Programming

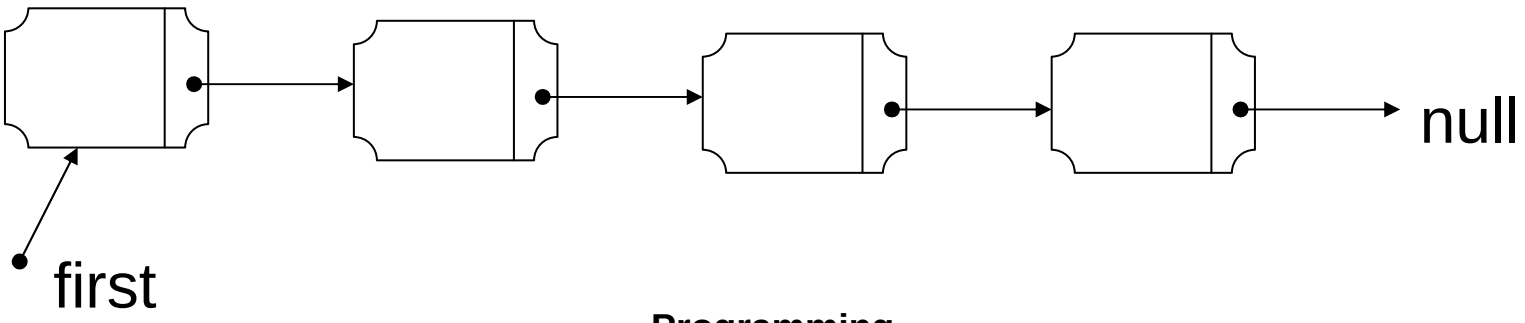
Linear list

6th lecture

What is a linear list?

- A set of elements where each element has a pointer (a reference, a link) to “next” element

- Graphical representation:



Elements of a linear list

- Elements - nodes
- Node - object
- Components
  - Data space
  - Reference to the next element

```
public class Seznam{
    int glava;
    Seznam rep;
//metode
}
```

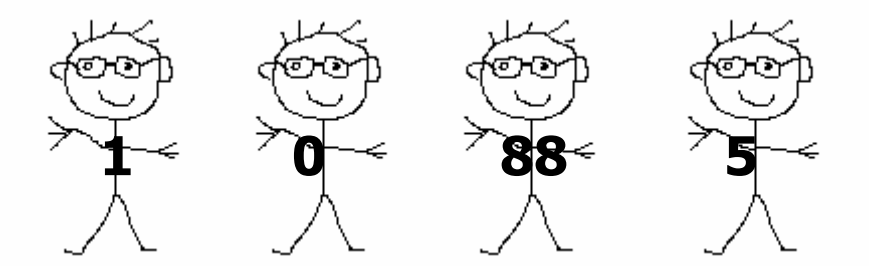
Basics

- Povezani seznam (linked list)
- A special version of the linked list:
  - List as a RDS
  - A list can be: empty or non-empty
  - If the list is non-empty it consists of:
    - head
    - sublist (which can be empty or …)

Lists with management

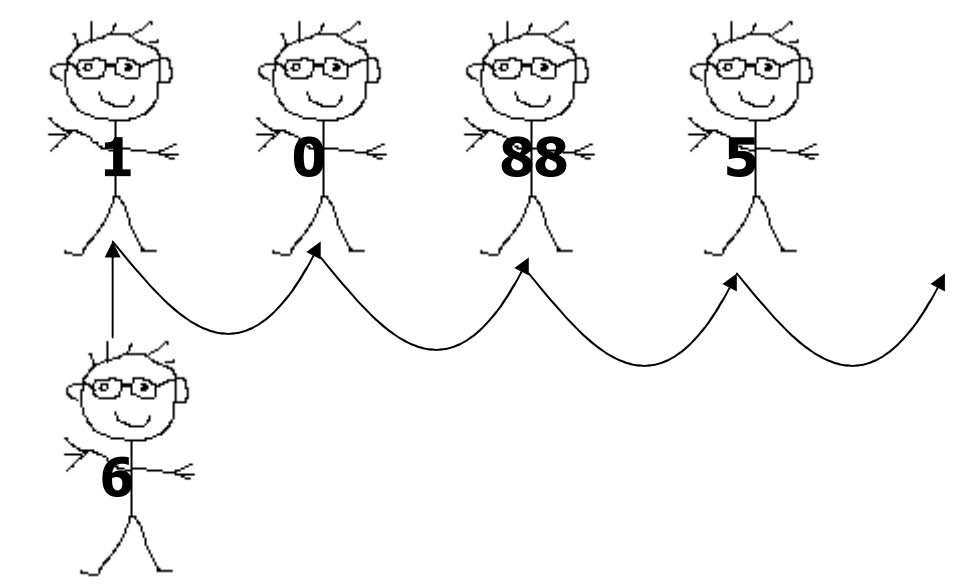
- Is the list empty?
- Move elements in the list.
- All operations of a non-empty list.
- Set the name and get the name.
- Store the length of the list.

Visualization of the list



A list with elements: 1, 0, 88, 5

Visualization of the list



Insert new element: 6, at the end

Visualization of the list

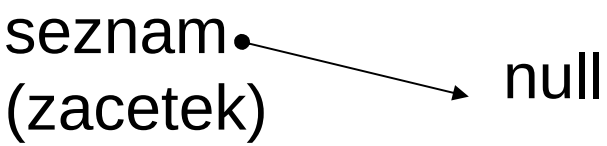
New list

Class Seznam - methods

```
//metoda bo vstavila nov element
public void vstavi(int elt){
  if(rep == null){ //naredili smo kuglo ki kaze na null
    rep = new Seznam(); //kugla ki kaze na null naredi
                                    nov seznam
    rep.glava = elt;
  }
  else{
    rep.vstavi(elt);
  }
}
```

Seznam - problem

- vstavi prvi element

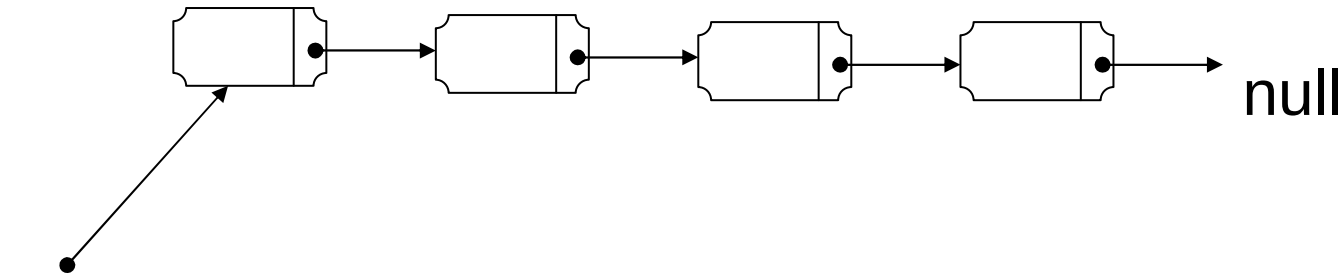


Class Seznam - methods

```
public boolean brisi(int elt){
  if(glava==elt){
    return(false); //to se ne zgodi
  }
  else{
    if(rep==null){
      return(false);
    }
    else{
      if (rep.glava==elt){
        rep=rep.rep;
        return(true);
      }
      return(rep.brisi(elt));
    }
  }
}
```

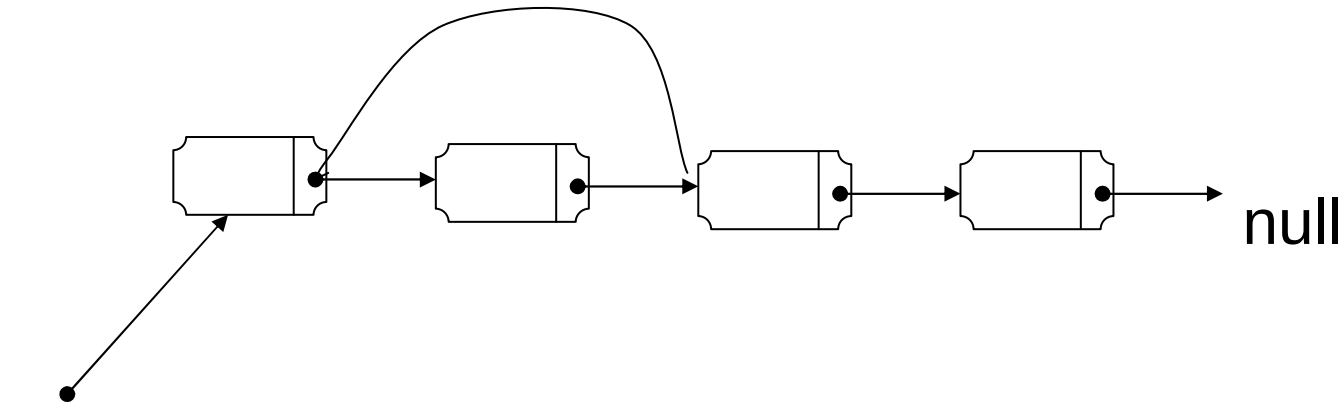
Seznam - problem

- Brišemo element



Seznam - rešitev

- Brišemo naslednji element



Seznam – nov problem

- Kaj pa prvi element?

Class Seznam - methods

```
public boolean najdi(int elt){
  if(glava==elt){
    return(true);
  }
  else{
    if(rep==null){ //nima iskanega elta
      return(false);
    }
    else{
      return(rep.najdi(elt));
    }
  }
}
```

Class PovezanSeznam

```
public class PovezanSeznam{
    Seznam sz;

      //methods
}
```

Class PovezanSeznam

```
public void vstavi(int elt){
  if(sz!=null){
    sz.vstavi(elt);
  }
  else {
    sz=new Seznam();
    sz.glava=elt;
  }
}
```

Class PovezanSeznam

```
public boolean najdi(int elt){
  if(sz!=null){
    return(sz.najdi(elt));
  }
  else {
    return(false);
  }
}
```

Class PovezanSeznam

```
public boolean brisi(int elt){
  if(sz!=null){ //ce seznam ni null, obstaja
vsaj 1 kugla
    if(sz.glava==elt){
      sz=sz.rep; //cel seznam preko kuglice
zbrisemo
      return(true);
    }
    else {
      return(sz.brisi(elt));
    }
  }
  else {
    return(false); //ce ni nobene kuglice,
nimamo kaj nardit
  }
```

Class PovezanSeznam

```
public static void main(String a[]){
  PovezanSeznam ps = new PovezanSeznam();
  ps.vstavi(10);
  ps.vstavi(15);
  System.out.println(ps.brisi(40));
  System.out.println(ps.brisi(15));
  ps.vstavi(55);
  ps.vstavi(66);
  System.out.println(ps.najdi(55));
}
```

Razred PovezanSeznam

```
public static void main(String a[]){
  PovezanSeznam ps = new PovezanSeznam();
  ps.vstavi(10);
  ps.vstavi(15);
  System.out.println(ps.brisi(40));
  System.out.println(ps.brisi(15));
  ps.vstavi(55);
  ps.vstavi(66);
  System.out.println(ps.najdi(55));
}
```