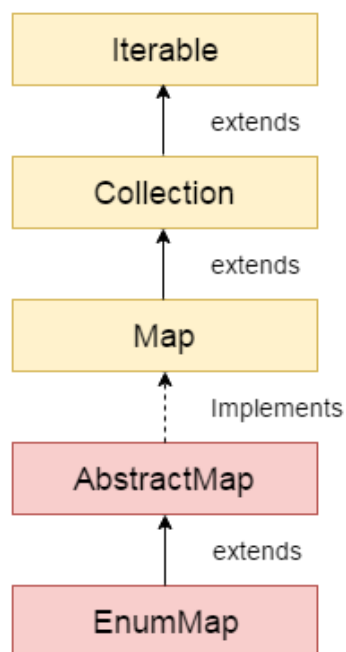


# Java EnumMap class

Java EnumMap class is the specialized Map implementation for enum keys. It inherits Enum and AbstractMap classes.

## EnumMap class hierarchy

The hierarchy of EnumMap class is given in the figure given below.



## EnumMap class declaration

Let's see the declaration for java.util.EnumMap class.

```
public class EnumMap<K extends Enum<K>,V> extends AbstractMap<K,V> implements Serializable, Cloneable
```

## EnumMap class Parameters

Let's see the Parameters for java.util.EnumMap class.

- **K:** It is the type of keys maintained by this map.
- **V:** It is the type of mapped values.

## Constructors of Java EnumMap class

Constructor	Description
-------------	-------------

EnumMap(Class<K> keyType)	It is used to create an empty enum map with the specified key type.
EnumMap(EnumMap<K,? extends V> m)	It is used to create an enum map with the same key type as the specified enum map.
EnumMap(Map<K,? extends V> m)	It is used to create an enum map initialized from the specified map.

## Methods of Java EnumMap class

Method	Description
void clear()	It is used to remove all mappings from this map.
boolean containsKey(Object key)	This method return true if this map contains a mapping for the specified key.
boolean containsValue(Object value)	This method return true if this map maps one or more keys to the specified value.
boolean equals(Object o)	It is used to compare the specified object with this map for equality.
V get(Object key)	This method returns the value to which the specified key is mapped.
V put(K key, V value)	It is used to associate the specified value with the specified key in this map.
V remove(Object key)	It is used to remove the mapping for this key.
Collection<V> values()	It is used to return a Collection view of the values contained in this map.
int size()	It is used to return the number of key-value mappings in this map.

## Java EnumMap Example

```
import java.util.*;

public class EnumMapExample {

    // create an enum
    public enum Days {
        Monday, Tuesday, Wednesday, Thursday
    };

    public static void main(String[] args) {
        //create and populate enum map
        EnumMap<Days, String> map = new EnumMap<Days, String>(Days.class);
        map.put(Days.Monday, "1");
        map.put(Days.Tuesday, "2");
        map.put(Days.Wednesday, "3");
        map.put(Days.Thursday, "4");
        // print the map
        for(Map.Entry m:map.entrySet()){
            System.out.println(m.getKey()+" "+m.getValue());
        }
    }
}
```

```
    }  
    }  
}
```

Output:

```
Monday 1  
Tuesday 2  
Wednesday 3  
Thursday 4
```

## Java EnumMap Example: Book

```
import java.util.*;  
  
class Book {  
    int id;  
    String name,author,publisher;  
    int quantity;  
    public Book(int id, String name, String author, String publisher, int quantity) {  
        this.id = id;  
        this.name = name;  
        this.author = author;  
        this.publisher = publisher;  
        this.quantity = quantity;  
    }  
}  
  
public class EnumMapExample {  
    // Creating enum  
    public enum Key{  
        One, Two, Three  
    };  
    public static void main(String[] args) {  
        EnumMap<Key, Book> map = new EnumMap<Key, Book>(Key.class);  
        // Creating Books  
        Book b1=new Book(101,"Let us C","Yashwant Kanetkar","BPB",8);  
        Book b2=new Book(102,"Data Communications & Networking","Forouzan","Mc Graw Hill",4);  
        Book b3=new Book(103,"Operating System","Galvin","Wiley",6);  
        // Adding Books to Map  
        map.put(Key.One, b1);  
        map.put(Key.Two, b2);
```

```
map.put(Key.Three, b3);  
// Traversing EnumMap  
for(Map.Entry<Key, Book> entry:map.entrySet()){  
    Book b=entry.getValue();  
    System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);  
}  
}  
}
```

Output:

```
101 Let us C Yashwant Kanetkar BPB 8  
102 Data Communications & Networking Forouzan Mc Graw Hill 4  
103 Operating System Galvin Wiley 6
```

[< prev](#)[next >](#)

## Share this page



## Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces