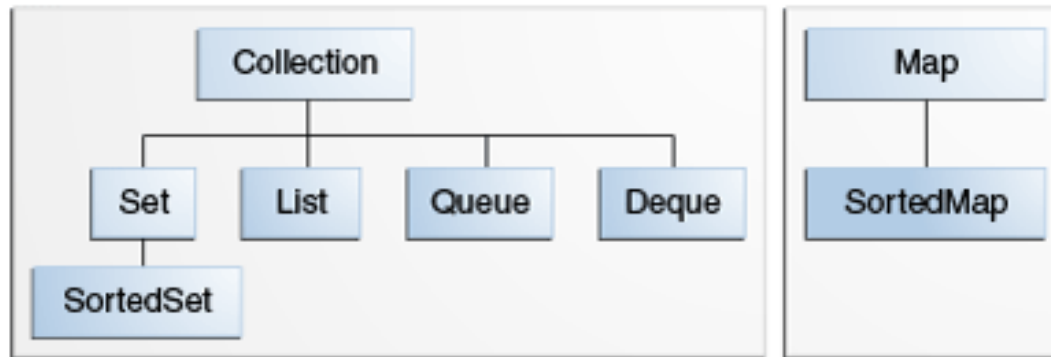


Kolekcije

Ugrađeni interfejsi za kolekcije



Ugrađeni interfejsi - svojstva

- **Collection** – grupa elemenata - kolekcija
- **Set** – kolekcija bez ponavljanja elemenata - skup
- **List** – uređena kolekcija
- **Queue** – uređena kolekcija sa dodavanjem na kraj i skidanjem sa početka – FIFO (first in – first out)
- **Deque** – kao Queue, samo skidanje i dodavanje na oba kraja
- **Map** – kolekcija uređenih parova objekata - mapa (ključ – vrednost)
- **SortedSet** – uređeni skup
- **SortedMap** – mapa u kojoj su ključevi uređeni

Collection interfejs

- Ugrađene operacije:
 - add
 - remove
 - contains
 - clear
 - isEmpty
 - addAll
 - containsAll
 - ...

Set interfejs

- Ugrađene implementacije:
 - HashSet – najbrži, ne garantuje uređenost
 - TreeSet – sporiji, uređenje po vrednostima
 - LinkedHashSet – brz, uređenost prema unetom redosledu
- Sakrivanje implementacije iza interfejsa:

```
Set<String> s1 = new HashSet<String>();
```

```
Set<String> s2 = new TreeSet<String>();
```

Set interfejs

```
Set<String> s1 = new HashSet<String>();  
Set<String> s2 = new TreeSet<String>();  
Set<String> s3 = new LinkedHashSet<String>();  
String[] reci = new String[]{"aaa", "aab", "zzzzz", "ba"};  
for(String r : reci){  
    s1.add(r); s2.add(r); s3.add(r);  
}  
for(String s : s1) System.out.println(s); //šta je ispis?  
for(String s : s2) System.out.println(s); //šta je ispis?  
for(String s : s3) System.out.println(s); //šta je ispis?
```

Operacije nad Set interfejsom

- `s1.containsAll(s2)` : Ispituje da li s1 sadrži sve iz s2
- `s1.addAll(s2)` : Dodaje sve iz s2 u s1
- `s1.retainAll(s2)` : Zadržava u s1 sve koji se nalaze i u s2
- `s1.removeAll(s2)` : Uklanja iz s1 sve koji su u s2

- Kako bi napisali metodu za simetričnu razliku dva skupa korišćenjem pomenutih metoda?

- **Simetrična razlika dva skupa je skup koji sadrži sve one elemente koji se nalaze u jednom od skupova, ali nisu zajednički.**

List interfejs

- Ugrađene implementacije:
 - ArrayList – brža, indeksirana nizom
 - LinkedList – sporija, povezana pokazivačima
- Diskusija o internoj implementaciji ArrayList-e i LinkedList-e
 - složenosti operacija:
 - umetanja - `list.add(i, el)`
 - pronalaženja – `list.get(i)`
 - brisanja – `list.remove(i)`

List operacije

- Operacije koje imaju smisla samo u uređenim kolekcijama:
 1. `Collections.sort(list)`
 2. `Collections.shuffle(list)`
 3. `Collections.reverse(list)`
 4. `Collections.binarySearch(list, el)`
 5. ...
- (pogledati i ostale statičke metode u klasi `Colletions`)

Map interfejs

- Ugrađene implementacije:
 - HashMap
 - TreeMap
 - LinkedHashMap
- Ponašanje slično kao kod različitih implementacija skupova
- Osnovne operacije:
 - `get(key)`
 - `put(key, value)`

Map interfejs

```
Map<Integer,Osoba> registar = new HashMap<Integer, Osoba>();  
registar.put(1011986840201, new Osoba("Aleksandar", "Kartelj"));  
registar.put(3342532352532, new Osoba("Marko", "Markovic"));  
...
```

```
Osoba o1 = registar.get(3423532533523); //vraća null  
Osoba o2 = registar.get(1011986840201); //vraća nastavnika
```

```
Map<Integer, Osoba> registar2 = new TreeMap<Integer, Osoba>();  
registar2.putAll(registar);
```

```
registar.keySet(); //vraća skup ključeva  
registar.values(); //vraća kolekciju vrednosti
```

Zadatak 1

- Korisnik unosi sa konzole spisak imena sve dok ne unese reč KRAJ.
 - A. Ispisati na konzoli spisak imena sortiran leksikografski
 - B. Ispisati na konzoli spisak imena sortiran prema rastućem broju slova (proslediti Comparator u Collections.sort)
 - C. Ispisati na konzoli leksikografski sortiran spisak imena bez ponavljanja (skup)
 - D. Ispisati na konzoli skup imena sa brojem pojavljivanja

Zadatak 1 – test primeri

Ulaz: Ana Petar Mirko Petar Darko Nikola Ana Petar KRAJ

Izlaz:

A: Ana Ana Darko Mirko Nikola Petar Petar Petar

B: Ana Ana Mirko Petar Petar Petar Darko Nikola

C: Ana Darko Mirko Nikola Petar

D:

Ana 2

Darko 1

Mirko 1

Nikola 1

Petar 3

Zadatak 2

- Napisati program koji računa frekvencije reči u tekstu datom u datoteci tekst.txt
- Pritom podrazumevati da su dve reči iste bez obzira na veličinu slova koja se koriste.
- Takođe, pre računanja frekvencija, eliminisati tačke, zareze itd.

Zadatak 2 – test primeri

- Ovo je početak teksta sadržanog u tekst.txt
- Ako bi samo za taj deo pustili program:
„Far far away, behind the word mountains, far from“
- Očekivani izlaz bi bio:
far 3
away 1
behind 1
the 1
word 1
mountains 1
from 1