Java Collections Framework

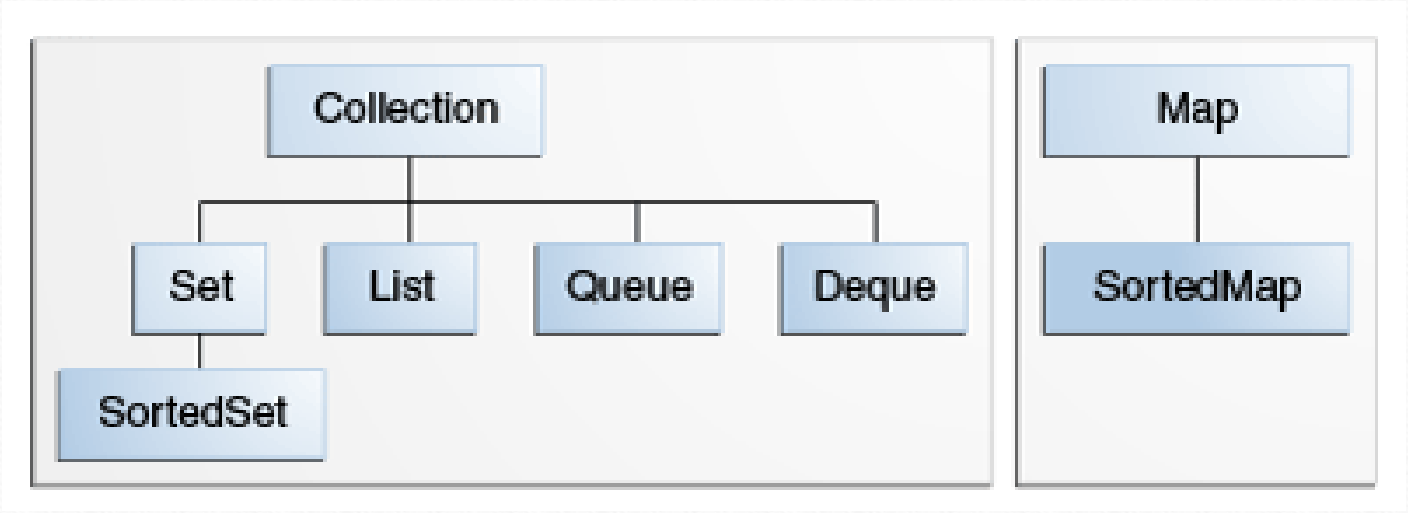
Branislav Vidojević

branislav.vidojevic@outlook.com

Java Collections Framework (JCF)

- Jedinstvena arhitektura za čuvanje i manipulisanje grupom objekata.
- Sadrži:
  - Interfejse (apstraktni tipovi koji sadrže specifikaciju rada sa kolekcijama)
  - Implementacije (konkretne klase koje implementiraju interfejse)
  - Algoritme (polimorfne metode za rad sa kolekcijama)

Interfejsi



Interfjsi

- Collection interfejs – generički interfejs (Java Generics)

```
public interface Collection <E>…
```

- Hijerahija Map interfejsa je odvojena od Collection interfejsa
- Mape u suštini nisu kolekcije, ali su deo JCF-a

Klase

https://docs.oracle.com/javase/tutorial/collections/implementations/index.html

General-purpose Implementations					
Interfaces	**Hash table Implementations**	**Resizable array Implementations**	**Tree Implementations**	**Linked list Implementations**	**Hash table + Linked list Implementations**
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Queue					
Deque		ArrayDeque		LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

Klase

- **Pravilo:** Razmišljati o interfejsu, ne i implementaciji
- Od implementacije zavise samo performanse
- **Preporuka:** kolekcije deklarisati putem interfejsa, a inicijalizovati putem konkretne implementacije. Na taj način program ne postaje zavistan od implementacije same kolekcije.
- **Primer:** `List<Osoba> kolekcija = new LinkedList<>();`

Algoritmi

- Sorting
- Shuffling
- Routine Data Manipulation
- Searching
- Composition
- Finding Extreme Values

Algoritmi - Sorting

```
List<String> lista = new LinkedList<>();
lista.add(“maca”); lista.add(“peca”); lista.add(“aca”);
Collections.sort(lista);
System.out.println(lista);


=> [aca, maca, peca]
```

Algoritmi - Shuffling

```
List<String> lista = new ArrayList<String>();
lista.add("A"); lista.add("E"); lista.add("I"); lista.add("O"); lista.add("U");
Collections.shuffle(obj);
System.out.println(obj);


=> [I, U, A, O, E]
```

Algoritmi – Routine Data Manipulation

- **reverse** - okreće redosled elemenata u listi
- **fill** - stavlja novu datu vrednost umesto svakog elementa kolekcije
- **copy** - kopira elemente jedne liste u drugu, koja mora imati dužinu makar jednaku listi iz koje se kopiraju elementi. Ako je druga lista duža, ostatak elemenata ostaje neizmenjen.
- **swap** - menja pozicije datim elementima liste.
- **addAll** - dodaje elemente kolekciji. Mogu biti prosleđeni kao objekti, ili kao niz.

Algoritmi - Searching

```
int pos = Collections.binarySearch(list, key);
if (pos < 0)
   l.add(-pos-1, key);
//pos predtavlja index elementa gde se nalazi element koji se traži
//ukoliko elementa nema u kolekciji, vraća -1
```

Algoritmi - composition

- **frequency** – broji broj ponavljanja nekog elementa u kolekciji
- **disjoint** – proverava da li su dve kolekcije disjunktni skupovi

Algoritmi – Finding Extreme Values

- min – vraća najmanji element kolekcije
- max – vraća najveći element kolekcije

- Natural Ordering
- Comparable Intefrace
- https://docs.oracle.com/javase/tutorial/collections/interfaces/order.html

Iteracija

- For Loop
- Advanced For Loop
- Iterator
- While Loop
- Collections's stream() util (Java8)

Iteracija – for loop

```
for (int i = 0; i < lista.size(); i++){
   //kod koji se izvršava u jednoj iteraciji
   //element liste uzeti lista.get(i)
}
```

Iteracija – Advanced for loop

```
for (Object o : lista){
   //kod koji se izvršava u jednoj iteraciji
   //element liste pristupati preko reference o
   //Na primer: o.toString();
}
```

Iteracija – Iterator

```
Iterator<String> iterator = lista.iterator();
while (lista.hasNext()) {
   System.out.println(lista.next());
}
//primer se odnosi na listu String objekata
```

Iteracija – While Loop

```
int i = 0;
while (i < lista.size()) {
   //ovde kod koji će se izvršiti u jednoj iteraciji
   //element liste uzeti lista.get(i)
   i++;
}
```

Iteracija – Collections's stream() util (Java8)

```
lista.forEach((obj) -> {
   System.out.println(obj);
});
```

Korisni linkovi

- https://docs.oracle.com/javase/tutorial/collections/
- http://www.tutorialspoint.com/java/java_collections.htm
- http://www.vogella.com/tutorials/JavaCollections/article.html
- http://www.mkyong.com/java/java-object-sorting-example-comparable-and-comparator/
- http://crunchify.com/how-to-iterate-through-java-list-4-way-to-iterate-through-loop/