

Mapy

Table of Contents

Map	1
Příklad Map	1
Metody Map I	1
Metody Map II	2
Metody Map III	2
Příklad iterace mapy	3
Implementace mapy HashMap	3

Map

Asociativní pole, mapa, slovník

- ukládá **dvojici klíč — hodnota**
- umožňuje rychlé vyhledání hodnoty podle klíče
- klíče v mapě jsou vždy unikátní
- mapa je **kontejner** — dynamická datová struktura
- mapa rozhodně **není** `Collection<E>`, nemá jednotlivé prvky
- implementuje rozhraní `Map<K, V>`
 - **K** = objektový typ klíče, **V** = objektový typ hodnoty
 - např. mapa ID a osob — `Map<Long, Person>`

Příklad Map

Následující mapa ukládá značky aut (klíče) a počty kusů (hodnoty):

```
Map<String, Integer> vehicles = new HashMap<>();
vehicles.put("BMW", 2);
vehicles.put("Audi", 4);
vehicles.put("Opel", 1);

vehicles.get("BMW"); // 2
```

Metody Map I

`int size()`

velikost mapy

`void clear()`

vyprázdní mapu

`boolean isEmpty()`

`true`, když je mapa prázdná

`boolean containsKey(Object key)`

dotaz na přítomnost klíče

`boolean containsValue(Object value)`

dotaz na přítomnost hodnoty

`V remove(Object key)`

odstraní dvojici s klíčem `key`, vrací hodnotu (nebo `null`)

`V replace(K key, V value)`

nahradí existující klíč hodnotou

Metody Map II

`V put(K key, V value)`

- vloží dvojici *klíč — hodnota* do mapy
- jestli daný klíč už existuje, hodnota je **přepsána**
- vrací přepsanou hodnotu nebo `null`

`V putIfAbsent(K key, V value)`

- vloží dvojici pouze v případě, že klíč zatím v mapě neexistuje

`V get(Object key)`

- výběr hodnoty odpovídající zadanému klíči
- jestli klíč neexistuje, vrací `null`

`V getOrDefault(Object key, V defaultValue)`

- vrací hodnotu daného klíče nebo defaultní hodnotu

Metody Map III

`Set<K> keySet()`

- vrací **množinu** všech klíčů
- Proč množina? Každý klíč je v mapě maximálně jednou

`Collection<V> values()`

- vrací **kolekci** všech hodnot (může obsahovat duplicity)

`Set<Map.Entry<K,V>> entrySet()`

- vrací množinu typu `Map.Entry` pro iteraci kolekce

- obsahuje metody `getKey()`, `getValue()`



Pro vkládání mapy do mapy existuje `putAll`.

Příklad iterace mapy

```
Map<Integer, String> map = Map.ofEntries(
    entry(1, "a"),
    entry(2, "b")
);

for (Map.Entry<Integer, String> entry : map.entrySet()) {
    System.out.println("key: " + entry.getKey());
    System.out.println("value: " + entry.getValue());
}
```

Implementace mapy `HashMap`

- `HashMap` je implementována pomocí hašovací tabulky, kde haš je zahašovaný klíč, hodnota tabulky je dvojice (*klíč, hodnota*).
- Složitost základních operací:
 - v praxi závisí na kvalitě hašovací funkce (metody `hashCode`) na ukládaných objektech,
 - teoreticky se blíží složitosti *konstantní*, $O(1)$.
- Klíče nejsou uspořádané, nelze iterovat v pořadí klíčů.
- Uspořádané mapy jsou `TreeMap`, viz dále.



Kolekce `HashSet` je implementována pomocí `HashMap` — klíč je prvek, hodnota je "dummy object".



[Javadoc třídy `HashMap`](#)