Napredno programiranje i programski jezici

08 Java

Fakultet tehničkih nauka, Novi Sad
23-24/Z
Dunja Vrbaški

```
public static void main(String[] args) {

	ArrayList<Osoba> osobe = new ArrayList<Osoba>();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);
}
```

```
public static void main(String[] args) {

	ArrayList<Osoba> osobe = new ArrayList<Osoba>();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	osobe.add(2, new Osoba("Draco", "Malfoy", "444444"));
	System.out.println(osobe);
}
```

```
public static void main(String[] args) {

	ArrayList<Osoba> osobe = new ArrayList<Osoba>();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	osobe.add(2, new Osoba("Draco", "Malfoy", "444444"));
	System.out.println(osobe);

	osobe.remove(2);
	System.out.println(osobe);
}
```

```
public static void main(String[] args) {

	ArrayList<Osoba> osobe = new ArrayList<Osoba>();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	osobe.add(2, new Osoba("Draco", "Malfoy", "444444"));
	System.out.println(osobe);

	osobe.remove(2);
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0);
	System.out.println(osoba0);

	osobe.remove(osoba0);
	System.out.println(osobe);
}
```

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	//... dodavanje

	osobe.add(2, new Osoba("Draco", "Malfoy", "444444"));
	System.out.println(osobe);

	osobe.remove(2);
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0);
	System.out.println(osoba0);

	osobe.remove(osoba0);
	System.out.println(osobe);

	if (osobe.contains(osoba0))
		System.out.println("osoba 0 je u listi");
	else
		System.out.println("osoba 0 nije u listi");
}
```

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	//... dodavanje

	osobe.add(2, new Osoba("Draco", "Malfoy", "444444"));
	System.out.println(osobe);

	osobe.remove(2);
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0);
	System.out.println(osoba0);

	osobe.remove(osoba0);
	System.out.println(osobe);

	if (osobe.contains(osoba0))
		System.out.println("osoba 0 je u listi");
	else
		System.out.println("osoba 0 nije u listi");
}
```

Moramo znati šta radimo

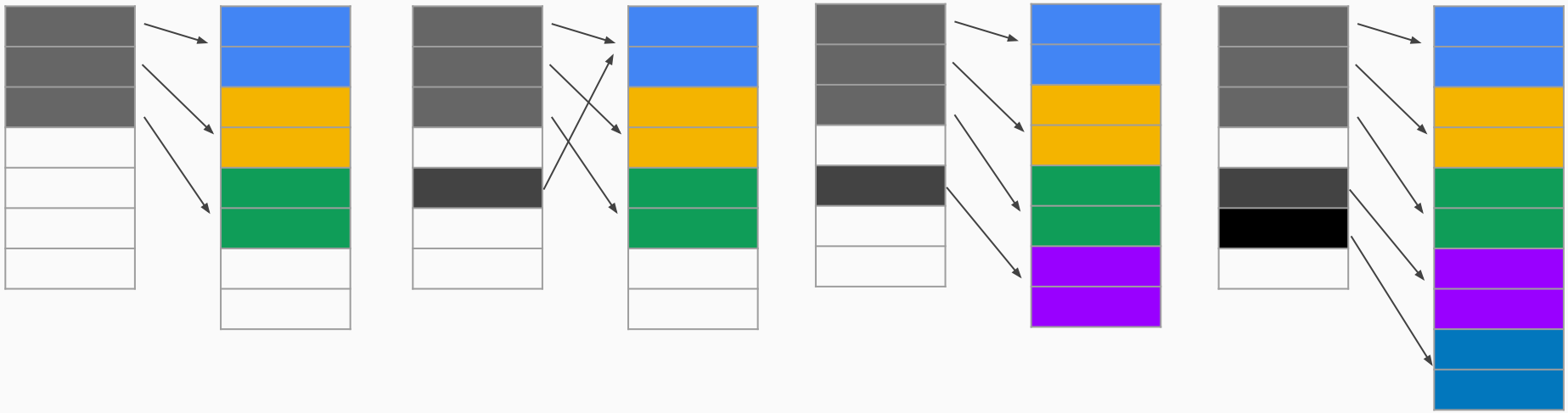
```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	//... prethodno
	osobe.clear();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0); // Hermione
	proveriOsobu(osobe, osoba0);

	osoba0 = new Osoba("Cho", "Chang", "555555");
	proveriOsobu(osobe, osoba0);

	Osoba osobaTemp = new Osoba("Hermione", "Granger", "111111");
	proveriOsobu(osobe, osobaTemp);
}

private static void proveriOsobu(ArrayList<Osoba> osobe, Osoba osoba ) {
	if (osobe.contains(osoba))
		System.out.println("osoba je u listi");
	else
		System.out.println("osoba  nije u listi");
}
```

?

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	//... prethodno
	osobe.clear();

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0); // Hermione
	proveriOsobu(osobe, osoba0);

	osoba0 = new Osoba("Cho", "Chang", "555555");
	proveriOsobu(osobe, osoba0);

	Osoba osobaTemp = new Osoba("Hermione", "Granger", "111111");
	proveriOsobu(osobe, osobaTemp);
}

private static void proveriOsobu(ArrayList<
	if (osobe.contains(osoba))
		System.out.println("osoba je u
	else
		System.out.println("osoba  nij
}
```

```
[Hermione Granger 111111 , Harry Potter 222222 , Ron Weasley 333333 ]
osoba je u listi
osoba  nije u listi
osoba  nije u listi
```

```
osobe.add(new Osoba("Hermione", "Granger", "111111"));
osobe.add(new Osoba("Harry", "Potter", "222222"));
osobe.add(new Osoba("Ron", "Weasley", "333333"));

Osoba osoba0 = osobe.get(0); // Hermione

osoba0 = new Osoba("Cho", "Chang", "555555");

Osoba osobaTemp = new Osoba("Hermione", "Granger", "111111");
```



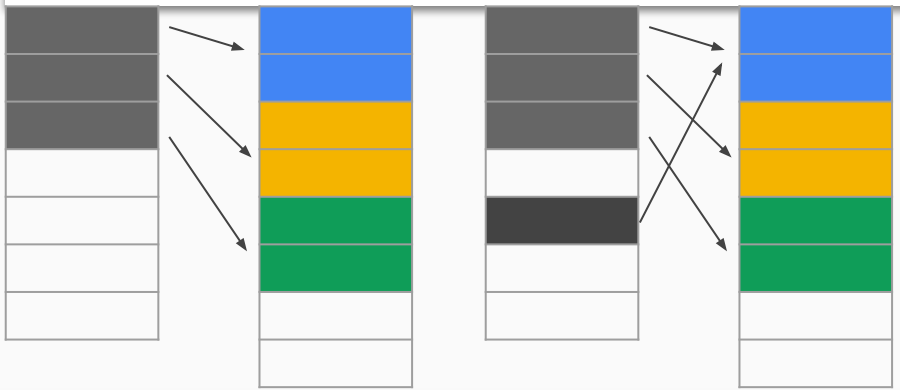
```
public static void main(String[] args) {
      ArrayList<Osoba> osobe = new ArrayList<Osoba>();
      ...

      osobe.add(new Osoba("Hermione", "Granger", "111111"));
      osobe.add(new Osoba("Harry", "Potter", "222222"));
      osobe.add(new Osoba("Ron", "Weasley", "333333"));
      System.out.println(osobe);

      Osoba osoba0 = osobe.get(0);
      osoba0.setIme(osoba0.getIme() + "*");
      System.out.println(osobe);
}
```

?

```
public static void main(String[] args) {
      ArrayList<Osoba> osobe = new ArrayList<Osoba>();
      ...

      osobe.add(new Osoba("Hermione", "Granger", "111111"));
      osobe.add(new Osoba("Harry", "Potter", "222222"));
      osobe.add(new Osoba("Ron", "Weasley", "333333"));
      System.out.println(osobe);

      Osoba osoba0 = osobe.get(0);
      osoba0.setIme(osoba0.getIme() + "*");
      System.out.println(osobe);
}
```

```
[Hermione Granger 111111 , Harry Potter 222222 , Ron Weasley 333333 ]
[Hermione* Granger 111111 , Harry Potter 222222 , Ron Weasley 333333 ]
```

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));
	System.out.println(osobe);

	Osoba osoba0 = osobe.get(0);
	osoba0.setIme(osoba0.getIme() + "*");
	System.out.println(osobe);
}
```



```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));

	osobe.add(null);

	Osoba osobaNull = null;
	osobe.add(osobaNull);

	System.out.println(osobe);
}
```

?

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...

	osobe.add(new Osoba("Hermione", "Granger", "111111"));
	osobe.add(new Osoba("Harry", "Potter", "222222"));
	osobe.add(new Osoba("Ron", "Weasley", "333333"));

	osobe.add(null);

	Osoba osobaNull = null;
	osobe.add(osobaNull);

	System.out.println(osobe);
}
```

```
[Hermione Granger 111111 , Harry Potter 222222 , Ron Weasley 333333 , null, null]
```

obratiti pažnju

```
public static void main(String[] args) {
      ArrayList<Osoba> osobe = new ArrayList<Osoba>();
      ...

      osobe.add(new Osoba("Hermione", "Granger", "111111"));
      osobe.add(new Osoba("Harry", "Potter", "222222"));
      osobe.add(new Osoba("Ron", "Weasley", "333333"));
      System.out.println(osobe);


}
```

Kako pronaći osobu sa zadatim matičnim brojem?

Pretraživanje kolekcija - čest zadatak

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...
	String mb = "111111";

	if (pronadji(osobe, mb))
		System.out.println("Osoba [MB = " + mb + "] je u listi");
	else
		System.out.println("Osoba [MB = " + mb + "] nije u listi");
}

private static boolean pronadji(ArrayList<Osoba> osobe, String MB) {
	for (Osoba osoba : osobe) {
		if (osoba.getMB() == MB)
			return true;
	}
	return false;
}
```

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...
	String mb = "111111";
	Osoba osobaTemp = pronadji(osobe, mb);

	if (osobaTemp != null)
		System.out.println("Osoba [MB = " + mb + "] je u listi");
	else
		System.out.println("Osoba [MB = " + mb + "] nije u listi")
}

private static Osoba pronadji(ArrayList<Osoba> osobe, String MB) {
	for (Osoba osoba : osobe) {
		if (osoba.getMB() == MB)
			return osoba;
	}
	return null;
}
```

```
public static void main(String[] args) {
	ArrayList<Osoba> osobe = new ArrayList<Osoba>();
	...
	String mb = "111111";
	int index = pronadji(osobe, mb);

	if (index >= 0)
		System.out.println("Osoba [MB = " + mb + "] je u listi");
	else
		System.out.println("Osoba [MB = " + mb + "] nije u listi");
}

private static int pronadji(ArrayList<Osoba> osobe, String MB) {
	for (int i = 0; i < osobe.size(); i++) {
		if (osobe.get(i).getMB() == MB)
			return i;
	}
	return -1;
}
```

Module java.base

Package java.util

Class HashMap<K,V>

java.lang.Object
    java.util.AbstractMap<K,V>
        java.util.HashMap<K,V>

Type Parameters:

K - the type of keys maintained by this map

V - the type of mapped values

All Implemented Interfaces:

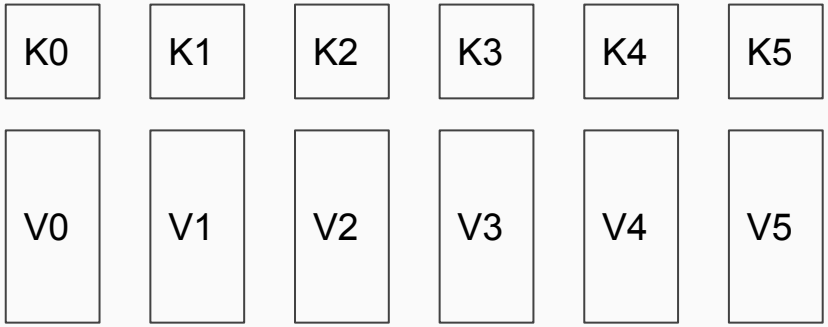
Serializable, Cloneable, Map<K,V>

Direct Known Subclasses:

LinkedHashMap, PrinterStateReasons

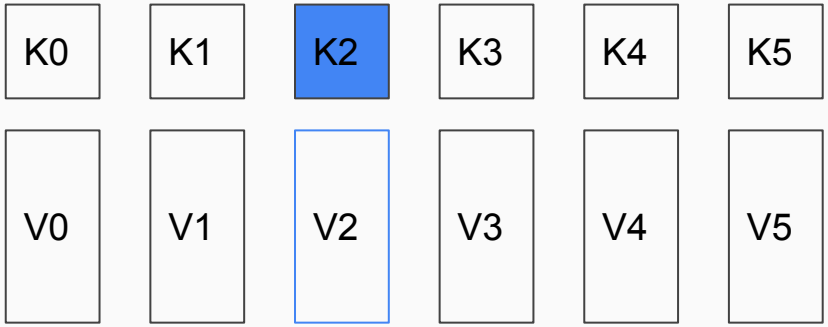
```
public class HashMap<K,V>
extends AbstractMap<K,V>
implements Map<K,V>, Cloneable, Serializable
```

```
package nppj;

import java.util.ArrayList;
import java.util.HashMap;
// import java.util.*

public class KolekcijeTest {

	public static void main(String[] args) {

		HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();
	}
}
```

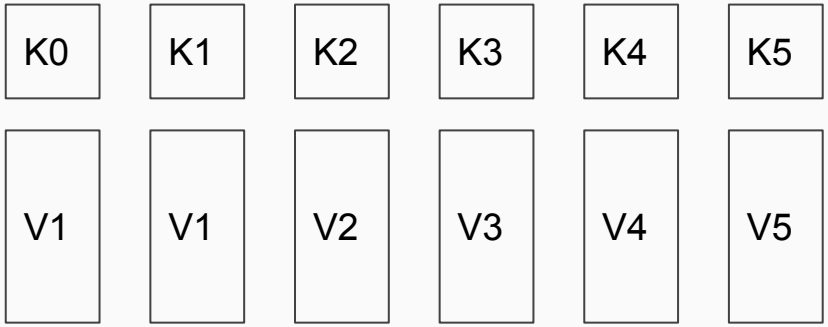


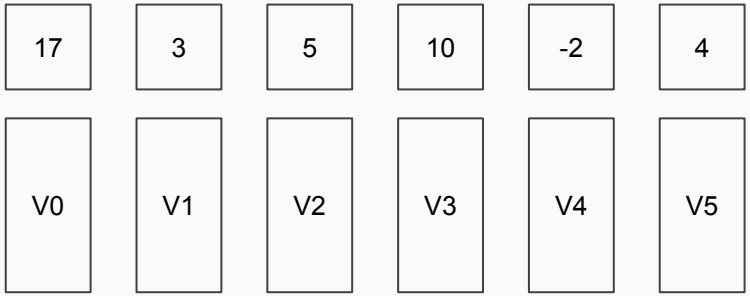
Parovi (ključ, vrednost) <Key, Value>

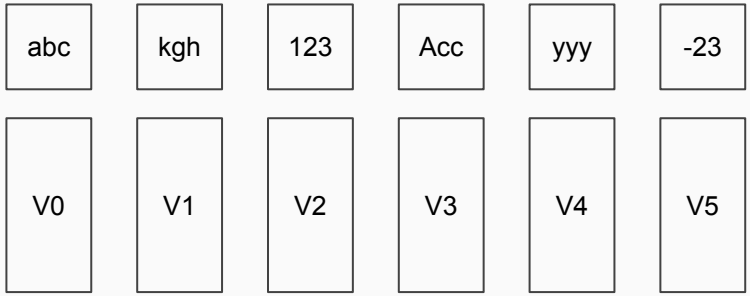
Mapira, preslikava, ključ na svoju vrednost
Ključevi su jedinstveni



Pristupamo vrednosti preko njenog ključa

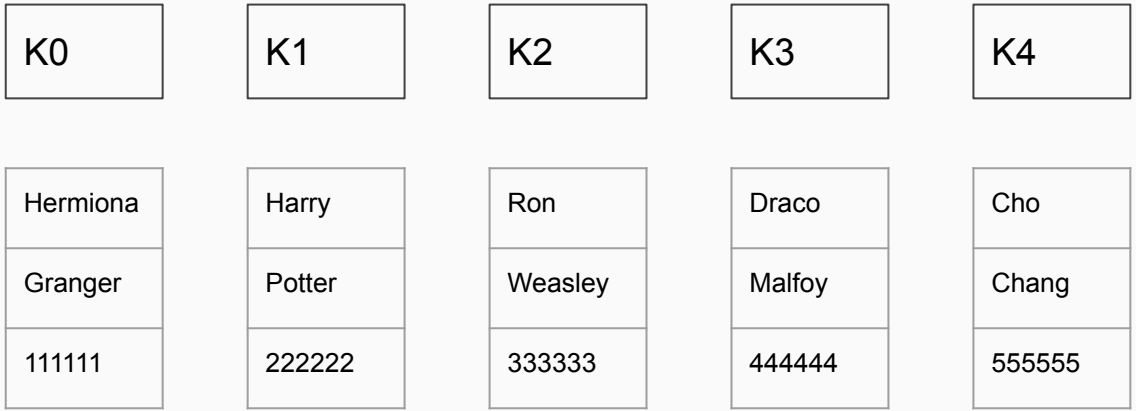

K0
K1
K2
K3
K4
K5
V1
V1
V2
V3
V4
V5
tip: Tip ključa
tip: Tip vrednosti






```
HashMap<int, Osoba>

HashMap<int, Pravougaonik>

HashMap<int, String>

...
```

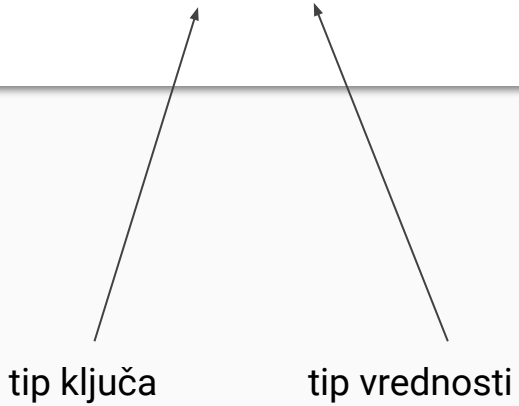
```
HashMap<String, Osoba>

HashMap<String, Pravougaonik>

HashMap<String, String>

...
```


K0
K1
K2
K3
K4
?
Hermiona
Granger
111111
Harry
Potter
222222
Ron
Weasley
333333
Draco
Malfoy
444444
Cho
Chang
555555

111111	222222	333333	444444	555555
Hermiona	Harry	Ron	Draco	Cho
Granger	Potter	Weasley	Malfoy	Chang
111111	222222	333333	444444	555555

Ako objekti imaju polje koje je jednoznačno - možemo iskoristiti kao ključ
Nije pravilo, ali je često

```
public static void main(String[] args) {

	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

}
```



```
public static void main(String[] args) {

	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);
}
```

HashMap je jedna implementacija interfejsa Map

```
public static void main(String[] args) {

	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);
}
```

```
{111111=Hermione Granger 111111 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
```

```
public static void main(String[] args) {

	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	String mb = "111111";
	System.out.println(osobe.get(mb));
}
```

Pristupamo preko ključa!
Ne preko indeksa kao kod niza ili liste.
Fizički raspored elemenata nije važan i u opstem slucaju nepoznat je.

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	Osoba osoba3 = new Osoba("Draco", "Malfoy", "444444");
	osobe.put(osoba0.getMB(), osoba3);

	System.out.println(osobe);
}
```

?

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	Osoba osoba3 = new Osoba("Draco", "Malfoy", "444444");
	osobe.put(osoba0.getMB(), osoba3);

	System.out.println(osobe);
}
```

```
{111111=Hermione Granger 111111 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
{111111=Draco Malfoy 444444 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
```

Paziti!

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	Osoba osoba3 = new Osoba("Draco", "Malfoy", "111111");
	osobe.put(osoba3.getMB(), osoba3);

	System.out.println(osobe);
}
```

```
{111111=Hermione Granger 111111 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
{111111=Draco Malfoy 111111, 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
```

Paziti!

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	Osoba osoba3 = new Osoba("Draco", "Malfoy", "111111");

	if (!osobe.containsKey(osoba3.getMB()))
		osobe.put(osoba3.getMB(), osoba3);

	System.out.println(osobe);
}
```

U realnom sistemu ovo ni ne bi trebalo da se desi. Kontrola na drugim nivoima.

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	osoba0.setMB(osoba2.getMB());
	System.out.println(osobe);


	System.out.println(osobe);
}
```

?

```
public static void main(String[] args) {
	HashMap<String, Osoba> osobe = new HashMap<String, Osoba>();

	Osoba osoba0 = new Osoba("Hermione", "Granger", "111111");
	Osoba osoba1 = new Osoba("Harry", "Potter", "222222");
	Osoba osoba2 = new Osoba("Ron", "Weasley", "333333");

	osobe.put(osoba0.getMB(), osoba0);
	osobe.put(osoba1.getMB(), osoba1);
	osobe.put(osoba2.getMB(), osoba2);

	System.out.println(osobe);

	osoba0.setMB(osoba2.getMB());
	System.out.println(osobe);


	System.out.println(osobe);
}
```

```
{111111=Hermione Granger 111111 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
{111111=Hermione Granger 333333 , 222222=Harry Potter 222222 , 333333=Ron Weasley 333333 }
```

Paziti!

U realnom sistemu promena identifikacionog obeležja ni ne bi trebala da se desi tako, ad-hoc. Kontrola na drugim nivoima.

Kako pronaći osobu sa zadatim matičnim brojem?
Pretraživanje kolekcija - čest zadatak

```
private static Osoba pronadji(
      ArrayList<Osoba> osobe,
      String MB) {

      for (Osoba osoba : osobe) {
            if (osoba.getMB() == MB)
                  return osoba;
      }
      return null;
}
```

```
private static Osoba pronadji(
            HashMap<String, Osoba> osobe,
            String MB) {

      return osobe.get(MB);
}
```

Ne moramo da pretražujemo
Ne treba da pretražujemo (bespotrebno)
Direktan pristup elementu preko ključa

ZADATAK: Dodati u kolekciju osobu ukoliko osoba sa istim matičnim brojem već ne postoji u kolekciji

ArrayList - prođi kroz listu i utvrdi da li postoji isti MB

HashMap - containsKey + put

ZADATAK: Ispisati kolekciju osoba sortirano po nekom polju

ArrayList - primeni algoritam sortiranja

HashMap - ….

Ili druga kolekcija

ili prebaciti vrednosti u listu i sortirati

ZADATAK

(neobavezno)

Pogledati dokumentaciju za ArrayList i HashMap.
Pogledati metode od interesa, šta vraćaju, koji su parametri, da li generišu neke izuzetke.

Podsetiti se šta je HashTable.

Pogledati koje još implementacije interfejsa map postoje.

ZADATAK

(neobavezno)

Isprobati sve iz prezentacije.