Algoritmi in podatkovne strukture 1

6. laboratorijske vaje

Preslikava

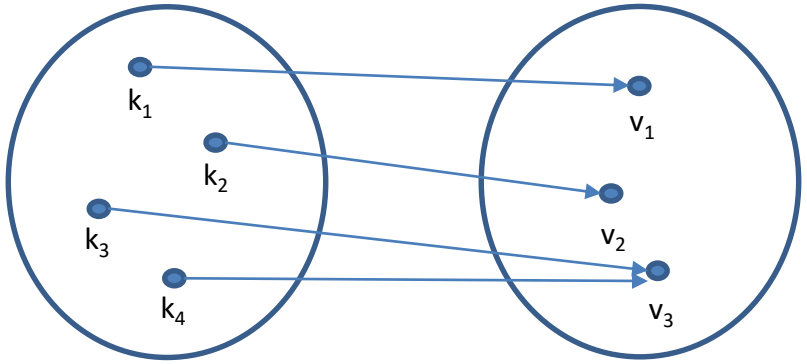
PRESLIKAVA

- Preslikava je zbirka parov (ključ, vrednost)

$$M(\textit{ključ}) = \textit{vrednost}$$

- Preslikava omogoča operacije dodajanja para, branja ali odstranjevanja para s pomočjo ključa.

- Ključi v preslikavi so vedno enolični (različni pari z enakim ključem niso dovoljeni).



Množica ključev Množica vrednosti

Primer: preslikava domenskega imena (ključ) v IP naslov (vrednost).

PRESLIKAVA

Implementacija z urejenim seznamom – hranimo pare (k_i,v_i), urejeno po ključih k_i

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

class OrderedLinkedList
{
    OrderedElement first;

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

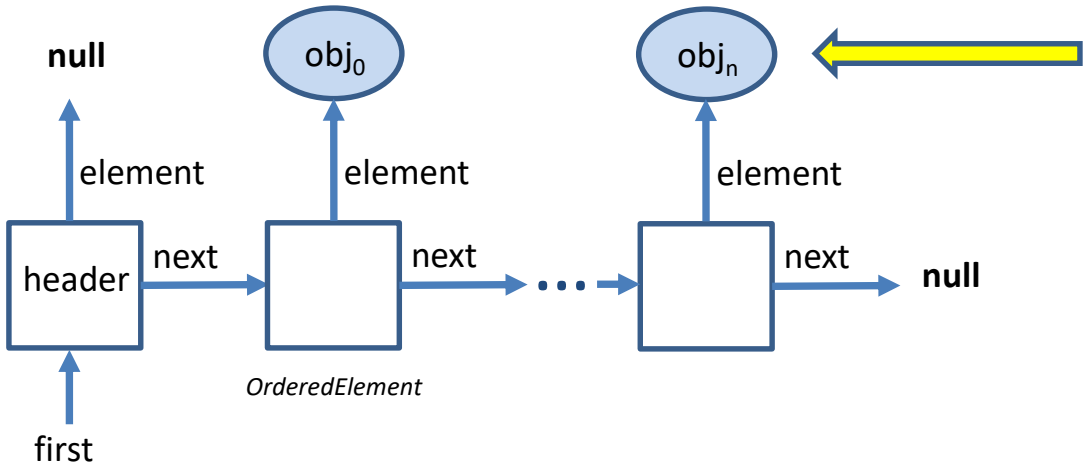
Vmesnik **Comparable** zahteva, da razred podpira metodo public int compareTo(Object o).

`a.compareTo(b)`

če je a < b, metoda vrne negativno celo število

če je a > b, metoda vrne pozitivno celo število

če je a = b, metoda vrne 0



Kot elemente seznama hranimo instance razreda Pair:

```
class Pair implements Comparable<Pair>
{
    Comparable key;
    Object value;

    public int compareTo(Pair p)
    {
        return key.compareTo(p.key);
    }
    …
}
```

PRESLIKAVA

Implementacija z odprto zgoščeno tabelo:

```
class HashMap
{
     Set[] table;

     private int hash(Object key) {return Math.abs(key.hashCode()) % table.length;}
     …
}
```

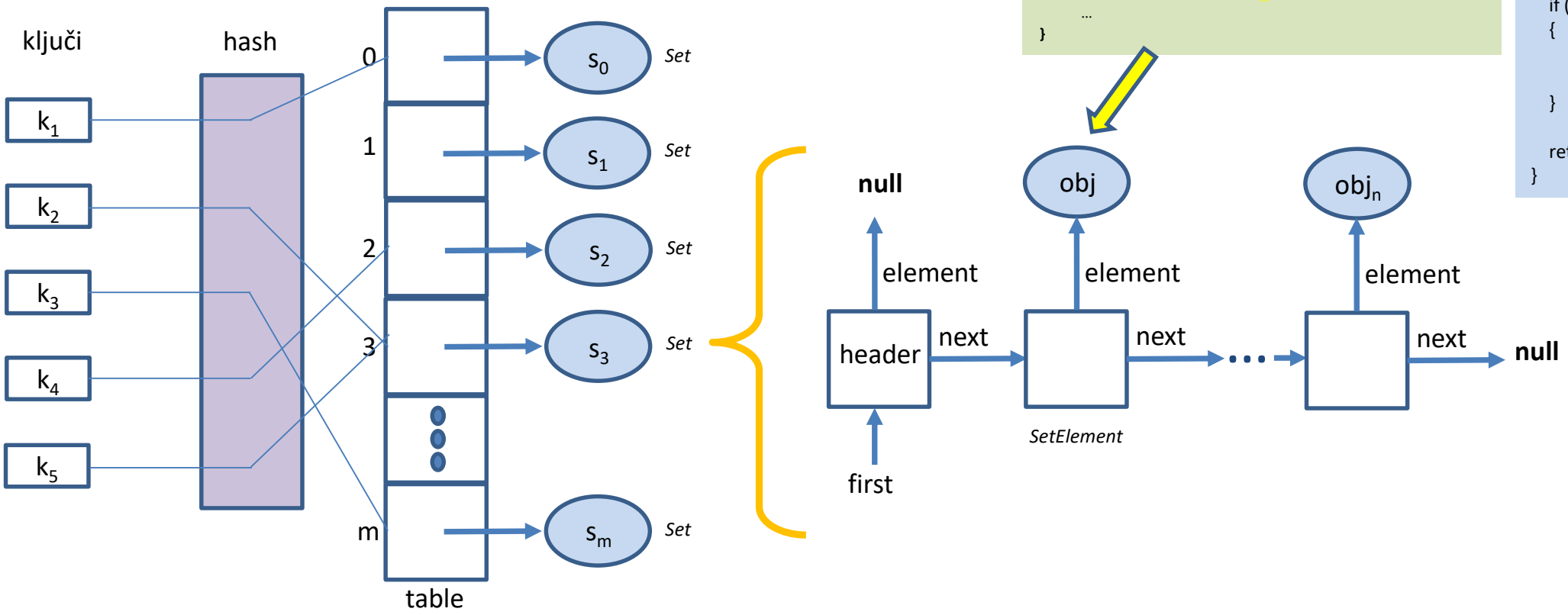
Kot elemente množic hranimo instance razreda HashMapNode:

```
class HashMapNode
{
     Object key;
     Object value;

     public boolean equals(Object obj) {…}
     …
}
```

```
public boolean equals(Object obj)
{
   if (obj instanceof HashMapNode)
   {
      HashMapNode n = (HashMapNode)obj;
      return key.equals(n.key);
   }

   return false;
}
```



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.

ZGLED ZA OGREVANJE (METODA COMPUTE)

```
public Object compute(Object key)
{
    Set l = table[hash(key)];
    SetElement pos = l.locate(new HashMapNode(key, null));
    if (pos != null)
        return ((HashMapNode) l.retrieve(pos)).getValue();

    return null;
}
```