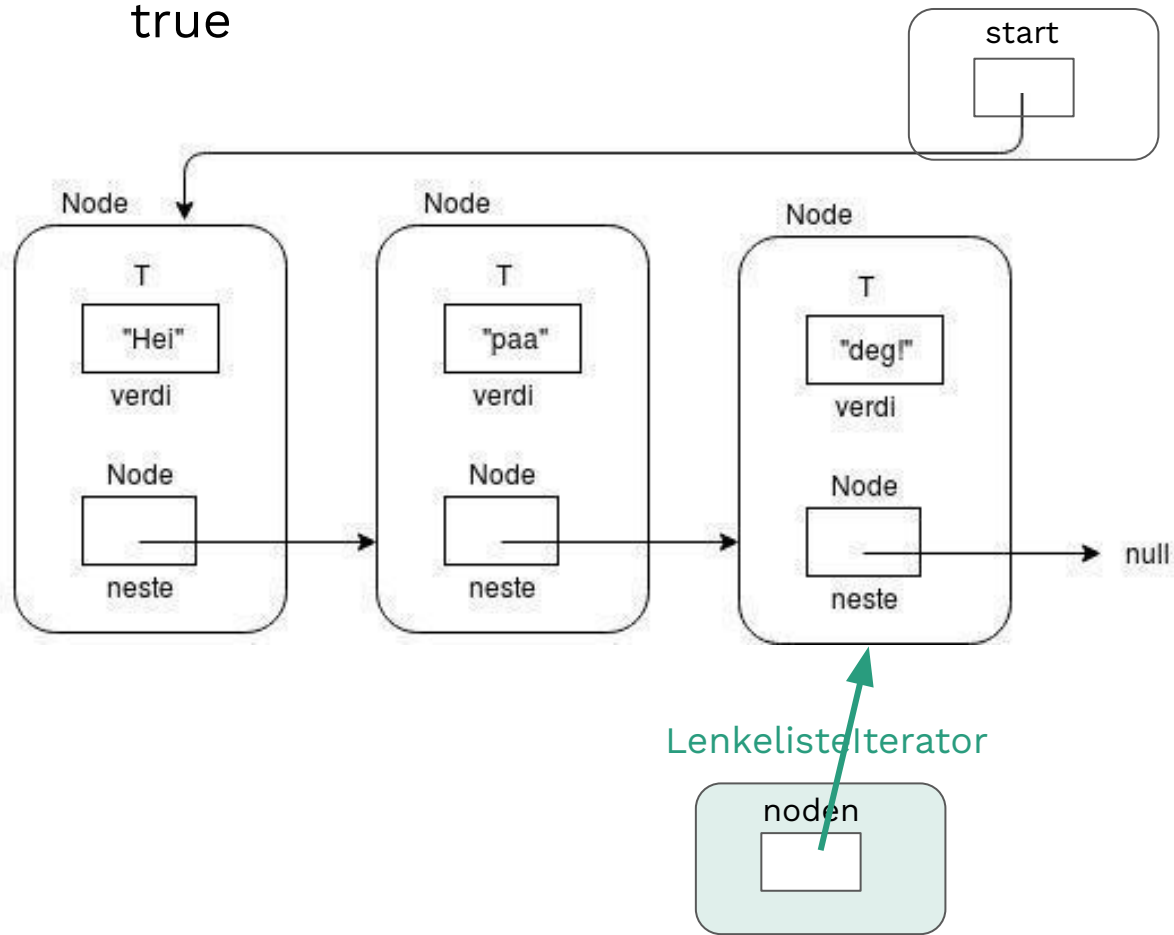
Lenkeliste<T>



*next():*  
"Hei"  
"paa"

*hasNext():*  
true

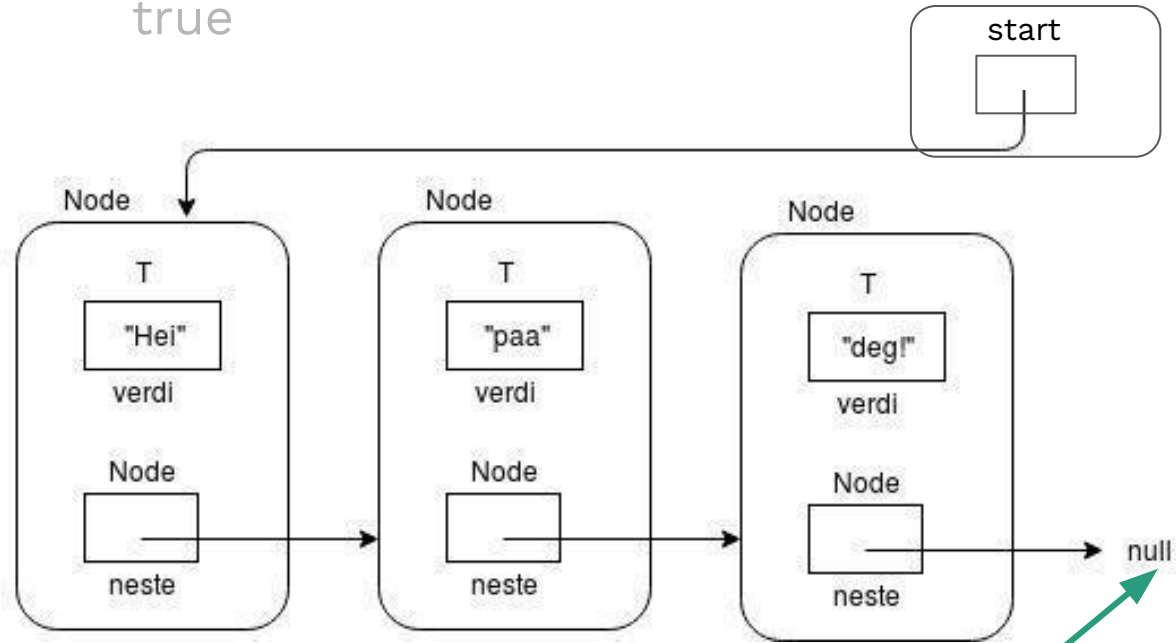
Lenkeliste<T>



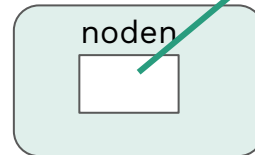
*next():*  
"Hei"  
"paa"  
"deg!"

*hasNext():*  
true

Lenkeliste<T>



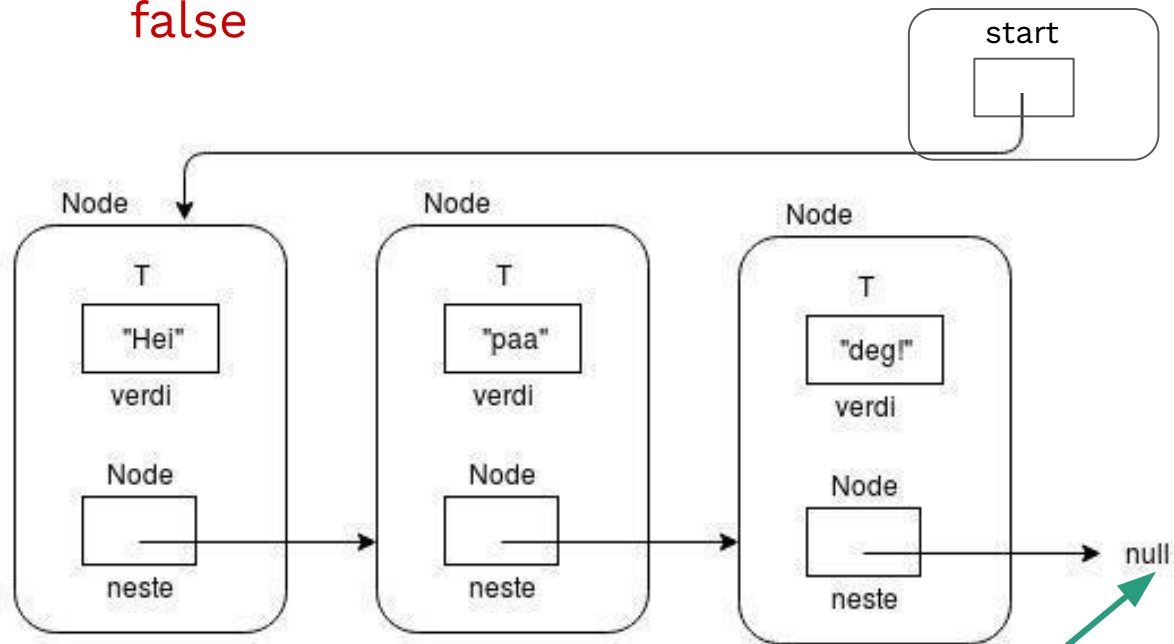
LenkelisteIterator



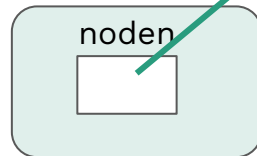
*next():*  
"Hei"  
"paa"  
"deg!"

*hasNext():*  
**false**

Lenkeliste<T>



LenkelisteIterator





# Iterable

- Kodeeksempel: Hva skjer inne i iterable- og iterator-klassene?

# Exceptions

*Kodeeksempel: UgyldigNavnException*



# Løse problem i koden: tips



# Noen tips

- Feil når kompilerer/kjører? → Les feilmeldingen!
  - Hva er feilen? F.eks:
    - *NullPointerException*  
= prøver å kalle metode på variabel som =null
    - *IndexOutOfBoundsException*  
= prøver å hente ut fra array på ugyldig indeks
  - Hvor er feilen? Hvilken fil, metode, linje?

```
Exception in thread "main" java.lang.NullPointerException: Cannot  
invoke "Vare.hentPris()" because "v" is null  
at Hovedprogram.main(Hovedprogram.java:12)
```

- Print ut variabler i metodene - har de riktig verdi?

```
int pris = 120;  
Vare v = new Vare(pris, navn: "sjokolade");  
int rabatt = (int) Math.round(pris * 0.25);  
  
v.taNedPris(rabatt);
```



```
int pris = 120;  
Vare v = new Vare(pris, navn: "sjokolade");  
int rabatt = (int) Math.round(pris * 0.25);
```

```
System.out.println("pris foer:" + v.hentPris());  
System.out.println("rabatt: " + rabatt);
```

```
v.taNedPris(rabatt);
```

```
System.out.println("pris etter:" + v.hentPris());
```

```
pris foer:120  
rabatt: 30  
pris etter:90
```

- Metode fungerer ikke - feil verdi for instansvariabel?

1. Prøv å finne problemet inni metoden

2. Hvis ikke finner:

Hvilke andre metoder kan påvirke? Hvor ellers i koden endres instansvariabelen?

→ Sjekk for feil i disse metodene også!

- Usikker på om loops / if-setninger fungerer som de skal?  
→ print ut ting inni for å se hva som egentlig skjer

```
int[][] rutenett = new int[3][3];
```

```
if (rutenett[0][2] == 0) {  
    System.out.println(x: "Inne i if nr. 1");  
    rutenett[0][2] = 7;  
}
```

```
if (rutenett[0][0] != 0) {  
    System.out.println(x: "Inne i if nr. 2");  
    rutenett[0][2] = 3;  
}
```

```
for (int[] rad : rutenett) {  
    System.out.println(x: "Ytre loop(rad)");  
    for (int kol : rad) {  
        System.out.println("Indre loop: " + kol);  
    }  
}
```

- Usikker på om loops / if-setninger fungerer som de skal?  
→ print ut ting inni for å se hva som egentlig

```
int[][] rutenett = new int[3][3];

if (rutenett[0][2] == 0) {
    System.out.println(x: "Inne i if nr. 1");
    rutenett[0][2] = 7;
}

if (rutenett[0][0] != 0) {
    System.out.println(x: "Inne i if nr. 2");
    rutenett[0][2] = 3;
}
```

```
for (int[] rad : rutenett) {
    System.out.println(x: "Ytre loop(rad)");
    for (int kol : rad) {
        System.out.println("Indre loop: " + kol);
    }
}
```

```
Inne i if nr. 1
Ytre loop(rad)
Indre loop: 0
Indre loop: 0
Indre loop: 7
Ytre loop(rad)
Indre loop: 0
Indre loop: 0
Indre loop: 0
Ytre loop(rad)
Indre loop: 0
Indre loop: 0
Indre loop: 0
```



- Utskriftene for rask testing trenger ikke å være fine

```
int tall = 10;

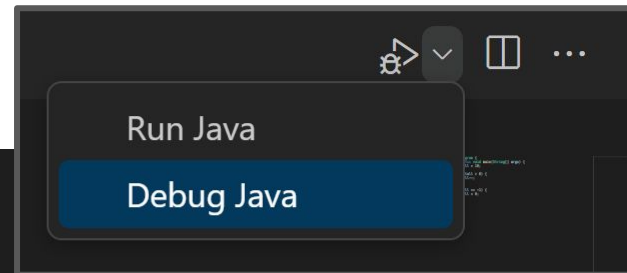
while(tall > 0) {
    System.out.println(x: "inne i while");
    tall--;
}

System.out.println(x: "kom ut");

if (tall == -1) {
    System.out.println(x: "hallooo");
    tall = 0;
}
```

- Kan alternativt bruke debugger-en i VSCode (Istedenfor å printe ut selv)

```
10
11     int tall = 10; tall = 3
12
13     while(tall > 0) { tall = 3
14         tall--; tall = 3
15     }
16
17     if (tall == -1) {
18         tall = 0;
19     }
```



# Jobb med oppgaver!

IN1010 Emnesiden → Grupper →  
Gruppe 4 → Uke8

