

Map

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Map<K,V>

- <https://docs.oracle.com/javase/10/docs/api/java/util/Map.html>
- É uma coleção de pares chave / valor
 - Não admite repetições do objeto chave
 - Os elementos são indexados pelo objeto chave (não possuem posição)
 - Acesso, inserção e remoção de elementos são rápidos
- Uso comum: cookies, local storage, qualquer modelo chave-valor
- Principais implementações:
 - **HashMap** - mais rápido (operações $O(1)$ em tabela hash) e não ordenado
 - **TreeMap** - mais lento (operações $O(\log(n))$ em árvore rubro-negra) e ordenado pelo `compareTo` do objeto (ou `Comparator`)
 - **LinkedHashMap** - velocidade intermediária e elementos na ordem em que são adicionados

Alguns métodos importantes

- put(key, value), remove(key), containsKey(key), get(key)
 - Baseado em equals e hashCode
 - Se equals e hashCode não existir, é usada comparação de ponteiros
- clear()
- size()
- keySet() - retorna um Set<K>
- values() - retornaa um Collection<V>

Demo 1

```
package application;

import java.util.Map;
import java.util.TreeMap;

public class Program {

    public static void main(String[] args) {

        Map<String, String> cookies = new TreeMap<>();

        cookies.put("username", "maria");
        cookies.put("email", "maria@gmail.com");
        cookies.put("phone", "99771122");

        cookies.remove("email");
        cookies.put("phone", "99771133");

        System.out.println("Contains 'phone' key: " + cookies.containsKey("phone"));
        System.out.println("Phone number: " + cookies.get("phone"));
        System.out.println("Email: " + cookies.get("email"));
        System.out.println("Size: " + cookies.size());

        System.out.println("ALL COOKIES:");
        for (String key : cookies.keySet()) {
            System.out.println(key + ": " + cookies.get(key));
        }
    }
}
```

Demo 2

```
package application;

import java.util.HashMap;
import java.util.Map;

import Entities.Product;

public class Program {

    public static void main(String[] args) {

        Map<Product, Double> stock = new HashMap<>();

        Product p1 = new Product("Tv", 900.0);
        Product p2 = new Product("Notebook", 1200.0);
        Product p3 = new Product("Tablet", 400.0);

        stock.put(p1, 10000.0);
        stock.put(p2, 20000.0);
        stock.put(p3, 15000.0);

        Product ps = new Product("Tv", 900.0);

        System.out.println("Contains 'ps' key: " + stock.containsKey(ps));
    }
}
```

```
package entities;

public class Product {

    private String name;
    private Double price;

    public Product(String name, Double price) {
        this.name = name;
        this.price = price;
    }

    // getters, setters, equals, hashCode
}
```