



Delo s podatki



Programiranje 2, Tomaž Dobravec



Tip podatkov

- Dva tipa podatkov:

- Vrednosti

ZBIRKA

```
(vrednost1, vrednost2, vrednost3, vrednost4, vrednost5, vrednost6, ...)
```

Primer:

```
("Miha", "Micka", "Jože", "Francka", "Mojca", "Polde", ...)
```

- Pari (ključ, vrednost)

SLOVAR

```
((kljuc1, vrednost1), (kljuc2, vrednost2), (kljuc3, vrednost3), ...)
```

Primer:

```
((63191234, "Micka"), (63209876, "Miha"), (63183232, "Tone"), ...)
```





Uporaba tabele

- Zbirko podatkov lahko hranimo tabeli (to že znamo).
- Tabela ima nekatere lepe lastnosti, vendar ni vedno najboljša možna izbira.
- Slabosti tabele:
 - fiksna velikost,
 - počasno dodajanje elementov na začetek.
- Prednosti tabele:
 - tabela je vgrajena v jezik,
 - enostavna uporaba,
 - hiter dostop do poljubnega elementa preko indeksa.





Podatkovne strukture v Javi

Java pod skupnim imenom **Java Collection Framework (JCF)** ponuja veliko tipov in orodij za delo z večjo količino podatkov.

Delo z zbirko podatkov

java.util.Collection

(vrednost1, vrednost2, ...)

java.util.Map

Delo s slovarjem podatkov

((ključ1, vrednost1), (ključ2, vrednost2), ...)



Nekateri vmesniki in razredi JCF

Vmesniki (abstraktni podatkovni tipi)

Collection

Set

SortedSet

List

Map

SortedMap

Razredi (izvedbe)

<- **HashSet**

<- **TreeSet**

<- **ArrayList**

<- **LinkedList**

<- **Stack**

<- **HashMap**

<- **TreeMap**

<http://java.sun.com/docs/books/tutorial/collections/>





java.util.Collection

Nekatere metode vmesnika Collection

```
boolean add(Object o)
```

```
boolean remove(Object o)
```

```
int size()
```

```
boolean contains(Object o)
```

```
boolean isEmpty()
```

```
void clear()
```

```
Iterator iterator()
```





Primer uporabe množice

- Množica (`TreeSet`) za shranjevanje dni v tednu.

prevajalniku sporočimo,
kakšnega tipa bodo
podatki v zbirki

uporabimo lahko vse
metode, ki jih predpisuje
vmesnik `Collection`

```
TreeSet<String> dnevi = new TreeSet<>();  
  
dnevi.add("PON");  
dnevi.add("TOR");  
dnevi.add("SR");  
  
System.out.println(dnevi.size());           // 3  
System.out.println(dnevi.contains("PON"));  // true
```



Iterator

- Iterator uporabimo za “sprehod” čez podatke
- Metode iteratorja:

```
public boolean hasNext()  
public Object next()  
public void remove()
```

- Primer uporabe:

prevajalniku sporočimo,
kakšnega tipa bodo
podatki, po katerih se
bo “sprehodil” iterator

```
Iterator<String> it = dnevi.iterator();  
while (it.hasNext()) {  
    String dan = it.next();  
    System.out.println(dan);  
}
```





Zanka `for` za sprehod po podatkih

Zanka `for` ima v javi dve obliki:

- klasična zanka z uporabo indeksov

```
for (int i=0; i<10; i++)
```

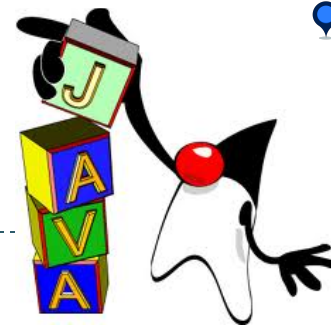
- zanka za “sprehod” po podatkovni zbirki

```
for (Tip element: zbirka)
```

Primer:

```
for (String dan: dnevi) {  
    System.out.println(dan);  
}
```





Napiši program, ki za dani števili a in b izpiše

- a) vsa števila, ki delijo bodisi a bodisi b ,
- b) vsa števila, ki delijo a in b (skupni delitelj).



Seznam elementov (`ArrayList`)

`ArrayList` je podoben tabeli

- ▶ vsak element ima svoj indeks;
- ▶ dodajamo lahko na določeno mesto (`set`) ali na konec (`add`).

Prednost:

- ▶ na začetku ni treba navesti velikosti seznama;
- ▶ elemente lahko brišemo (`remove`).

Primer: Seznam imen.



Slovarji (java.util.Map)

- Za zbirke, katerih podatki so tipa (vrednost, ključ)
- Primerjava tabele in slovarja
 - pri **tabeli** se na elemente sklicuješ po številskem indeksu:

```
String a[] = new String[5];  
a[1] = "Triglav";  
...  
System.out.println(a[1]);
```

- pri **slovarju** za sklic navedeš poljuben objekt:

```
HashMap m = new HashMap();  
m.put("Januar", 31);  
...  
System.out.println(m.get("Januar"));
```





Slovarji (java.util.Map)

- ključ (indeks v zbirki) mora biti enoličen

Primer: Kaj izpiše spodnji program?

```
m.put("Januar", 31);  
m.put("Januar", 30);  
System.out.println(m.get("Januar"));
```

- ključi se primerjajo z metodo `equals()`, pri razporejanju po koših se uporabi metoda `hashCode()`.



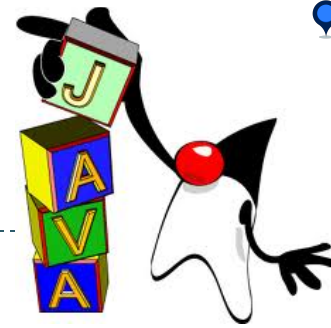


Slovarji (java.util.Map)

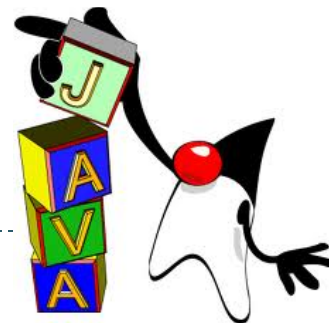
Nekatere metode vmesnika Map:

```
public int size()  
public boolean isEmpty()  
public boolean containsKey(Object key)  
public boolean containsValue(Object value)  
public Object get(Object key)  
public Object put(Object key, Object value)  
public Object remove(Object key)  
public Set keySet()  
public Collection values()
```





Napiši program, ki v slovarju hrani število dni posameznega meseca. Vse podatke iz slovarja nato izpiši s pomočjo iteratorja.



oi/Drzava.java, oi/Drzave.java

V datoteki `drzave.txt` so zapisani podatki o državah, ki sodelujejo na olimpijskih igrah. Izdelaj razred `Drzava` za hranjenje podatkov o eni državi (kratica, glavno mesto, število prebivalcev), nato preberi podatke iz datoteke `drzave.txt` v slovar (ključ: kratica države, vrednost: objekt razreda `Drzava`).

`drzave.txt`:

```
SLO:Ljubljana:2009245
ITA:Rome:58147733
AVT:Vienna:8199783
FIN:Helsinki:5238460
ZDA:Washington, DC:301139947
NEM:Berlin:82400996
FRA:Paris:64057790
SPA:Madrid:40448191
CRO:Zagreb:4493312
```

- test1: iz tipkovnice preberemo kratico, izpišemo vse podatke o državi,
- test2: izpis vseh držav (uporaba `iteratorja` po vrednostih),
- test3: izpis vseh držav (uporaba `iteratorja` po ključih).





Urejanje podatkov

Java zna urejati podatke, ki so shranjeni v

tabeli

- `Arrays.sort()`

zbirki

- `Collections.sort()`





Urejanje podatkov

Kakšnega tipa morajo biti podatki, da jih Java zna urediti z vgrajeno metodo `sort()` ?

Java zna urediti zbirko le v primeru, da zna podatke med seboj primerjati!

Java zna primerjati

- ▶ podatke primitivnih tipov (`int`, `double`, `char`, ...) in
- ▶ podatke tipov, ki imajo *implementiran vmesnik* `Comparable` (med njimi je med drugim tudi `String`).



Urejanje podatkov - Comparable

Če želimo primerjati objekte razreda `Oseba` glede na njihovo višino, razred `Oseba` napišemo takole:

```
class Oseba implements Comparable<Oseba> {  
    String ime;  
    int visina;  
  
    public int compareTo(Oseba o) {  
        return new Integer(this.visina).compareTo(o.visina);  
    }  
}
```

```
// ... uporaba (na primer v main metodi):  
ArrayList<Oseba> osebe = new ArrayList<Oseba>();  
osebe.add(...);  
Collections.sort(osebe);
```





Urejanje podatkov - Comparator

Urejamo lahko tudi tako, da ob klicu metode `sort()` podamo objekt razreda `Comparator`:

```
ArrayList<Oseba> osebe = new ArrayList<Oseba>();  
osebe.add(. . . );
```

```
Collections.sort(osebe, new Comparator<Oseba>() {  
    public int compare(Oseba o1, Oseba o2) {  
        return o1.ime.compareTo(o2.ime);  
    }  
});
```

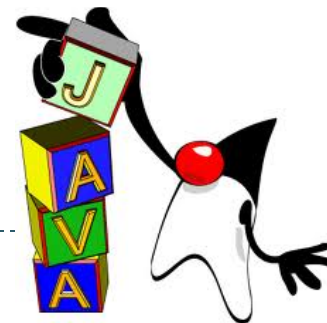




Orodja za delo z zbirkami

- Metode iz razreda `java.util.Collections`
 - `sort(List l)`
 - `max(Collection c)`
 - `min(Collection c)`
 - `replaceAll(List l, Object o, Object no)`
 - `swap(List l, int i, int j)`
- `java.util.Arrays.sort(Object[] o)`





| No. ↕ | Athlete ↕ | Nation ↕ | Sport ↕ | Years ↕ | Games ↕ | Gender ↕ | Gold ↕ | Silver ↕ | Bronze ↕ | Total ↕ |
|-------|----------------------|---------------|----------------------|-----------|---------|----------|--------|----------|----------|---------|
| 1 | Michael Phelps | United States | Swimming | 2000-2016 | Summer | M | 23 | 3 | 2 | 28 |
| 2 | Larisa Latynina | Soviet Union | Gymnastics | 1956-1964 | Summer | F | 9 | 5 | 4 | 18 |
| 3 | Marit Bjørgen | Norway | Cross-country skiing | 2002-2018 | Winter | F | 8 | 4 | 3 | 15 |
| 4 | Nikolai Andrianov | Soviet Union | Gymnastics | 1972-1980 | Summer | M | 7 | 5 | 3 | |
| 5 | Ole Einar Bjørndalen | Norway | Biathlon | 1998-2014 | Winter | M | 8 | 4 | 1 | 13 |
| 6 | Boris Shakhlin | Soviet Union | Gymnastics | 1956-1964 | Summer | M | 7 | 4 | 2 | |
| 7 | Edoardo Mangiarotti | Italy | Fencing | 1936-1960 | Summer | M | 6 | 5 | 2 | |
| 8 | Takashi Ono | Japan | Gymnastics | 1952-1964 | Summer | M | 5 | 4 | 4 | |
| 9 | Paavo Nurmi | Finland | Athletics | 1920-1928 | Summer | M | 9 | 3 | 0 | |

V datoteki `medalje.txt` so zbrani podatki o večkratnih prejemnikih olimpijskih medalj. Podatke iz datoteke preberi in izpiši:

- 1) tri zimske in tri letne športe, v katerih so športniki dobili največ medalj
- 2) seznam kratic držav, urejen po skupnem številu prejetih medalj
- 3) seznam držav, urejen glede na normirano skupno število medalj glede na število prebivalcev države.