ADT LIST – ABSTRAKTNI RAZRED

```
public abstract class List {
   public abstract void makenull() ;
   public abstract Object first() ;
   public abstract Object last() ;
   public abstract Object next(Object pos) ;
   public abstract Object previous(Object pos) ;
   public abstract Object end() ;
   public abstract Object retrieve(Object pos) ;
   public abstract void insert(Object x) ;
   public abstract void insert(Object x, Object pos) ;
   public abstract void delete(Object pos) ;
   public abstract boolean overEnd(Object pos) ;
   public abstract boolean empty() ;
} // class List
```

ADT LIST – ABSTRAKTNI RAZRED

```
public abstract class List {

    public Object locate(Object x) {
      for (Object iter= first() ; !overEnd(iter) ; iter=next(iter))
        if (x.equals(retrieve(iter)))
            return iter ;
       return null ;
    }
    public void printList() {
      for (Object iter = first() ; ! overEnd(iter) ; iter=next(iter))
          System.out.print(retrieve(iter)+ ",␣") ;
      System.out.println();
    }


} // class List
```

Primer: Dolžina seznama

Iterativno:

```
public int len() {
    int n = 0 ;
    for (Object iter = first() ; ! overEnd(iter) ; iter=next(iter))
        n++ ;
    return n ;
 }
```

Rekurzivno:

- Seznam je sestavljen iz prvega elementa (glave) in iz repa, ki je tudi seznam, le da ima en element manj.
- Seznam je podan z položajem prvega elementa (first).
- Rep je podan s položajem next(first).

Primer: Dolžina seznama

Rekurzivno:

1) Rekurzijska spremenljivka: pos = položaj elementa na začetku je pos=first
2) Kaj mi pomaga dolžina repa seznama len(next(first))? Dolžina celega seznama je len+1.
3) Robni pogoj? overEnd(pos) → len = 0
4) Splošni primer? Glej 2)

```
public int lenRec() {
    return lenRec(first());
 }

 public int lenRec(Object pos) {
     if (overEnd(pos)) return 0 ;
     else return lenRec(next(pos))+1;
  }
```

PRIMER: VSOTA ELEMENTOV SEZNAMA

Iterativno:

```
public int sum() {
    int n = 0 ;
    for (Object iter = first() ; ! overEnd(iter) ; iter=next(iter))
        n = n + (Integer)(retrieve(iter)) ;
    return n ;
  }
```

PRIMER: VSOTA ELEMENTOV SEZNAMA

Rekurzivno:

1) Rekurzijska spremenljivka: pos = položaj elementa na začetku je pos=first
2) Kaj mi pomaga vsota elementov repa sum(next(first))? Vsota elementov celega seznama je sum+retrieve(first).
3) Robni pogoj? overEnd(pos) → sum = 0
4) Splošni primer? Glej 2)

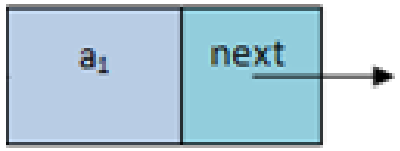
```
public int sumRec() {
      return sumRec(first());
       }

 public int sumRec(Object pos) {
      if (overEnd(pos)) return 0;
      else return sumRec(next(pos)) + (Integer)retrieve(pos);
  }
```

Enosmerni seznam s kazalci

Podatkovna struktura je podana z enim elementom seznama:

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



ter s samim seznamom elementov:

```
public class ListLinked {
   protected ListLinkedNode first, last;
   …
}
```

Seznam je definiran s položajem prvega in zadnjega elementa.

Enosmerni seznam s kazalci

Kreiranje praznega seznama

MAKENULL(L)

```
public void makenull() {
    first = new ListLinkedNode(null,null) ;
    last = null ;
}
```