

# java.util.Map

---

ALBERTO FERRARI

# java.util.Map<K,V>

---

Map non estende Collection

memorizza insiemi di coppie di elementi contraddistinti da una chiave e dal valore ad essa associato

la chiave identifica univocamente un elemento in una mappa

- non è possibile che in una mappa siano presenti due elementi con lo stesso valore di chiave

la mappa (Dictionary nel mondo .Net) è una collezione di oggetti il cui scopo principale è quello di rendere veloci ed efficienti operazioni di inserimento e ricerca di elementi

K e V (il tipo delle chiavi e il tipo dei valori) non possono essere tipi primitivi (~~int, char, double~~) ma tipi reference (classi)

- per ogni tipo primitivo in java è presente una classe wrapper

# Tipi primitivi e classi wrapper

---

## Primitive type

boolean

byte

char

float

int

long

short

double

## Wrapper class

Boolean

Byte

Character

Float

Integer

Long

Short

Double

# Metodi principali

---

V put(K key, V value)

- Associa il valore alla chiave. Se la mappa contiene già un valore associato a questa chiave il precedente valore viene rimpiazzato
- Restituisce il precedente valore associato alla chiave o null

V get(Object key)

- Restituisce il valore associate alla chiave. null se la chiave non è presente

boolean containsKey(Object key)

- Restituisce true se la mappa contiene un valore associate alla chiave

void clear()

- Elimina tutti gli elementi dalla mappa

Set<K> keySet()

- Restituisce l'insieme delle chiavi

Collection<V> values()

- Restituisce una Collection dei valori presenti nella mappa

# HashMap<K,V>

---

è una delle classi che implementa l'interfaccia Map

alcune altre implementazioni:

- TreeMap<K,V>
- LinkedHashMap<K,V>

Segue un esempio di utilizzo di HashMap

- necessario import `java.util.HashMap`

# put(K,V) - autoboxing

---

```
/* In questo esempio la chiave della mappa è una stringa e il valore è un integer */

HashMap<String, Integer> punteggio;

punteggio = new HashMap<String, Integer>();

/* Inserimento di chiave e valore nella mappa */

punteggio.put("Grifondoro", 100);

/* Autoboxing conversione automatica da tipo primitivo alla classe wrapper corrispondente */

punteggio.put("Serpeverde", 150);
punteggio.put("Tassorosso", 80);
punteggio.put("Corvonero", 65);
```

# size() – replace(K,V) – get(K) - unboxing

---

```
/* Numero di elementi presenti nella mappa size() */  
  
System.out.println("sono presenti " + punteggio.size() + " elementi");  
  
/* Modifica del valore associato a una chiave replace(K key, V value) */  
  
punteggio.replace("Grifondoro", 99);  
  
/* get - si ottiene il valore specificando la chiave */  
  
Integer punti = punteggio.get("Grifondoro");  
  
System.out.println("Grifondoro punti: " + punti);  
  
/* Unboxing - conversione dalla classe wrapper al tipo primitivo corrispondente */  
  
int p = punteggio.get("Tassorosso");  
  
System.out.println("Tassorosso punti: " + p);
```

# get(K) – keySet() – values()

---

```
/* get (chiave non presente) il valore ritornato è null */
punti = punteggio.get("Alpha"); System.out.println("Alpha punti: " + punti);
/* Accesso a tutti gli elementi della mappa keySet() restituisce l'insieme delle chiavi */
System.out.println("Elementi della mappa");
for(String s: punteggio.keySet()) {
    System.out.println(s + " punti: " + punteggio.get(s));
}
/* Accesso a tutti i valori della mappa - values() restituisce una Collection dei valori nella mappa */
System.out.println("Valori inseriti");
Collection<Integer> v = punteggio.values();
for(Integer i: v)
    System.out.println(i);
```

# containsKey(K) – containsValue(V)

---

```
/* Controllo se una chiave è presente containsKey(Object key) */  
if (punteggio.containsKey("Corvonero"))  
    System.out.println("Corvonero fa parte della mappa");  
else  
    System.out.println("Corvonero non fa parte della mappa");  
/* Controllo se un valore è presente containsValue(Object value) */  
if (punteggio.containsValue(155))  
    System.out.println("Il valore 155 è presente nella mappa");  
else  
    System.out.println("Il valore 155 non è presente nella mappa");
```

# clear() – isEmpty()

---

```
/* clear() - Elimina tutti gli elementi della mappa */  
punteggio.clear();  
  
System.out.println("Ora la mappa è vuota");  
  
/* Controllo se la mappa è vuota - isEmpty() */  
if (punteggio.isEmpty())  
    System.out.println("La mappa è vuota");
```

un esempio con la classe Punto