niende seminar

in1010 - gruppe 2

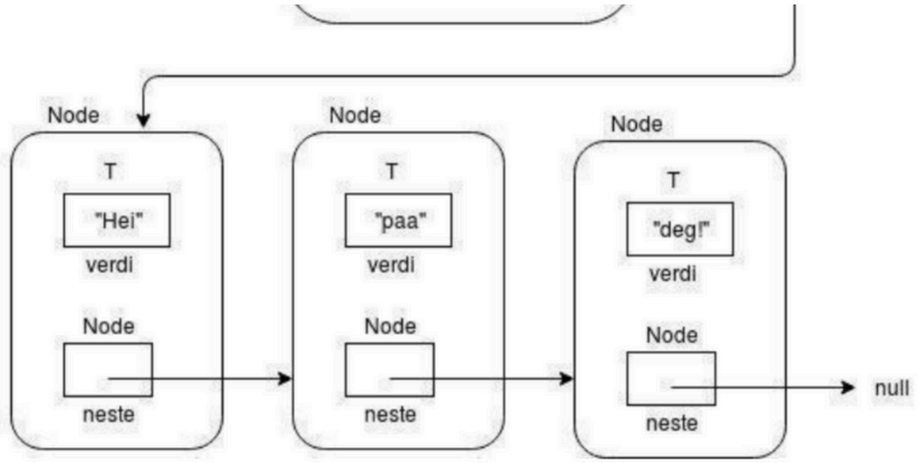
18 mars 2025

this week>

- Repetisjon av noder og lenkelister
- Generiske beholdere
- Seminaroppgaver

Noder

- En node er en beholder
- Den holder på data
- Ved å koble flere noder samme (med pekere), kan vi få trær eller lenkeliste

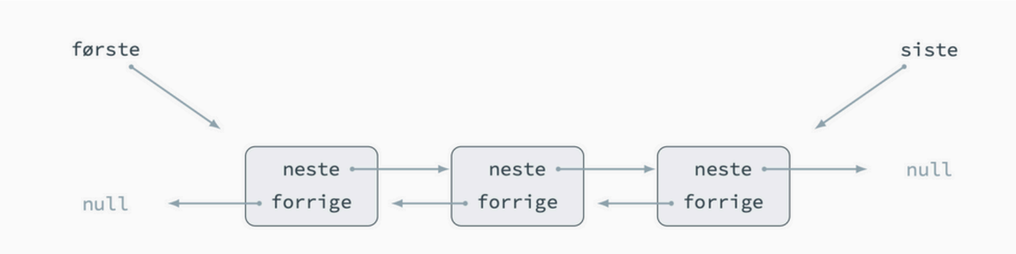


Enkeltlenket liste

- Første peker
- Siste peker
- Hver node har en peker til neste node
  - Ingen peker til noden før

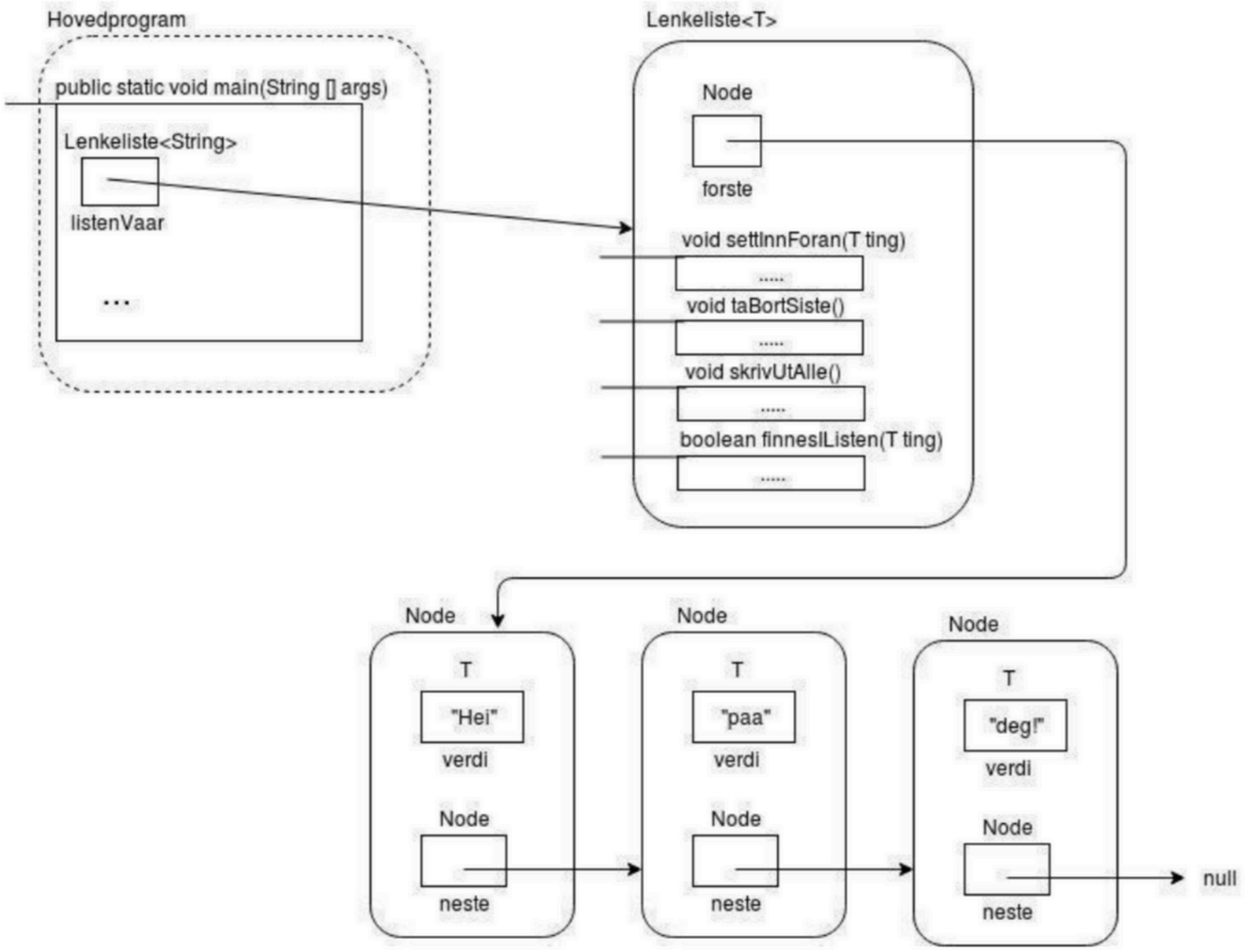
Dobbeltlenket liste

- Første peker
- Siste peker
- Hver node har en peker til neste node
  - Har peker til noden før
- Rask innsetting og sletting



Lenkeliste klassen

- Lenkeliste klassen har en indre node klasse
  - Referansen til nodene er gjennom lenkelisten
- Lenkelisteklassen håndterer innsetting, sletting og andre operasjoner



Generisk klasse

- Klassen er **generisk dersom** den tar inn én eller flere **typeparmetere** (typevariabler)
- Gjør at klassen kan **operere på spesifiserte** typer **uten å være** begrenset
  - T type til vilkårlig type
  - E element til samlinger, datastrukturer

Eksempel:

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>,
```

```
ArrayList<String> liste = new ArrayList<String>();
```

Med lenkelister kan vi implementere :

- Stabel (Last In First Out, LIFO)
- Kø (First In First Out)
- Prioritetskø (elementer settes og fjernes etter prioritet, f.eks høyest tall)

- Når man skal implementere, lurt å tegne når man skal sette inn, og fjerne elementer

Comparable grensesnittet

- Når vi har lyst til kan sammenligne objekter
  - Sortere objekter
- Må implementere compareTo()

```
public class Person implements Comparable<Person> {
    private int hoyde, alder;
    Person(int hoyde, int alder) {
    this.hoyde = hoyde;
    this.alder = alder;
    }
    @Override
    public int compareTo(Person annen) {
    int hoydeforskjell = this.hoyde - annen.hoyde;
    int aldersforskjell = this.alder - annen.alder;
    return hoydeforskjell == 0 ? aldersforskjell : hoydeforskjell;
    }
    }
```

Iterable grensesnittet

- Gjør at vi kan iterere/ gå gjennom listen med en for løkke
- Må implementere iterator(), og Iterable klassens metoder:hasNext() og next()

```java
import java.util.Iterator;
public class Lenkeliste<T> implements Iterable<T> {
    @Override
    public Iterator<T> iterator() {
        return new LenkelisteIterator();
    }
```

```java
class LenkelisteIterator implements Iterator<T> {
    Node denne = forste;

    @Override
    public boolean hasNext() {
        return denne != null;
    }

    @Override
    public T next() {
        T data = denne.data;
        denne = denne.neste;
        return data;
```

Anta at Person klassen implementerer comparable, og har en compareTo() metode som sammenligner på alder

- **Hva skjer hvis vi sorterer etter for-løkken?**
- **Hvorfor fungerer Array.sort(personer)?**

```
import java.util.Arrays;

class Testprogram {
    public static void main(String[] args) {
        Person[] personer = {
            new Person("Andreas", "Hurthe", 25),
            new Person("Kristine", "Henriksen", 30),
            new Person("Henrik", "Golsen", 25),
            new Person("Sigvart", "Dagsland", 29)
        };

        Arrays.sort(personer);

        for (Person p : personer) {
            System.out.println(p);
        }
    }
}
```

Seminaroppgaver