

EnumMap class in Java with Example

EnumMap is specialized implementation of **Map interface** for **enumeration types**.

It extends AbstractMap and implements **Map** Interface in Java. It is a generic class declared as:

Syntax:

```
public class EnumMap<K extends Enum<K>,V>  
K: specifies the keys  
V: specifies values
```

K must extend Enum, which enforces the requirement that the keys must be of specified **enum** type.

Important points:

- EnumMap class is a member of the **Java Collections Framework** & is not synchronized.
- EnumMap is ordered collection and they are maintained in the natural order of their keys(natural order of keys means the order on which enum constant are declared inside enum type)
- It's a high performance map implementation, much faster than **HashMap**.
- All keys of each EnumMap instance must be keys of a single **enum** type.
- EnumMap doesn't allow null key and throw NullPointerException, at same time null values are permitted.

```
// Java program to illustrate working of EnumMap and  
// its functions.  
import java.util.EnumMap;  
  
public class Example  
{  
    public enum GFG  
    {  
        CODE, CONTRIBUTE, QUIZ, MCQ;  
    }  
  
    public static void main(String args[])  
    {  
        // Java EnumMap Example 1: creating EnumMap in java with key  
        //as enum type STATE  
        EnumMap<GFG, String> gfgMap = new EnumMap<GFG, String>(GFG.class);
```



```
// Java EnumMap Example 2:
// putting values inside EnumMap in Java
// we are inserting Enum keys on different order than their natural order
gfgMap.put(GFG.CODE, "Start Coding with gfg");
gfgMap.put(GFG.CONTRIBUTE, "Contribute for others");
gfgMap.put(GFG.QUIZ, "Practice Quizes");
gfgMap.put(GFG.MCQ, "Test Speed with Mcqs");

// printing size of EnumMap in java
System.out.println("Size of EnumMap in java: " + gfgMap.size());

// printing Java EnumMap , should print EnumMap in natural order
// of enum keys (order on which they are declared)
System.out.println("EnumMap: " + gfgMap);

// retrieving value from EnumMap in java
System.out.println("Key : " + GFG.CODE + " Value: "
    + gfgMap.get(GFG.CODE));

// checking if EnumMap contains a particular key
System.out.println("Does gfgMap has :" + GFG.CONTRIBUTE + " : "
    + gfgMap.containsKey(GFG.CONTRIBUTE));

// checking if EnumMap contains a particular value
System.out.println("Does gfgMap has :" + GFG.QUIZ + " : "
    + gfgMap.containsValue("Practice Quizes"));
System.out.println("Does gfgMap has :" + GFG.QUIZ + " : "
    + gfgMap.containsValue(null));
}
```

[Run on IDE](#)

Output:

```
Size of EnumMap in java: 4
EnumMap: {CODE=Start Coding with gfg, CONTRIBUTE=Contribute for others, QUIZ=Practice Quizes
,MCQ=Test Speed with Mcqs}
Key : CODE Value: Start Coding with gfg
Does gfgMap has :CONTRIBUTE : true
Does gfgMap has :QUIZ : true
Does gfgMap has :QUIZ : false
```

Methods

- **put(K key, V value)** : Associates the specified value with the specified key in this map.
- **size()** : Returns the number of key-value mappings in this map.
- **get(Object key)** : Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
- **containsKey(Object key)** : Returns true if this map contains a mapping for the specified key.
- **containsValue(Object value)** : Returns true if this map maps one or more keys to the specified value.

