

Algoritmer og datastrukturer

Økt 6 – Beholdere, tabeller, lenkede lister

Beholder

```
public interface Beholder<T> extends Iterable<T> // foreløpig versjon
{
    public boolean leggInn(T verdi); // legger inn i beholderen
    public Iterator<T> iterator(); // returnerer en iterator

    // andre aktuelle metoder
}
```

```
for (T verdi : beholder) System.out.print(verdi + " ");
```

```
for (Iterator<T> i = beholder.iterator(); i.hasNext(); ) // for-løkke
{
    T verdi = i.next(); // henter neste i beholderen
    System.out.print(verdi + " "); // skriver ut
}
```

```
Iterator<T> i = beholder.iterator(); // henter iteratoren
```

```
while (i.hasNext()) // er det flere?
{
    System.out.print(i.next() + " "); // skriver ut neste
}
```

Tabelliste - iteratorer

10	7	14	2	10	8	3	7	13	9	6					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
// Skal ligge som en indre klasse i class TabellListe
private class TabellListeIterator implements Iterator<T>
{
    private int denne = 0;          // instansvariabel

    public boolean hasNext()        // sjekker om det er flere igjen
    {
        return denne < antall;      // sjekker verdien til denne
    }

    public T next()                 // returnerer aktuell verdi
    {
        if (!hasNext())
            throw new NoSuchElementException("Tomt eller ingen verdier igjen!");
        return a[denne++]; // a[denne] returneres før denne++
    }
} // TabellListeIterator
```

Iteratorer – fjerning av elementer

```
Iterator<String> i = liste.iterator();    // oppretter en iterator
System.out.println(i.next());             // den første i listen

liste.fjern("Per");                       // fjerner den første
System.out.println(i.next());             // den neste i listen
```

Iteratorer – fjerning / endring av tabell

```
public class TabellListe<T> implements Liste<T>
{
    private T[] a;
    private int antall;
    private int endringer;    // ny variabel

    // øvrige ting
}

private class TabellListeIterator implements Iterator<T>
{
    private int denne = 0;
    private boolean fjernOK = false;
    private int iteratorendringer = endringer;    // ny variabel

    // øvrige ting
}
```

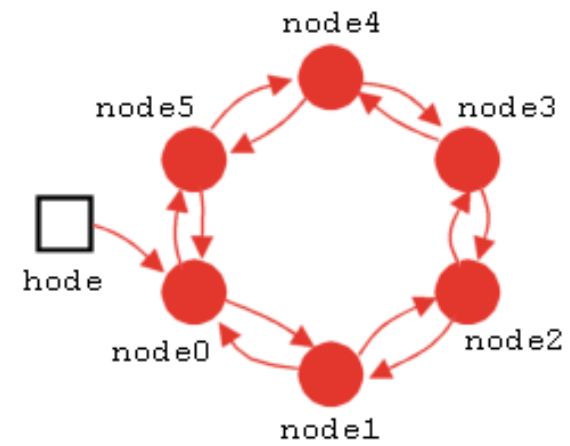
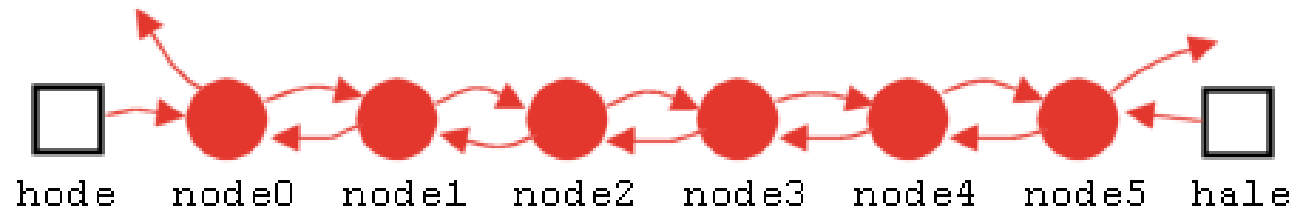
Iteratorer – fjerning / endring av tabell

```
public T next()
{
    if (iteratorendringer != endringer)
    {
        throw new ConcurrentModificationException("Listen er endret!");
    }

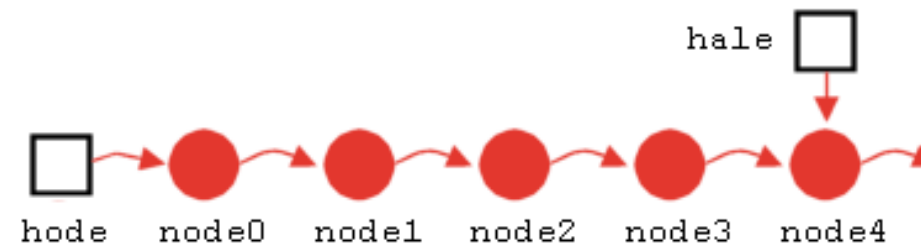
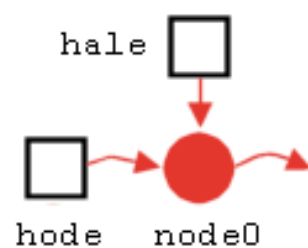
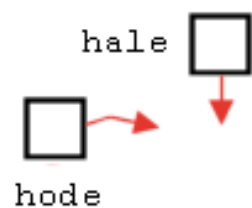
    if (!hasNext())
    {
        throw new NoSuchElementException("Tomt eller ingen verdier igjen!");
    }

    T denneVerdi = a[denne];    // henter aktuell verdi
    denne++;                    // flytter indeksen
    fjernOK = true;             // nå kan remove() kalles
    return denneVerdi;          // returnerer verdien
}
```

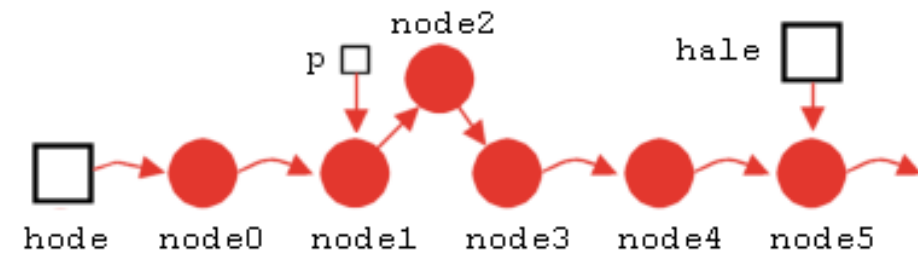
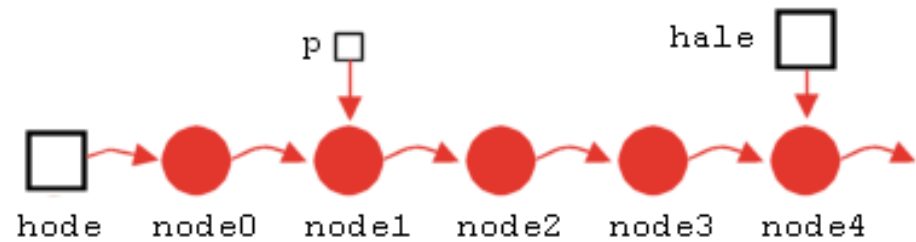
Lenket liste



Hode og hale-peker



Legge inn på gitt posisjon i listen



Fjerne fra listen

