

DVOSMERNI SEZNAM S KAZALCI

Vsaka celica seznama ima poleg elementa še dva kazalca:

- kazalec na naslednji element
- kazalec na predhodni element

Omogoča učinkovito iskanje predhodnika v seznamu – **$O(1)$**

```
class ListTwoWayLinkedNode {  
    Object element;  
    ListTwoWayLinkedNode next, previous;  
    ...  
}
```

```
public class ListTwoWayLinked {  
    protected ListTwoWayLinkedNode first, last;  
    ...  
}
```



IMPLEMENTACIJA SEZNAMA S POLJEM

- uporablja podatkovno strukturo polje za shranjevanje elementov seznama
- položaj elementa v seznamu je podan z indeksom polja
- potrebujemo še indeks zadnjega elementa v seznamu

```
public class ListArray {
    private Object elements[];
    private int lastElement;
    ...
    public ListArray(int noElem) {
        elements = new Object[noElem];
        ...
    }
    ...
}
```

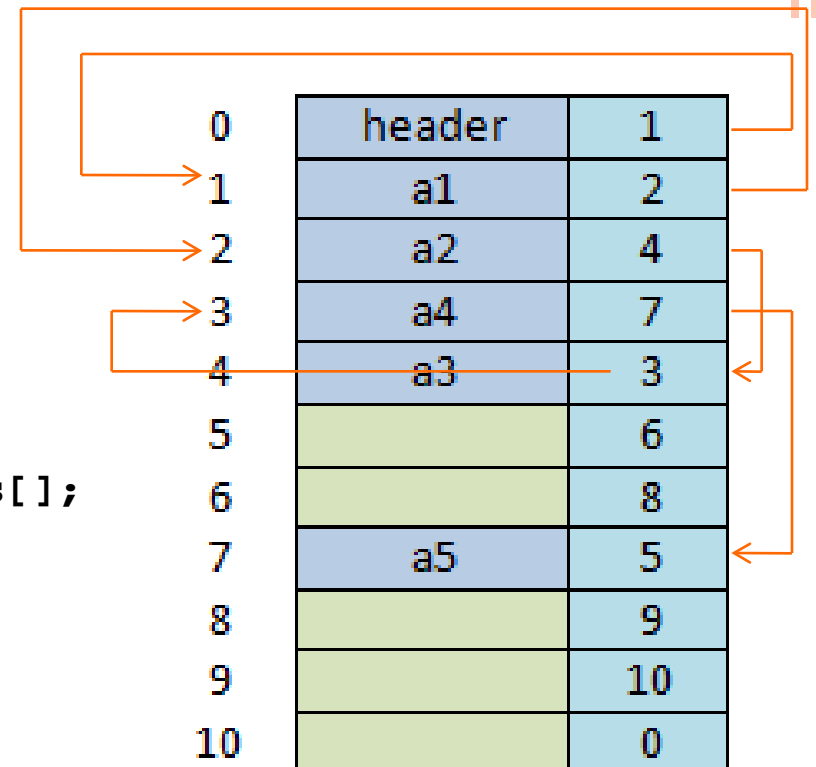


SEZNAM Z INDEKSNIMI KAZALCI

Uporaba v jezikih, ki ne omogočajo dinamičnih podatkovnih struktur.

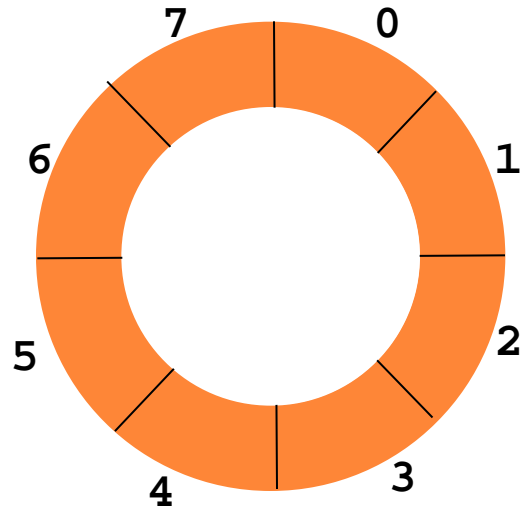
Vsaka celica polja je sestavljena iz elementa in iz indeksa naslednjega elementa.

```
class ListCursorNode {  
    Object element;  
    int next;  
    ...  
}  
  
public class ListCursor {  
    private ListCursorNode cells[];  
    private int last;  
    ...  
}
```



ADT QUEUE

Implementacija s t.i. krožnim poljem (circular array):
vrsta se krožno premika po polju.



Pomagamo si z operatorjem ostanka pri deljenju:

$$1\%5 = 1, \quad 2\%5 = 2, \quad 5\%5 = 0, \quad 8\%5 = 3$$

Premik indeksov front in rear (oziroma števca count):

```
front = (front + 1) % items.length;
```

```
rear = (rear + 1) % items.length;
```

```
rear = (front + count - 1) % items.length
```

