Algoritmi in podatkovne strukture 1

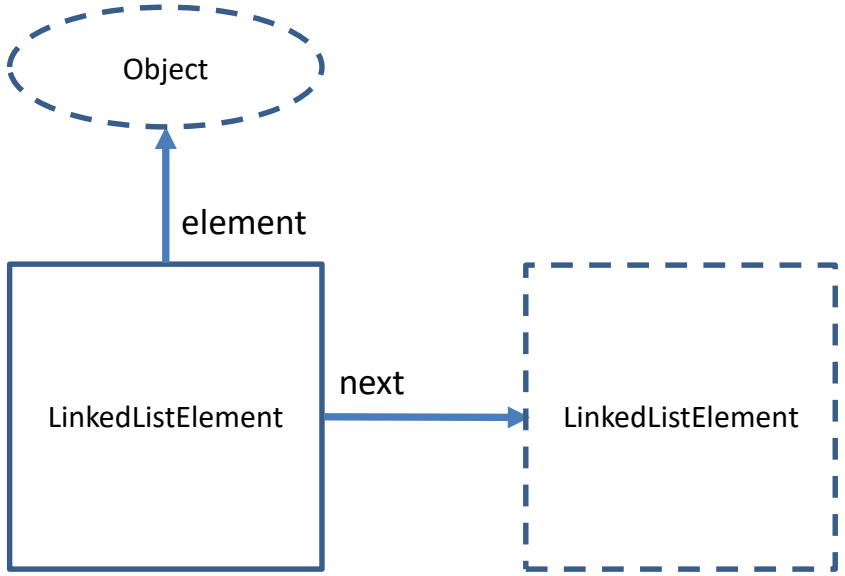
3. laboratorijske vaje

Linearni seznam s kazalci

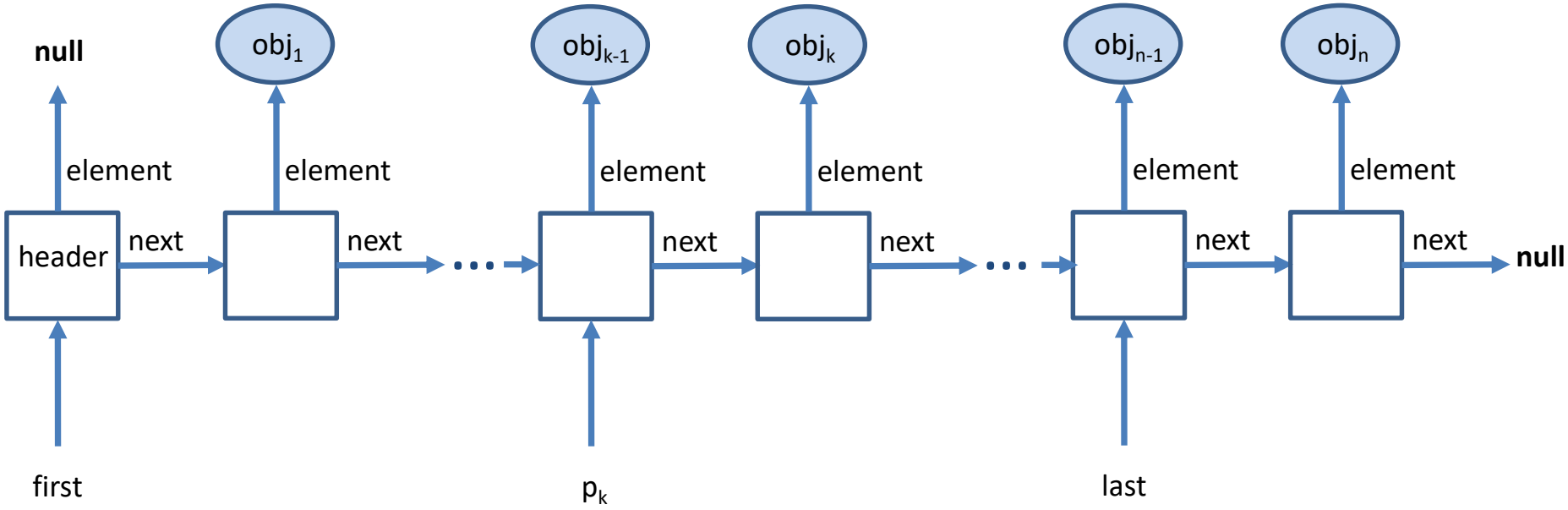
LINEARNI SEZNAM S KAZALCI

```
class LinkedListElement
{
    Object element;
    LinkedListElement next;
    …
}
```

```
class LinkedList
{
    LinkedListElement first;
    LinkedListElement last;
    …
}
```

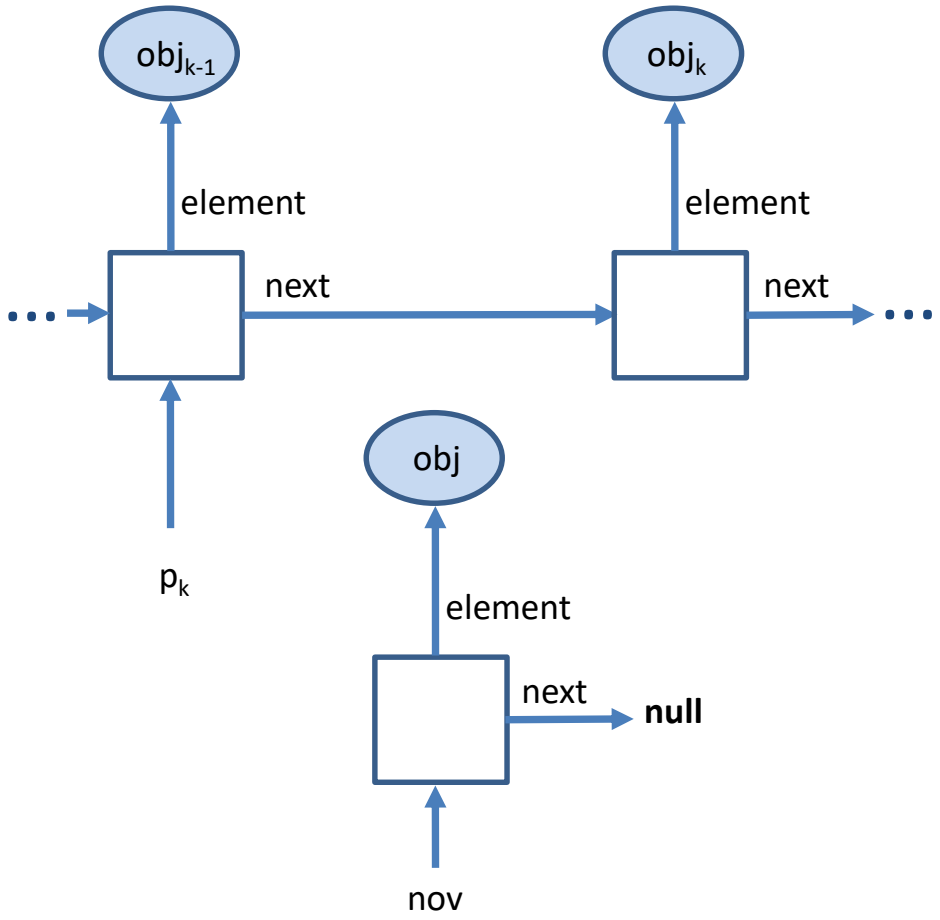


LINEARNI SEZNAM S KAZALCI



LINEARNI SEZNAM S KAZALCI

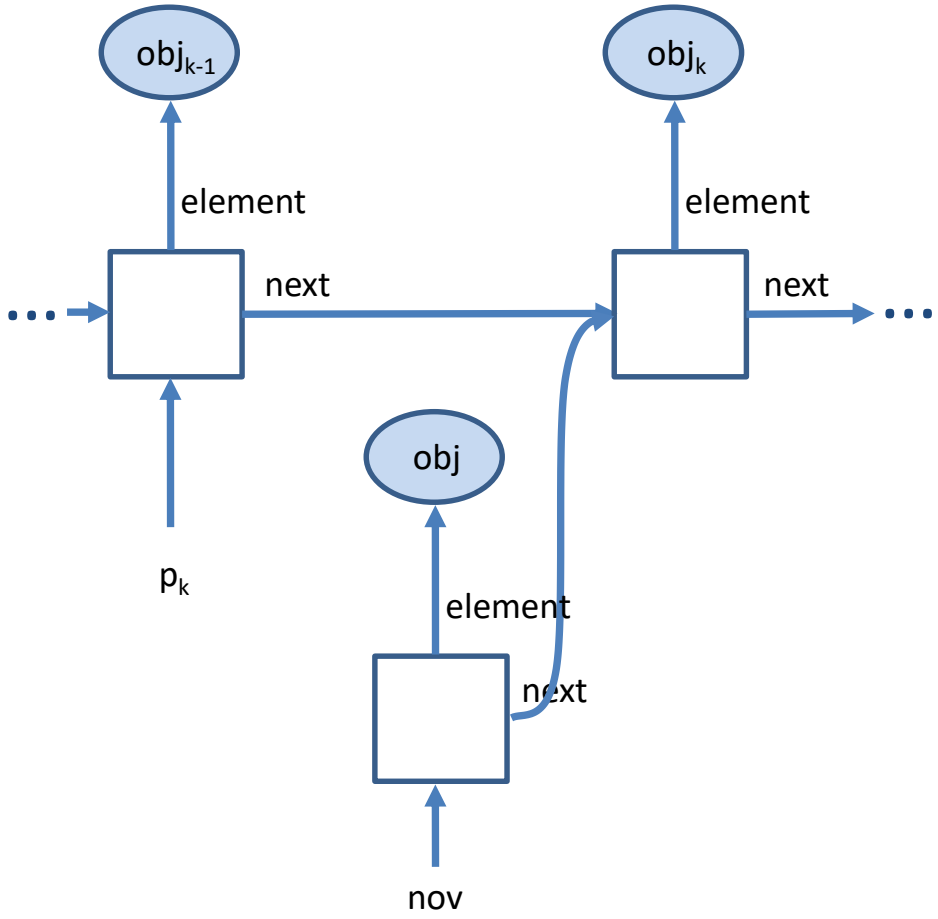
Vstavljanje novega elementa na k-to mesto v seznamu




```
LinkedListElement nov = new LinkedListElement(obj);
pk = …
```

LINEARNI SEZNAM S KAZALCI

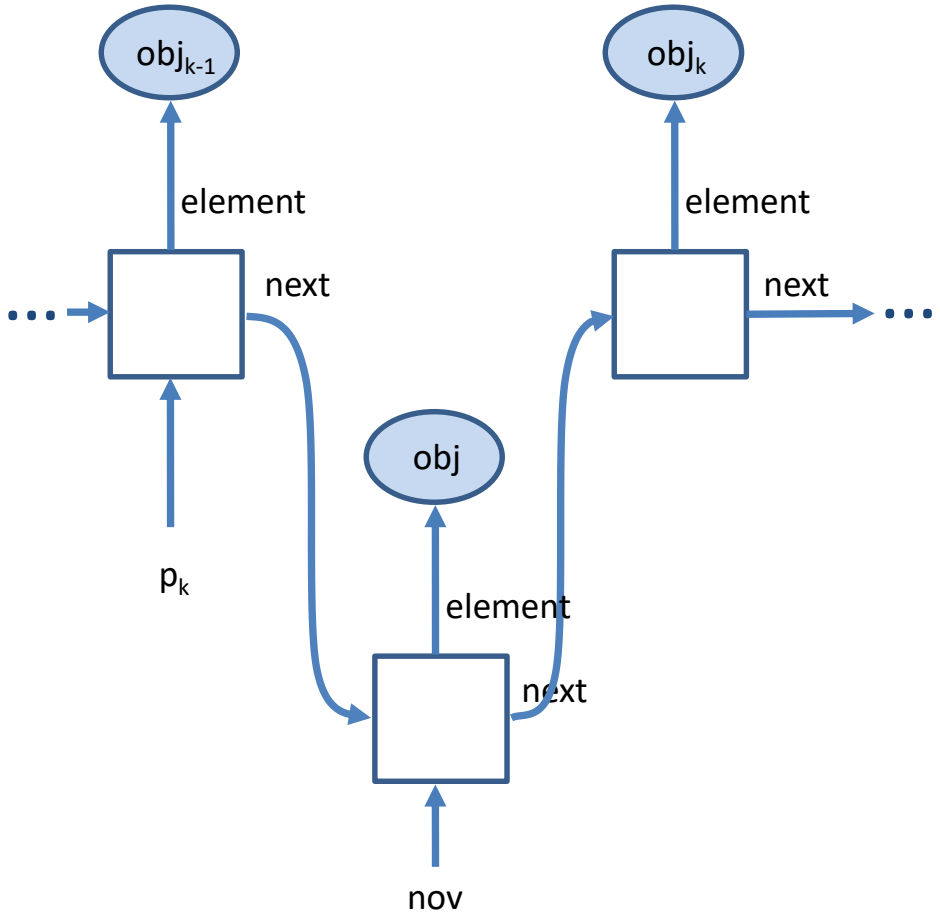
Vstavljanje novega elementa na k-to mesto v seznamu




```
LinkedListElement nov = new LinkedListElement(obj);
pk = …
nov.next = pk.next;
```

LINEARNI SEZNAM S KAZALCI

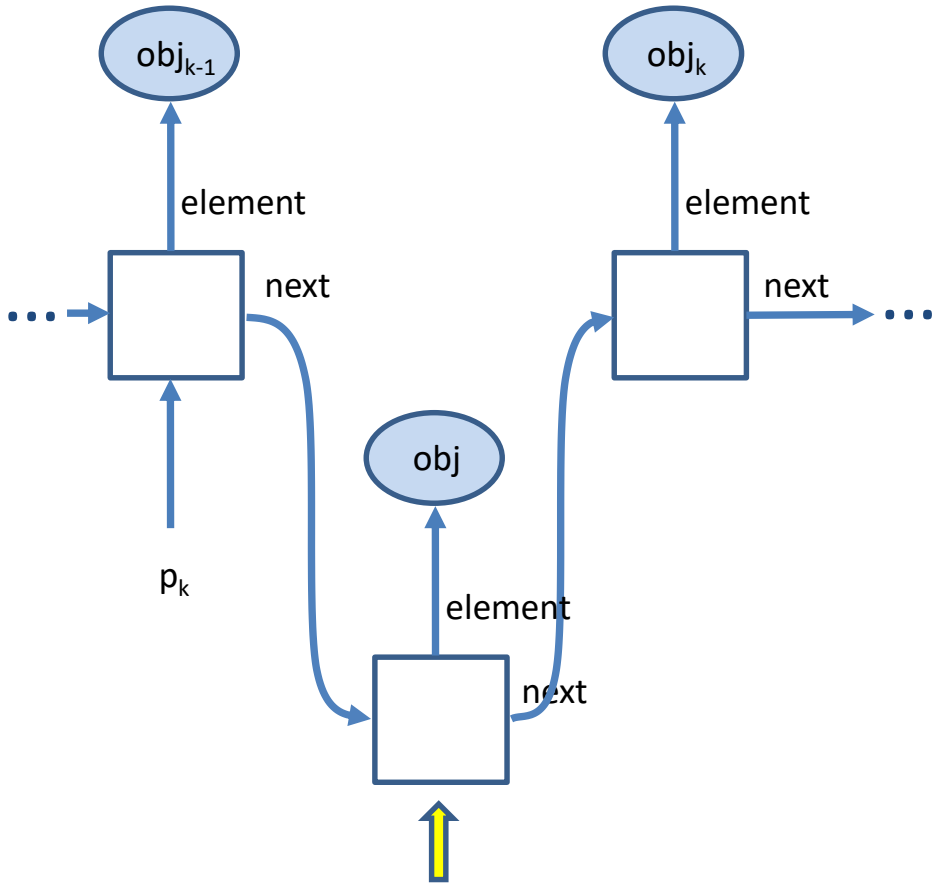
Vstavljanje novega elementa na k-to mesto v seznamu




```
LinkedListElement nov = new LinkedListElement(obj);
pk = …
nov.next = pk.next;
pk.next = nov;
```

LINEARNI SEZNAM S KAZALCI

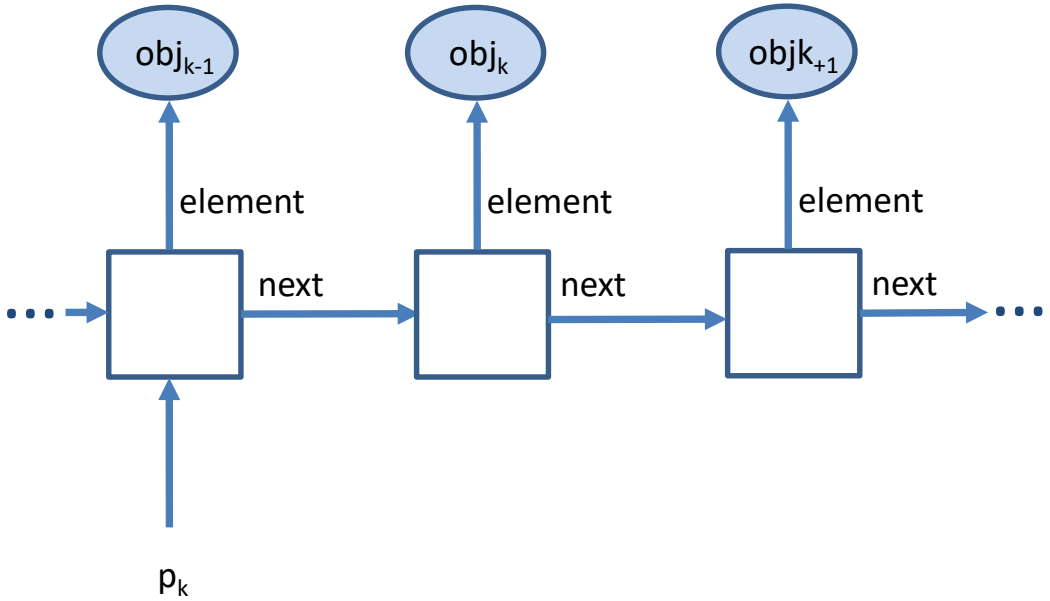
Vstavljanje novega elementa na k-to mesto v seznamu




```
LinkedListElement nov = new LinkedListElement(obj);
pk = …
nov.next = pk.next;
pk.next = nov;
```

LINEARNI SEZNAM S KAZALCI

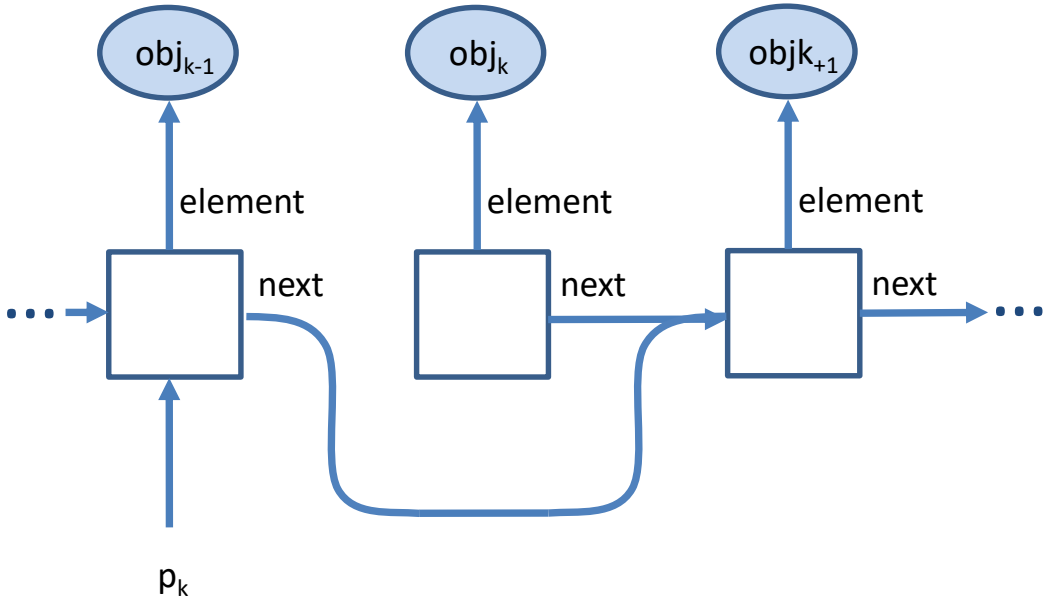
Brisanje k-tega elementa iz seznama



pk = …

LINEARNI SEZNAM S KAZALCI

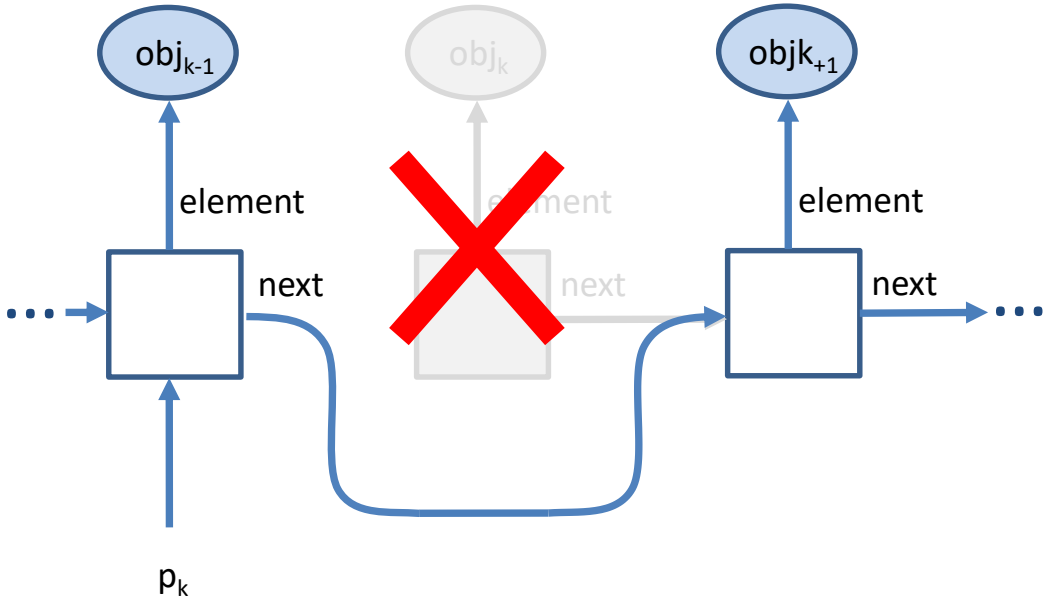
Brisanje k-tega elementa iz seznama




```
pk = …
pk.next = pk.next.next;
```

LINEARNI SEZNAM S KAZALCI

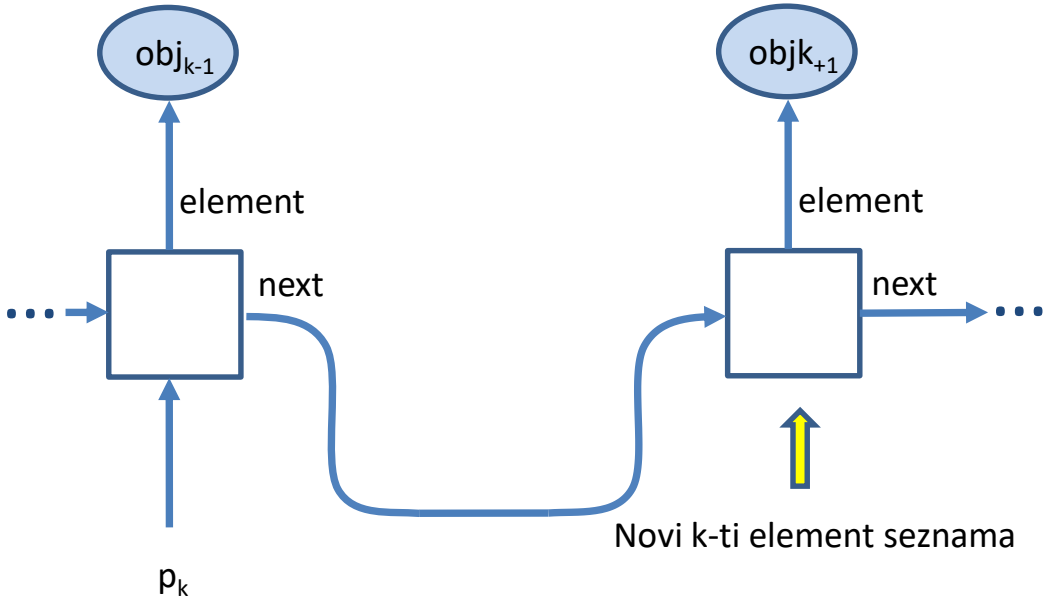
Brisanje k-tega elementa iz seznama




```
pk = …
pk.next = pk.next.next;
```

LINEARNI SEZNAM S KAZALCI

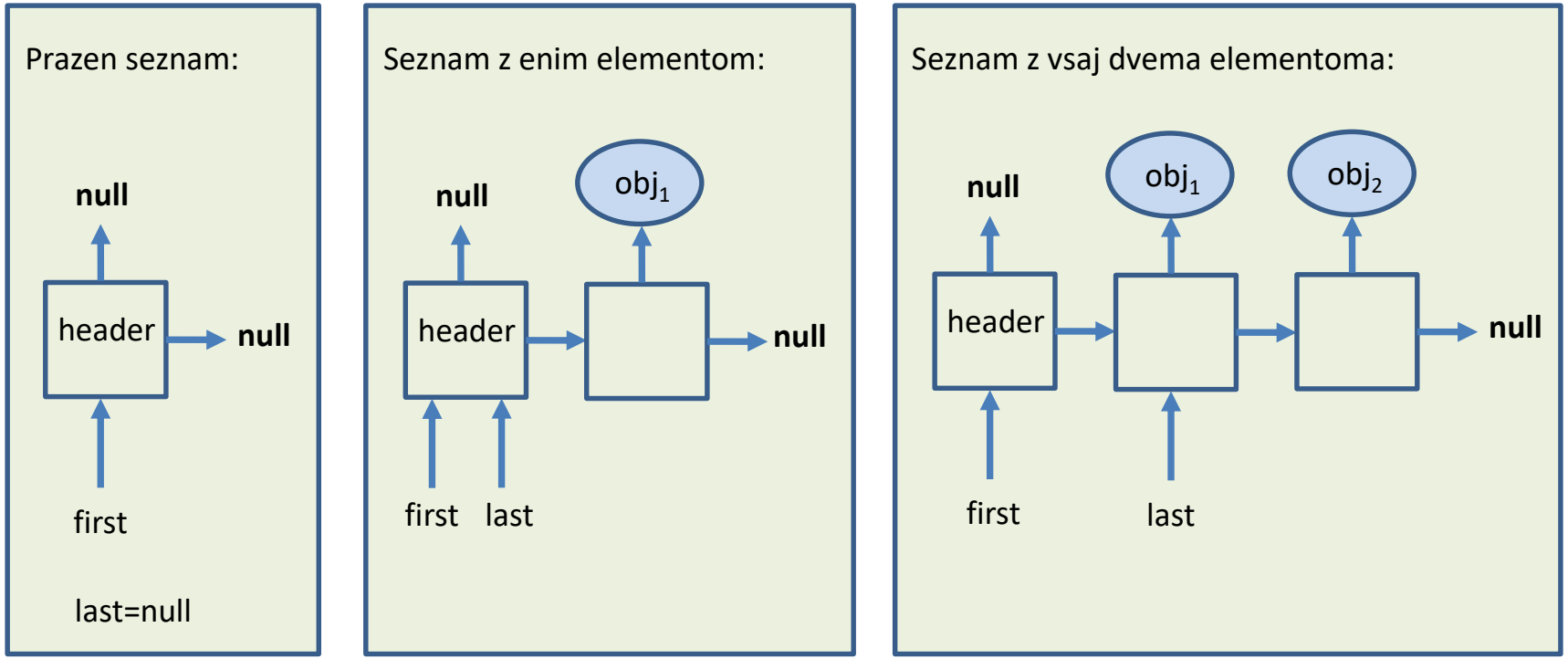
Brisanje k-tega elementa iz seznama




```
pk = …
pk.next = pk.next.next;
```

LINEARNI SEZNAM S KAZALCI

Posebni primeri



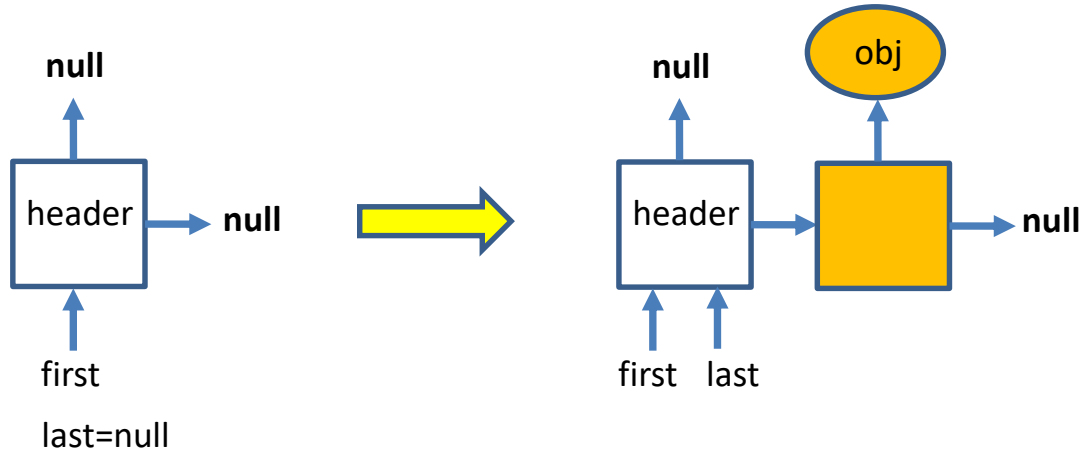
NALOGE

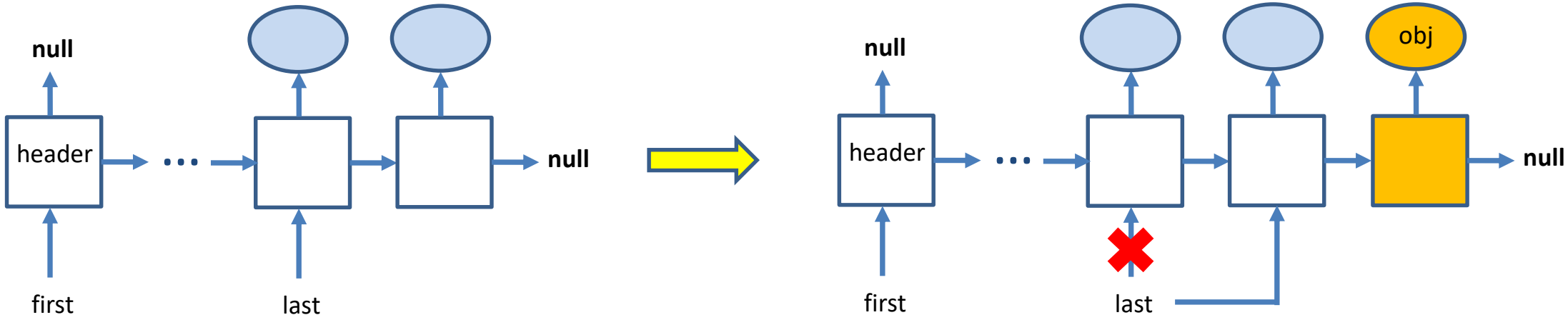
Implementirajte naslednje metode v razredu LinkedList:

- `void addLast(Object obj)` – doda element na konec seznama
- `void addFirst(Object obj)` – doda element na začetek seznama
- `boolean insertNth(Object obj, int n)` - vstavi element na n-to mesto v seznamu
- `boolean deleteNth(int n)` - izbriše element na n-tem mestu v seznamu
- `int length()` - vrne dolzino seznama (pri tem ne upošteva glave seznama)
- `int lengthRek()` - kliče rekurzivno funkcijo za izračun dolžine seznama
- `void reverse()` - obrne vrstni red elementov v seznamu (pri tem ignorira glavo seznama)
- `void reverseRek()` - kliče rekurzivno funkcijo, ki obrne vrstni red elementov v seznamu
- `void removeDuplicates()` - odstrani ponavljajoče se elemente v seznamu
- `void write()` - izpise elemente seznama

LINEARNI SEZNAM S KAZALCI

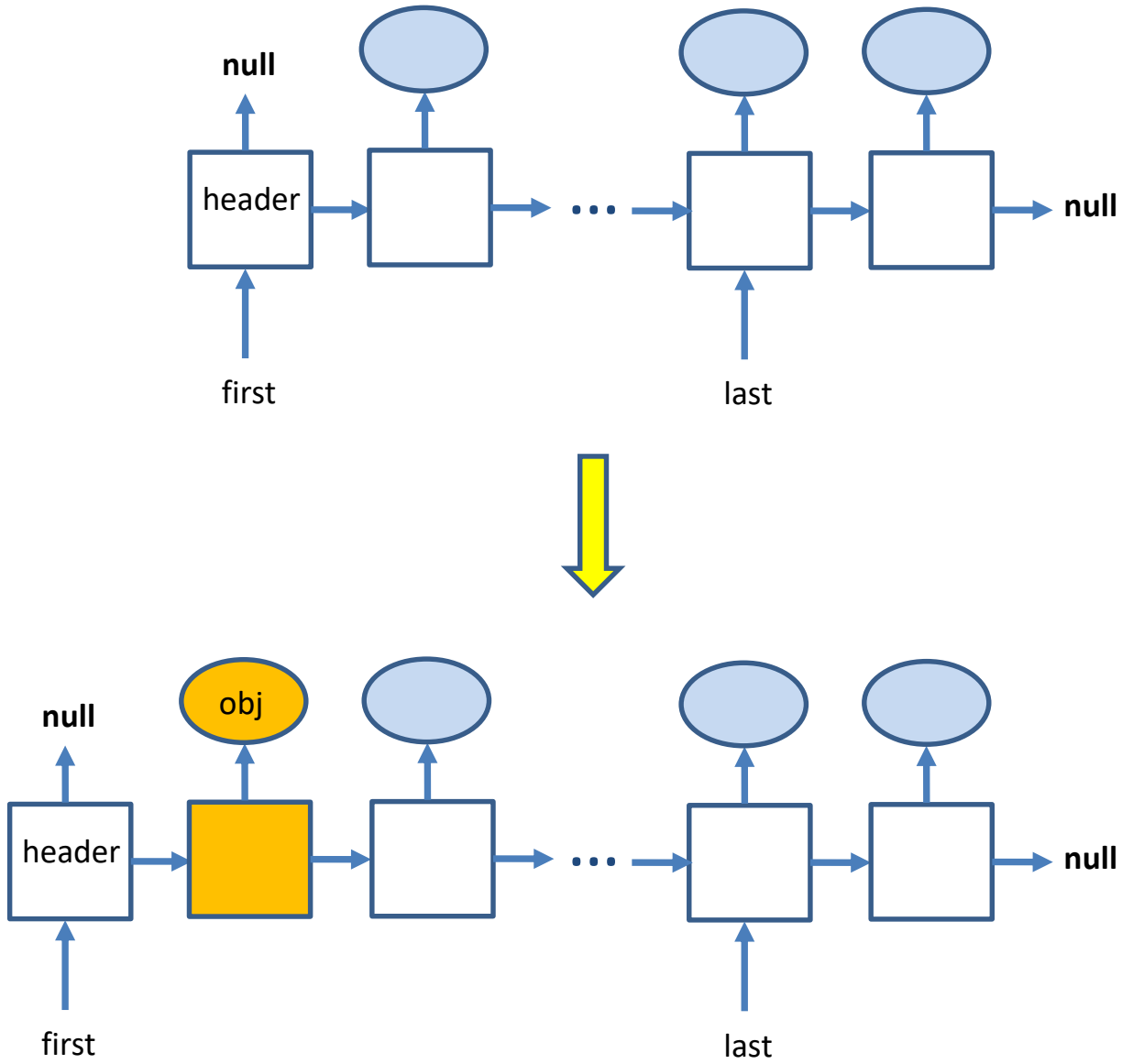
addLast(Object obj)





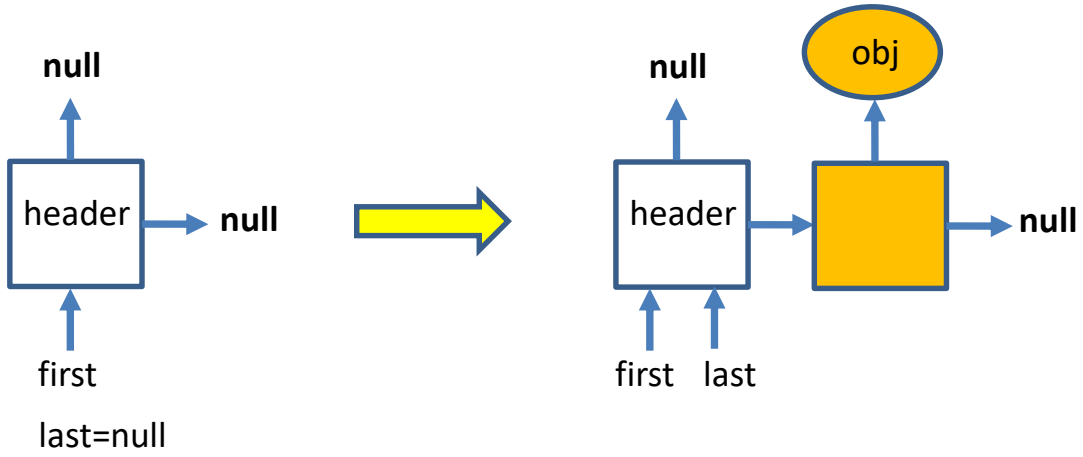
LINEARNI SEZNAM S KAZALCI

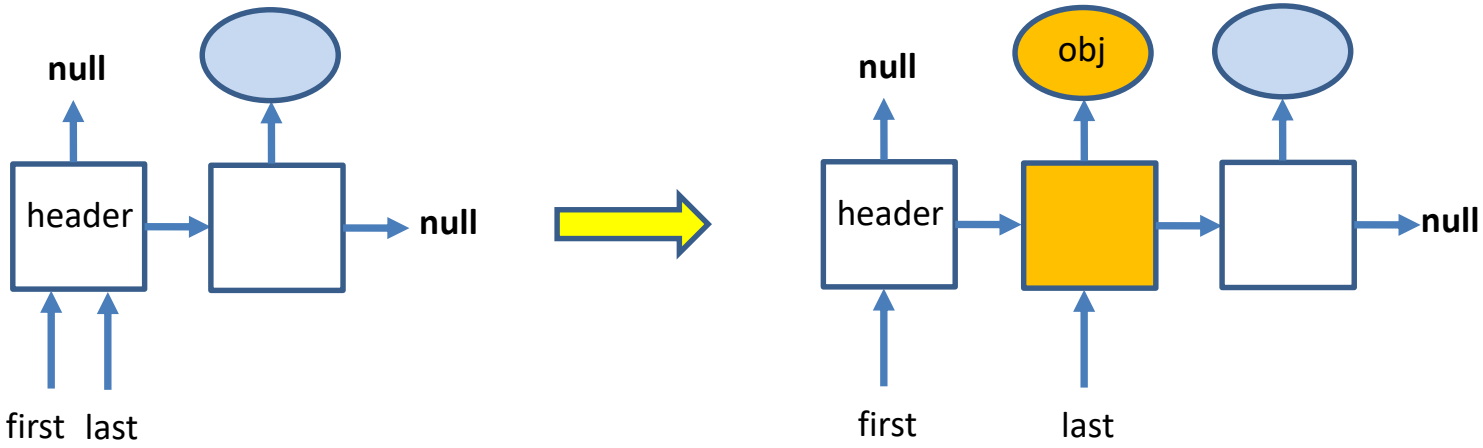
addFirst(Object obj)



LINEARNI SEZNAM S KAZALCI

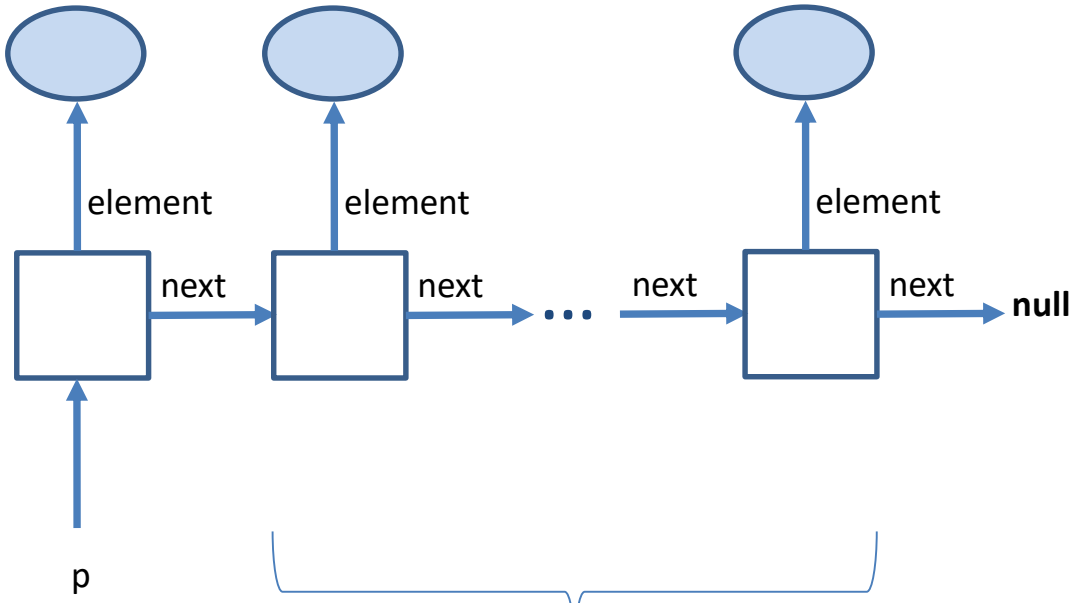
addFirst(Object obj)





LINEARNI SEZNAM S KAZALCI

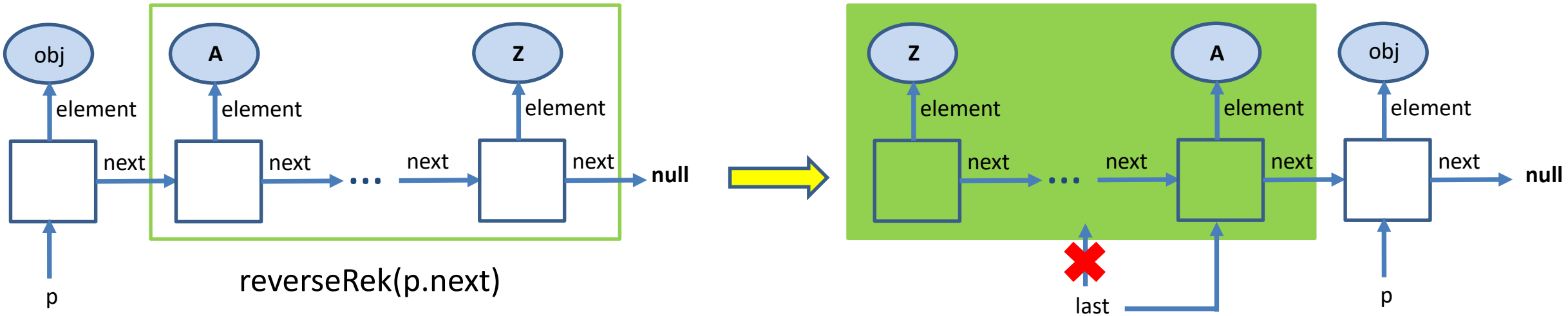
lengthRek(LinkedListElement p)



lengthRek(p) = 1 + lengthRek(p.next)

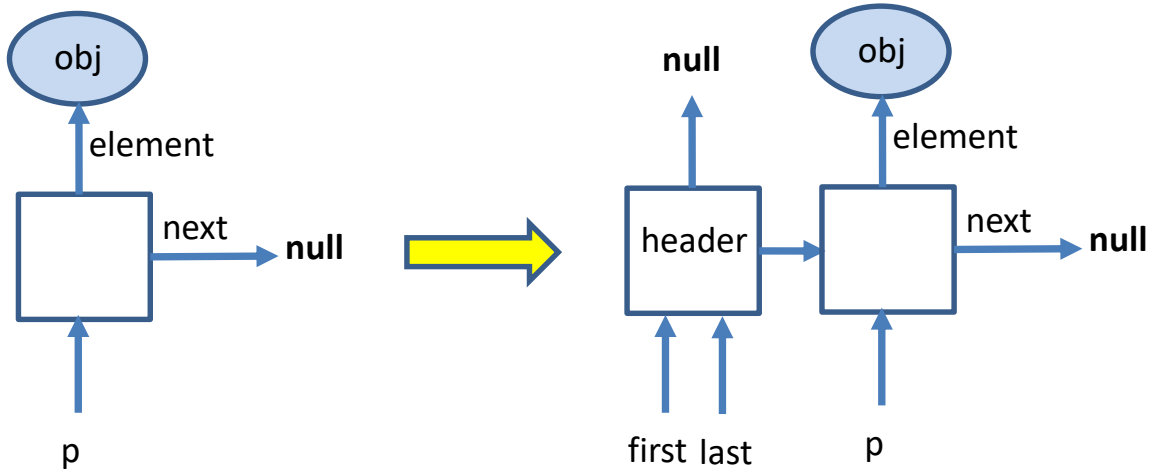
LINEARNI SEZNAM S KAZALCI

reverseRek(LinkedListElement p)



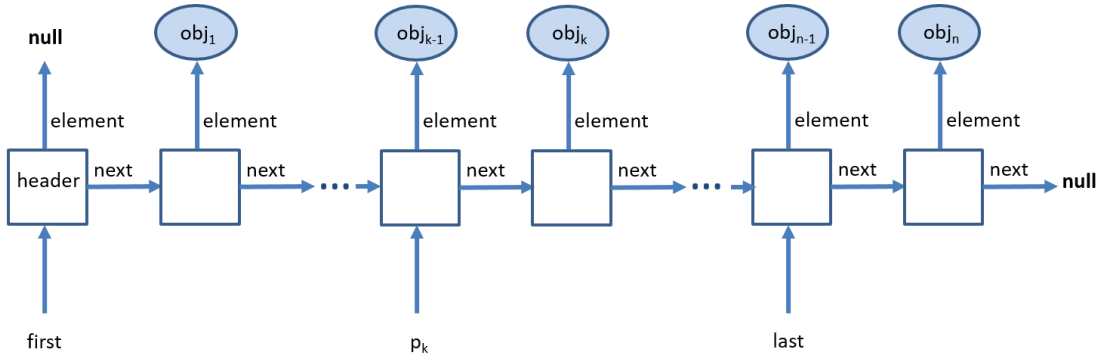
LINEARNI SEZNAM S KAZALCI

reverseRek(LinkedListElement p)



LINEARNI SEZNAM S KAZALCI

write()




```
public void write()
{
        LinkedListElement el;

        //zacnemo pri elementu za glavo seznama
        el = first.next;
        while (el != null)
        {
                System.out.print(el.element + ", ");
                el = el.next;
        }

        System.out.println();
}
```