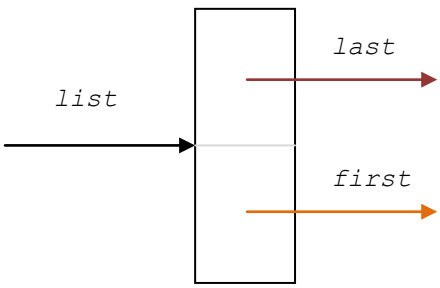
En länkad lista

En tom lista

```
class List
{
    private static class Node
    {
        public int      value;
        public Node     next;

        public Node (int value, Node next)
        {
            this.value = value;
            this.next = next;
        }
    }

    public Node     first;
    public Node     last;

    public List ()
    {
        first = null;
        last = null;
    }
}
```

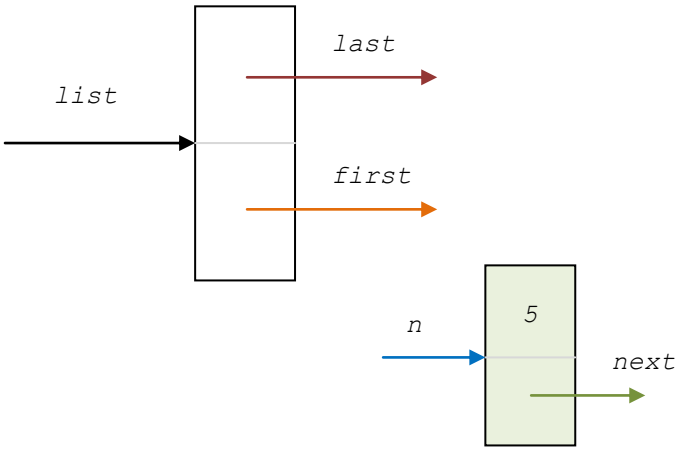
```
List    list = new List ();
```

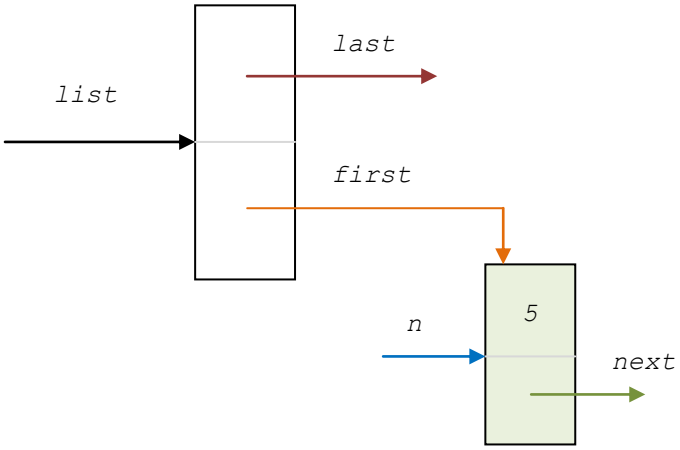


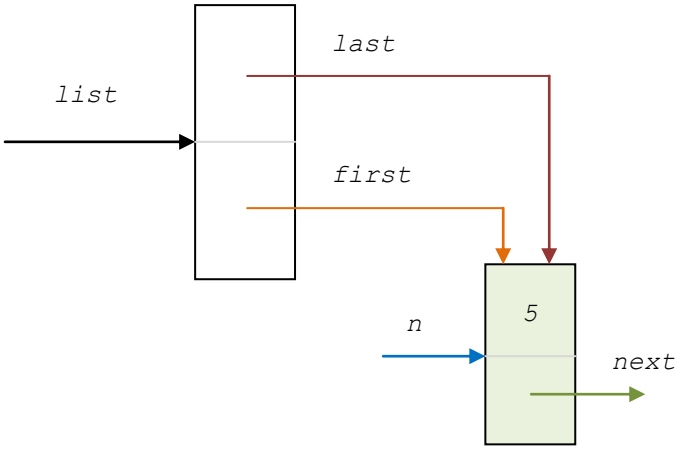
Lägg till element till listan

```
class List
{
    private static class Node
    {
        public int      value;
        public Node     next;

        public Node (int value, Node next)
        {
            this.value = value;
            this.next = next;
        }
    }

    public Node     first;
    public Node     last;

    public List ()
    {
        first = null;
        last = null;
    }

    public void add (int value)
    {
        Node     n = new Node (value, null);

        if (first == null)
            first = n;
        else
            last.next = n;
        last = n;
    }
}
```

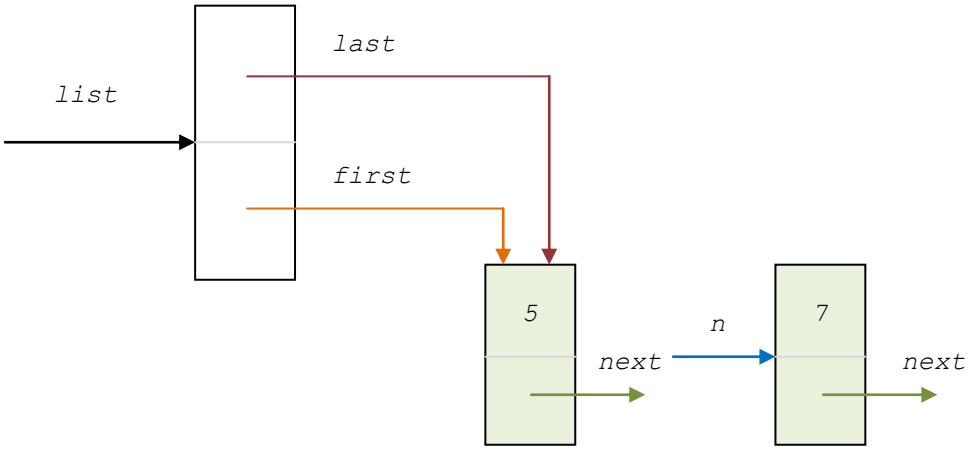
```
List    list = new List ();
list.add (5);
```

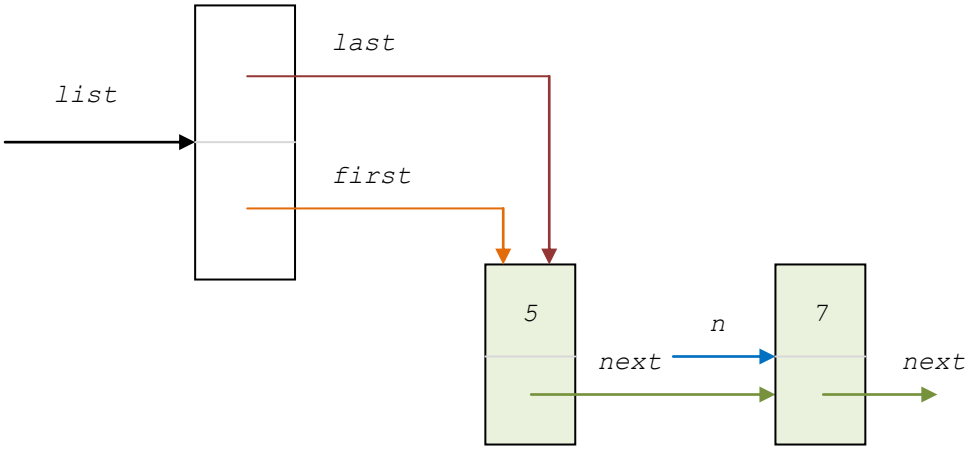


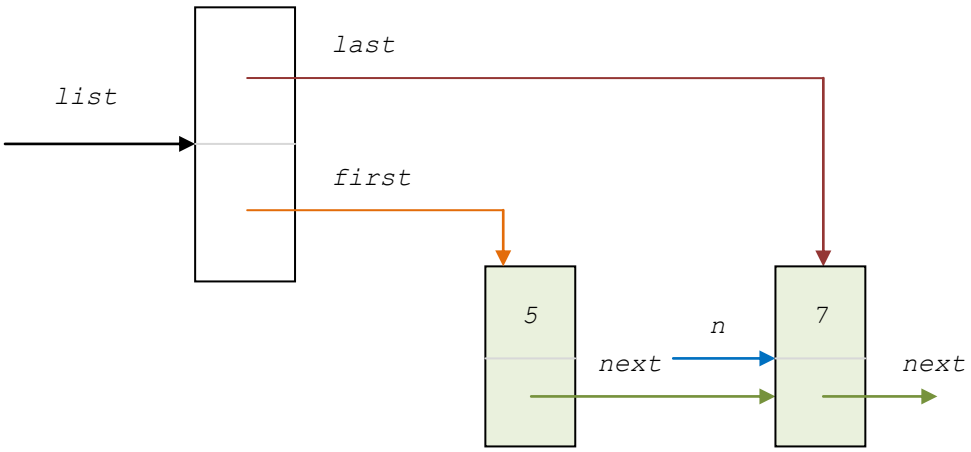




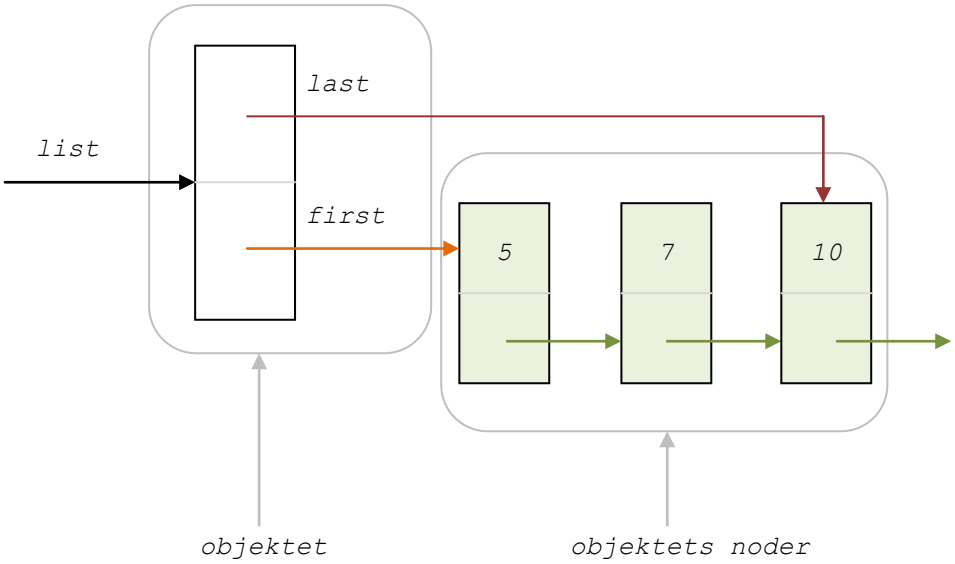
```
list.add (7);
```


last
list
first
5
n
7
next
next


last
list
first
5
n
7
next
next


last
list
first
5
n
7
next
next

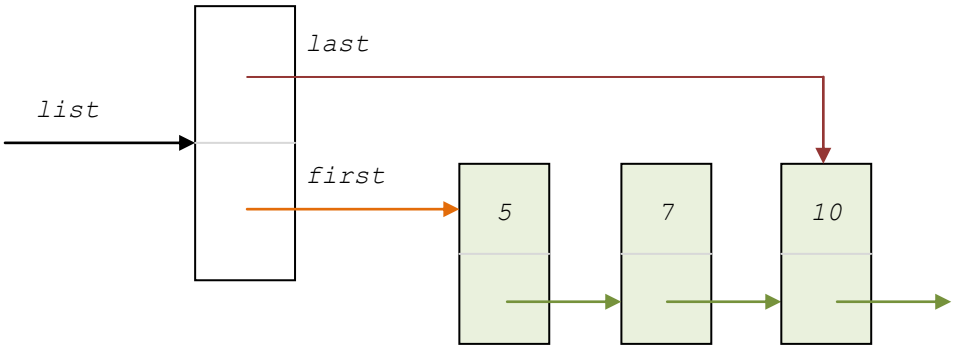
```
list.add (10);
```



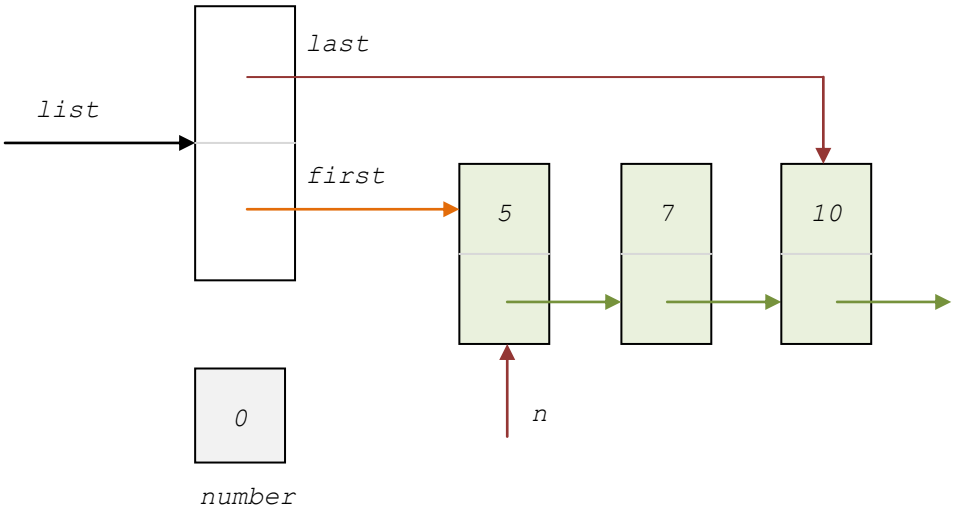
Antalet element i listan

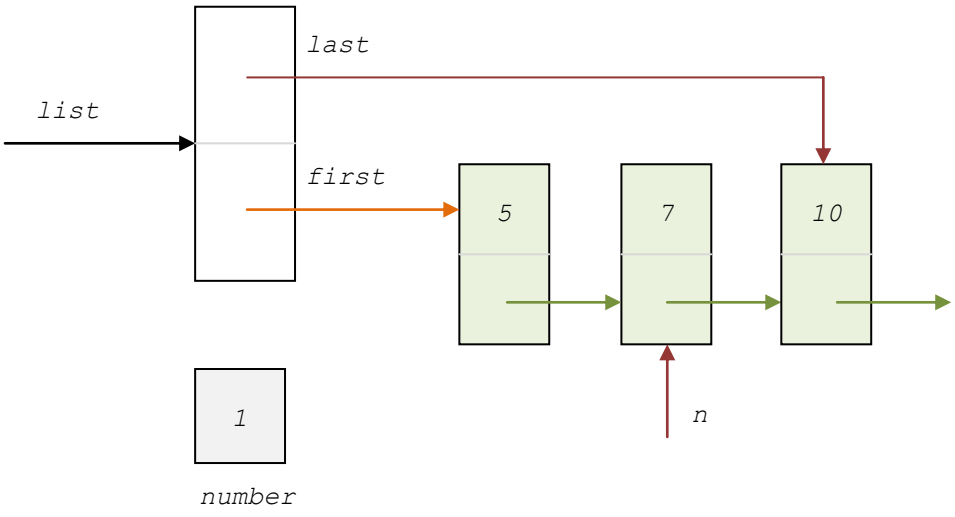
```
class List
{
    private static class Node
    {
        public int      value;
        public Node     next;

        public Node (int value, Node next)
        {
            this.value = value;
            this.next = next;
        }
    }

    public Node     first;
    public Node     last;

    public List ()
    {
        first = null;
        last = null;
    }

    public void add (int value)
    {
        Node     n = new Node (value, null);

        if (first == null)
            first = n;
        else
            last.next = n;
        last = n;
    }

    public int size ()
    {
        int     number = 0;
        Node     n = first;
        while (n != null)
        {
            number++;
            n = n.next;
        }

        return number;
    }
}
```

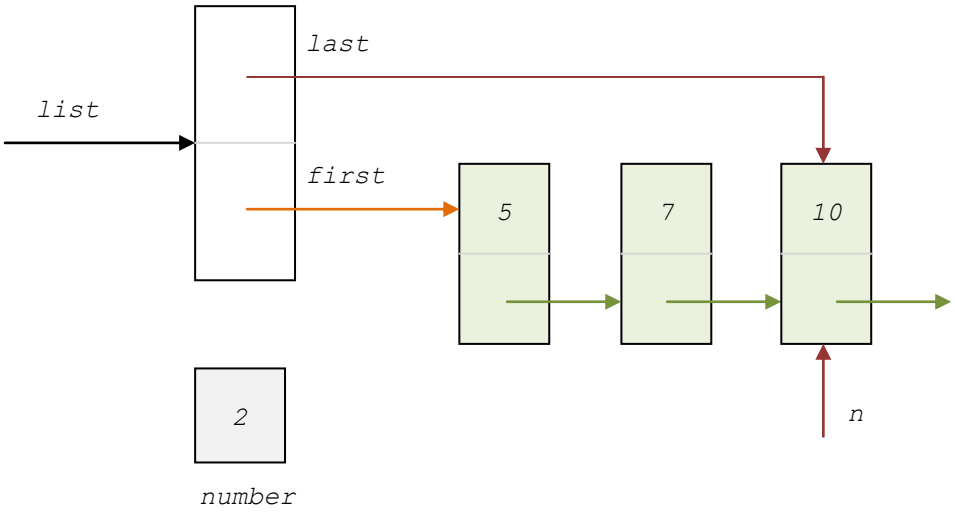
```
List    list = new List ();
list.add (5);
list.add (7);
list.add (10);
```

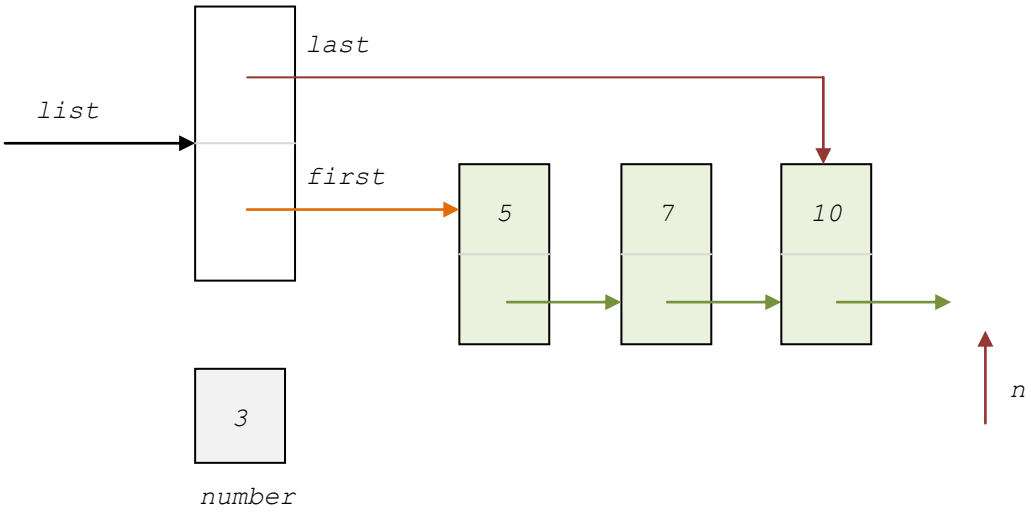


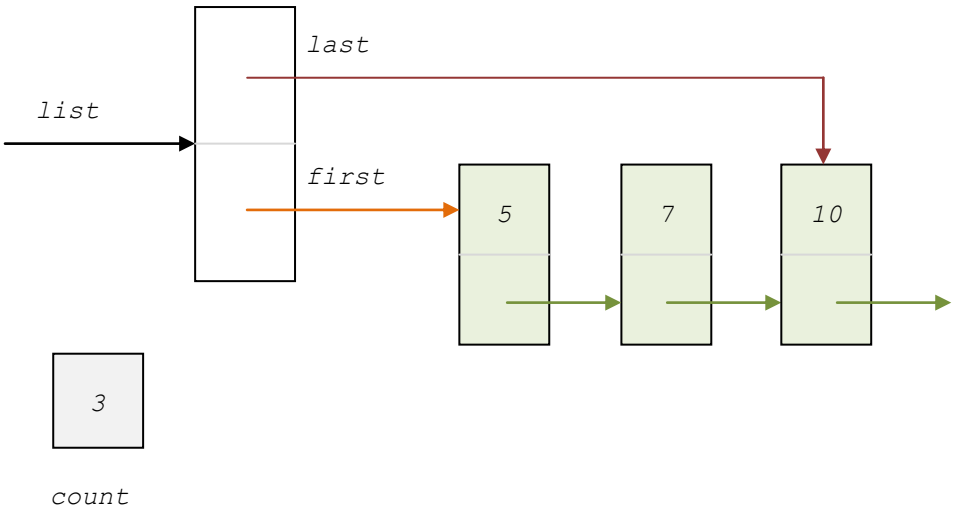

```
int    count = list.size ();
```






last
list
first
5
7
10
2
n
number

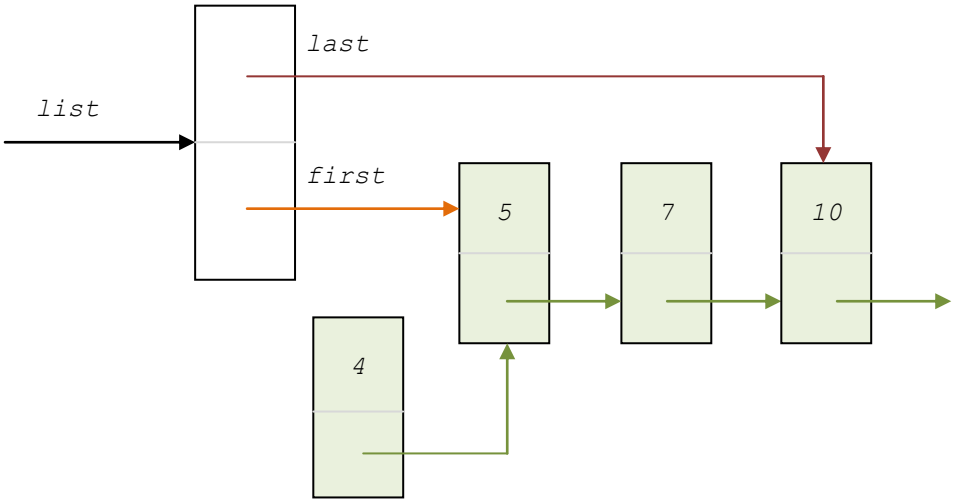

last
list
first
5
7
10
3
n
number

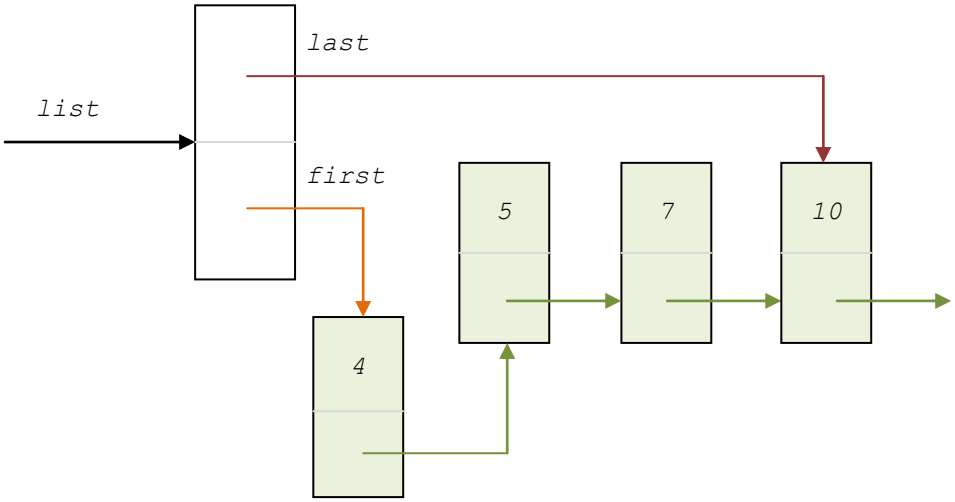

last
list
first
5
7
10
3
count

Sätt in ett element i listan

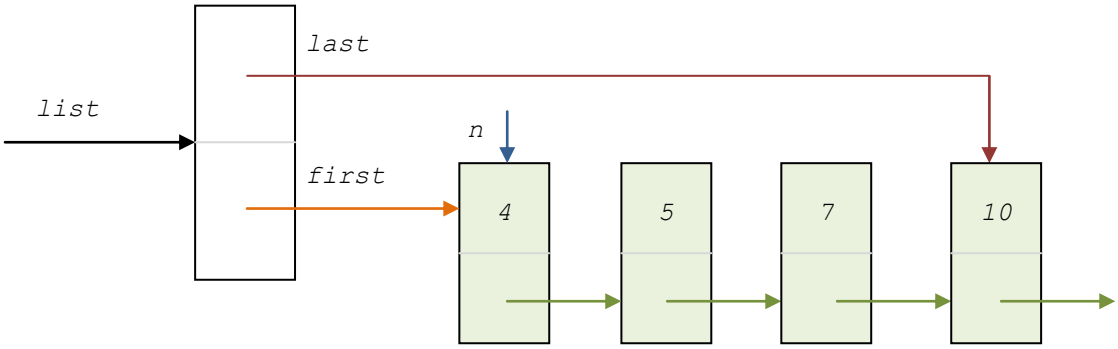
```
class List
{
    private static class Node
    {
        public int      value;
        public Node     next;

        public Node (int value, Node next)
        {
            this.value = value;
            this.next = next;
        }
    }

    public Node     first;
    public Node     last;

    public List ()
    {
        first = null;
        last = null;
    }

    public void add (int value)
    {
        Node     n = new Node (value, null);

        if (first == null)
            first = n;
        else
            last.next = n;
        last = n;
    }

    public void insert (int index, int value)
    {
        if (index == 0)
            first = new Node (value, first);
        else
        {
            Node     n = first;
            for (int i = 1; i < index; i++)
                n = n.next;
            n.next = new Node (value, n.next);
        }
    }
}
```

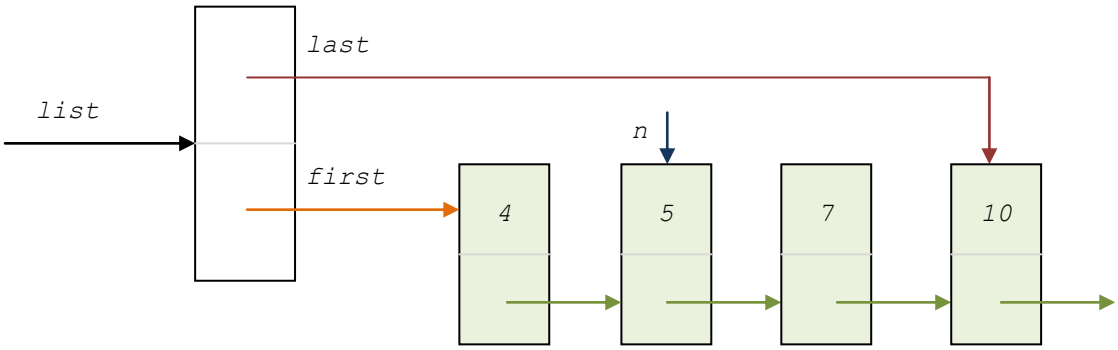
```
list.insert (0, 4);
```

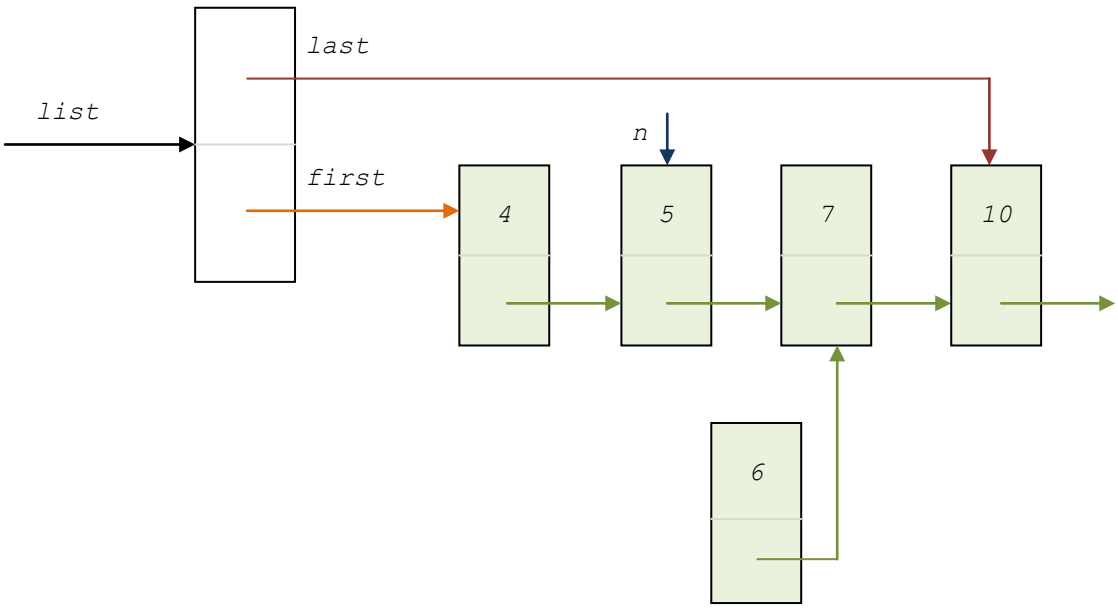


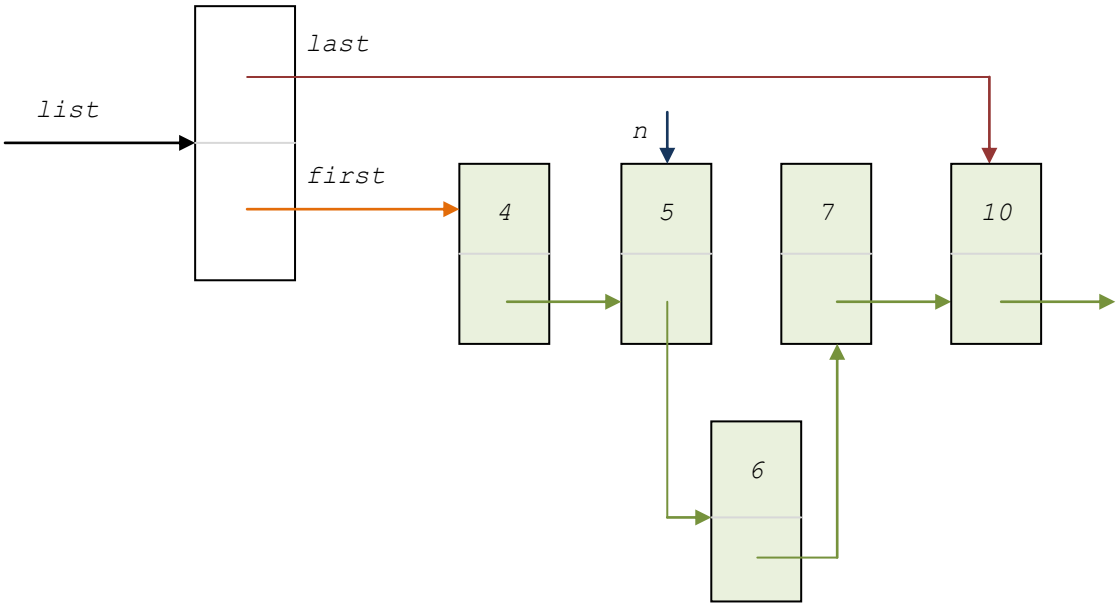


```
list.insert (2, 6);
```





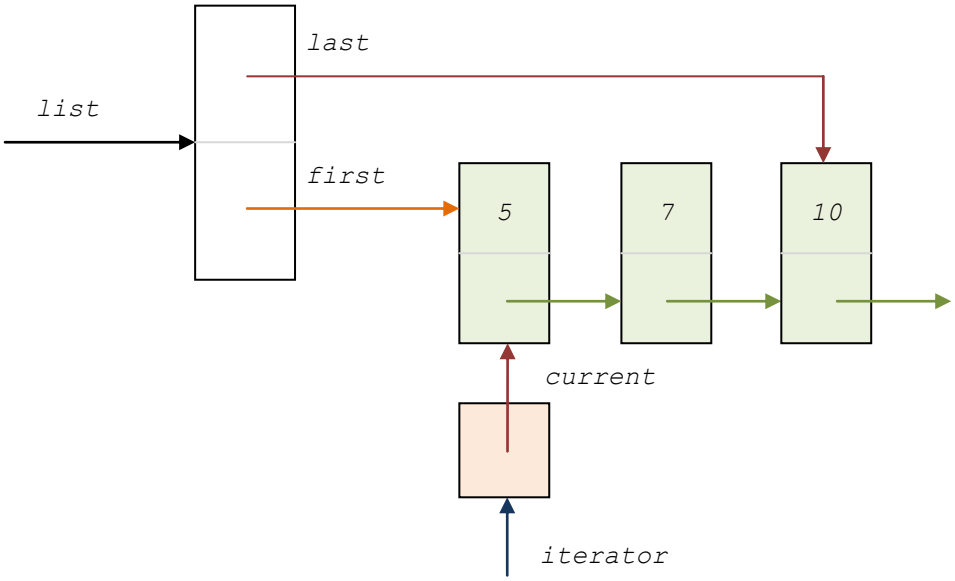



last
list
n
first
4
5
7
10
6

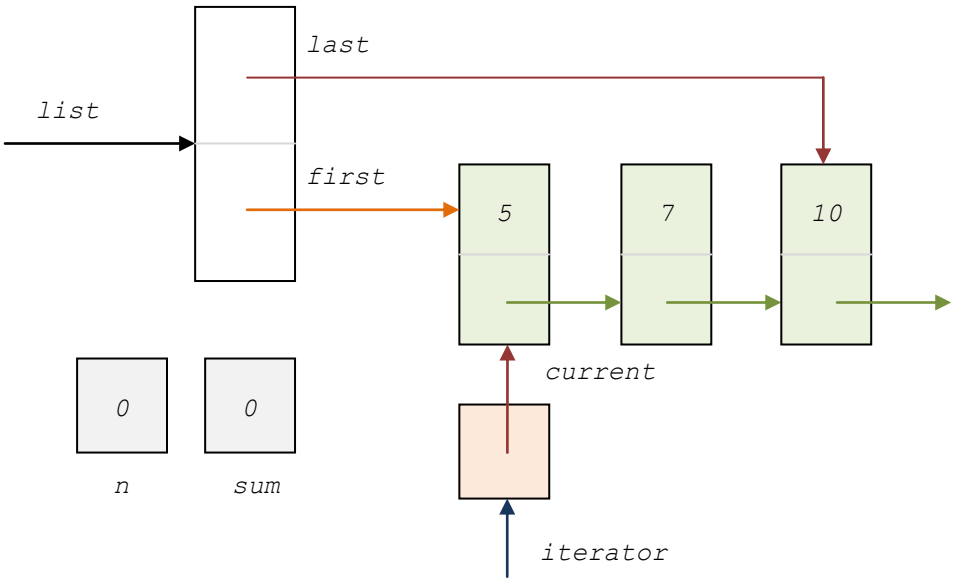
En iterator till listan

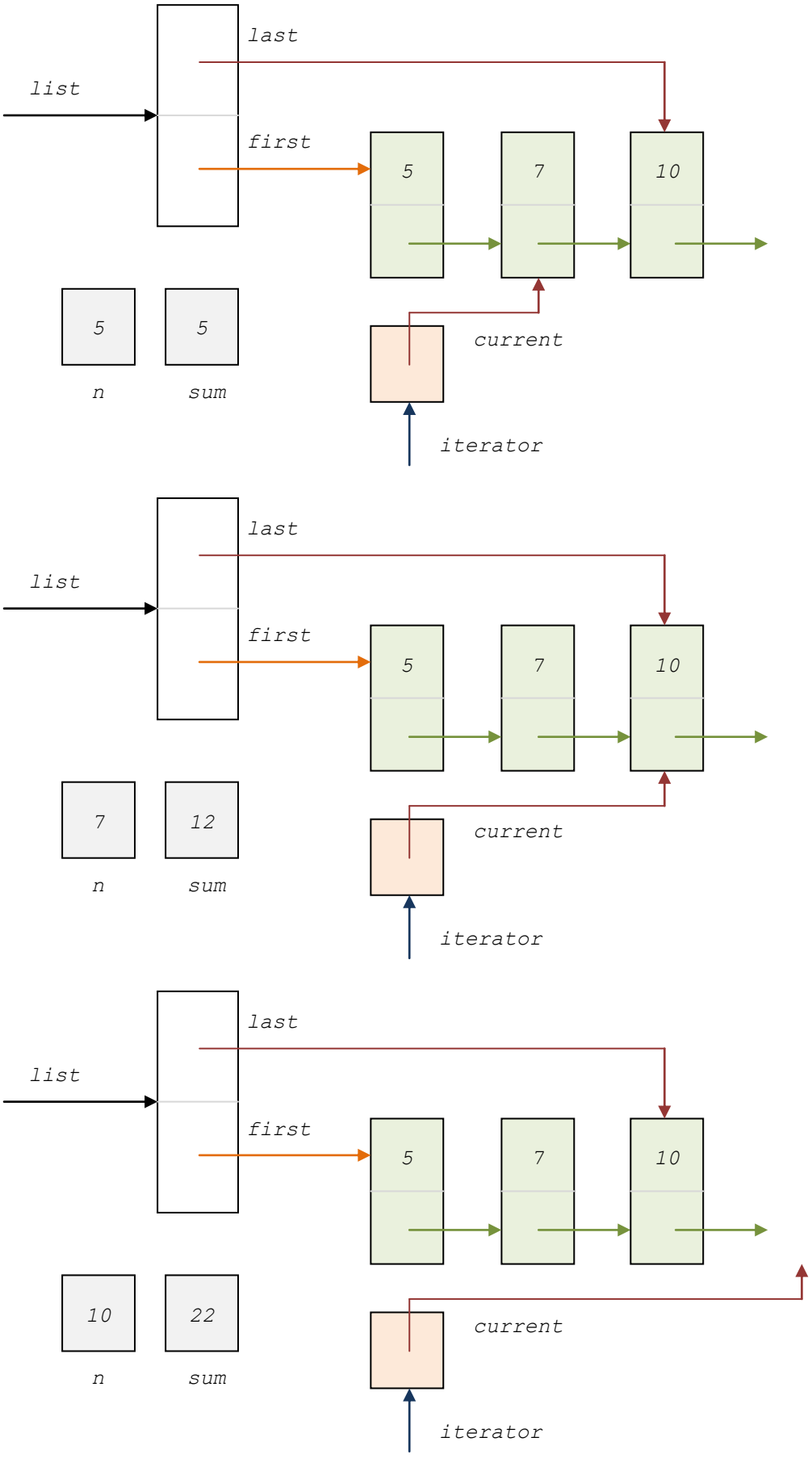
```
class List
{
    // klassen Node
    // instansvariabler
    // konstruktorer
    // metoder

    public class ListIterator
    {
        private Node    current = null;

        public ListIterator ()
        {
            current = first;
        }

        public boolean hasElement ()
        {
            return current != null;
        }

        public int element ()
           throws java.util.NoSuchElementException
        {
            if (!this.hasElement ()) throw new
                java.util.NoSuchElementException (
                            "end of the iteration");

            int    element = current.value;

            return element;
        }

        public void move ()
        {
            current = current.next;
        }
    }
}
```

```
List    list = new List ();
list.add (5);
list.add (7);
list.add (10);
List.ListIterator    iterator = list.new ListIterator ();
```




```
int    n = 0;
int    sum = 0;
while (iterator.hasElement ())
{
    n = iterator.element ();
    sum += n;

    iterator.move ();
};
```




last
list
first
5
7
10
5
5
n
sum
current
iterator
last
list
first
5
7
10
7
12
n
sum
current
iterator
last
list
first
5
7
10
10
22
n
sum
current
iterator