

# Lister og generiske klasser II

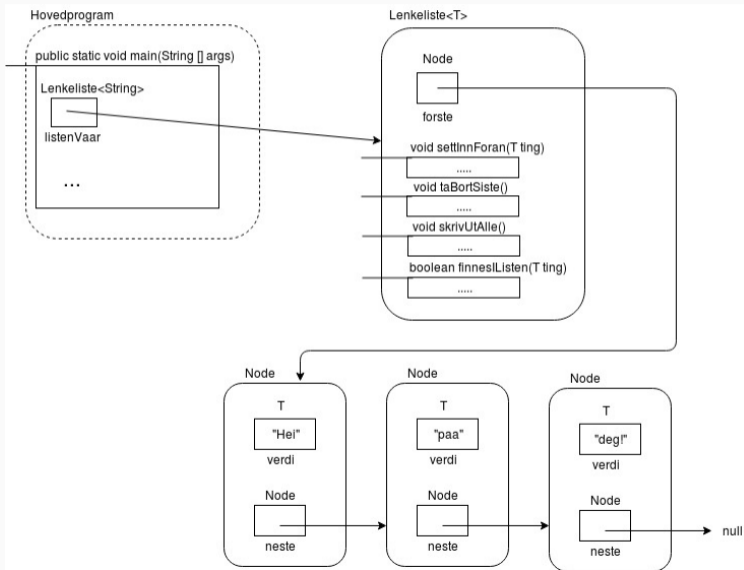
IN1010 - Uke 7

---

Tobias Paulsen

Vår 2021

- En beholder hvor noder peker på den neste og/eller forrige node i lista
- Hver node holder gjerne på data v noe slag
- Kan være enveis og toveis
- Man må ha minst en peker til en node for å ha tilgang til de andre
- Ofte den første noden i lista, men kan også være den siste eller begge
- FIFO - First In, First Out. Setter inn nye elementer bak
- LIFO - Last In, First Out. Setter inn nye elementer foran



# Comparable

- Er et interface for å sammenligne to objekter:

```
interface Comparable<T> {  
    int compareTo(T o);  
}
```

Hvis man skriver `a.compareTo(b)`;

- returnerer en positiv int om a er "større enn" b
- returnerer en negativ int om a er "mindre enn" b
- returnerer 0 om a og b er "like"

Eksempel: `Bankkonto.java`

# Iterator

- En iterator holder orden på hvor vi er, og gir oss elementene ett for ett

```
interface Iterator<T> {  
    boolean hasNext();  
    T next();  
}
```

- I og med at Iterator er et interface må vi lage en klasse som implementerer grensesnittet

# Iterable

- Hvis vi har en beholder vi ønsker skal være mulig å iterere over så må beholder-klassen implementere interfacet **Iterable**:

```
interface Iterable<T> {  
    Iterator<T> iterator();  
}
```

- Metoden **iterator()** returnerer ett objekt som implementerer **Iterator**-interfacet. Denne metoden blir kalt automatisk når man f.eks bruker **for-each**-løkker
- Skal ta eksempel på **Iterator** og **Iterable** på slutten av timen