

# ALGORITMI IN PODATKOVNE STRUKTURE 1



**6. laboratorijske vaje**

**Preslikava**

# PRESLIKAVA

---

$$M(d) = r$$

↑  
element  
(enoličen ključ)

↑  
vrednost

Primer: preslikava  $M(d) = r$  vsaki osebi  $d$  priredi številko bančnega računa  $r$ .

Osnovne operacije:

- `assign(M,d,r)`
- `compute(M,d)`
- `delete(M,d)`

# PRESLIKAVA

Implementacija z urejenim seznamom – hranimo pare  $\langle d_i, r_i \rangle$ , urejeno po ključih  $d_i$

```
class OrderedElement
{
    Comparable element;
    OrderedElement next;
    ...
}

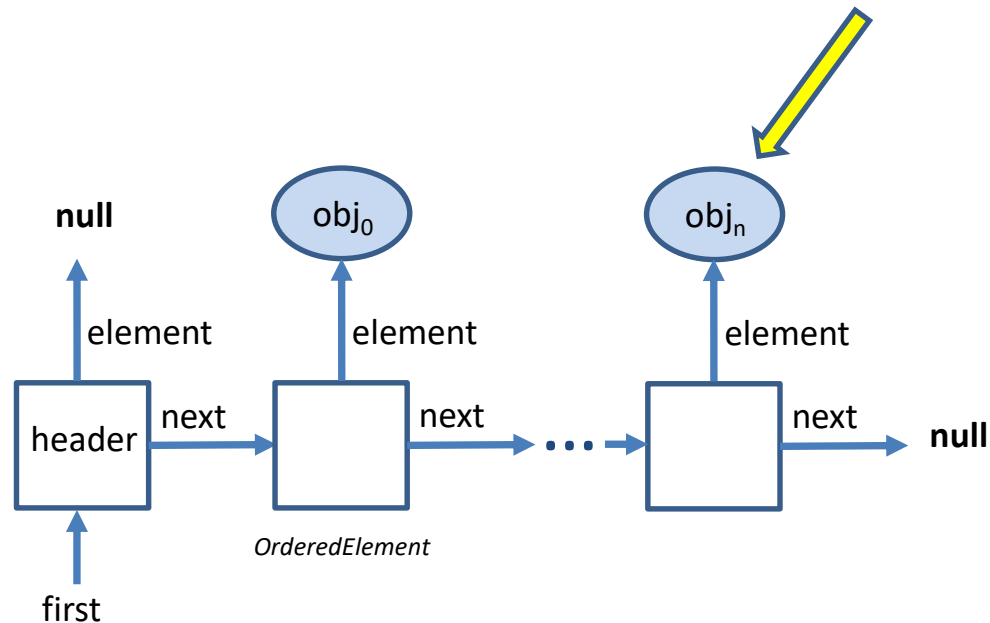
class OrderedLinkedList
{
    OrderedElement first;

    public void insert(Comparable obj) {...}
    ...
}
```

Na primer:

```
class Pair implements Comparable<Pair>
{
    Comparable d;
    Object r;

    public int compareTo(Pair p) {...}
    ...
}
```



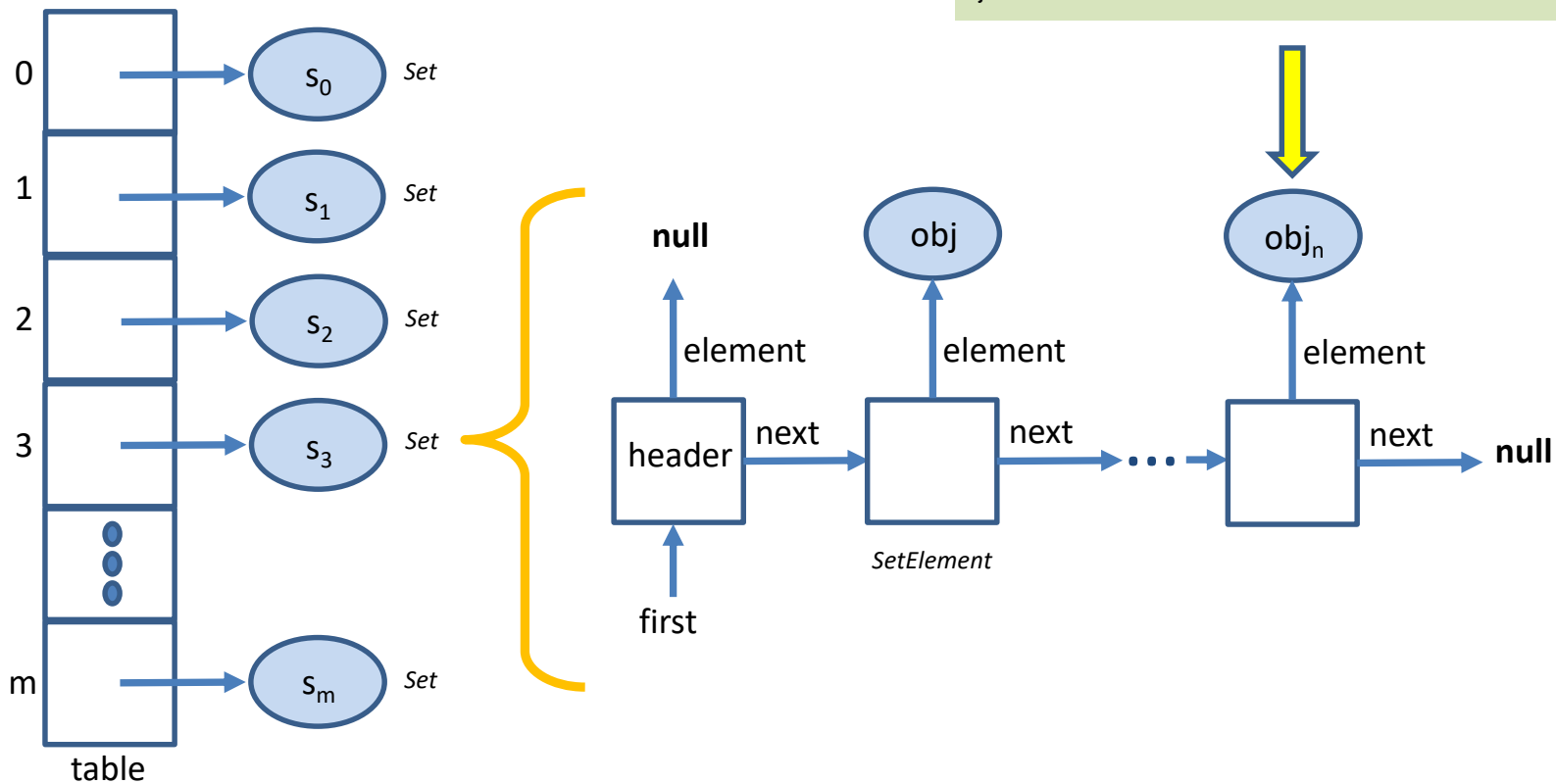
# PRESLIKAVA

Implementacija z odprto zgoščeno tabelo:

```
class HashMap  
{  
    Set[] table;  
  
    private int hash(Object d) {return Math.abs(d.hashCode()) % table.length;}  
    ...  
}
```

Na primer:

```
class HashMapNode  
{  
    Object d;  
    Object r;  
  
    public boolean equals(Object obj) {...}  
    ...  
}
```



# NALOGE

---

Implementirajte naslednje metode v razredu `HashMap`:

- `void assign(Object d, Object r)` – doda nov par  $(d, r)$  v preslikavo  $M$ .
- `Object compute(Object d)` – vrne vrednost  $M(d)$ .
- `void delete(Object d)` – odstrani par  $(d, *)$  iz preslikave  $M$ .
- `void rehash(int size)` – zgradi novo zgoščeno tabelo podane velikosti.

**Opcijsko** implementirajte naslednjo metodo v razredu `Oseba`:

- `int hashCode()` – predstavlja numerično predstavitev vsebine predmeta.