Java 9 Factory Methods

Java 9 Collection library includes static factory methods for List, Set and Map interface. These methods are useful to create small number of collection.

Suppose, if we want to create a list of 5 elements, we need to write the following code.

Java List Example

```
import java.util.ArrayList;
import java.util.List;
public class FactoryMethodsExample {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("Java");
        list.add("JavaFX");
        list.add("Spring");
        list.add("Hibernate");
        list.add("JSP");
        for(String l : list){
            System.out.println(l);
        }
    }
}
```

Output:

```
Java
JavaFX
Spring
Hibernate
JSP
```

In the above code, add method is called **repeatedly** for each list element, while in Java 9 we can do it in single line of code using factory methods.

Factory Methods for Collection

Factory methods are special type of static methods that are used to create **unmodifiable instances** of collections. It means we can use these methods to create list, set and map of small number of elements.

It is unmodifiable, so adding new element will throw **java.lang.UnsupportedOperationException**.

Each interface has it's own factory methods, we are listing all the methods in the following tables.

Factory Methods of List Interface

Modifiers	Methods	Description
static <E> List<E>	Of()	It It returns an immutable list containing zero elements.
static <E> List<E>	of(E e1)	It It returns an immutable list containing one element.
static <E> List<E>	of(E... elements)	It It returns an immutable list containing an arbitrary number of elements.
static <E> List<E>	of(E e1, E e2)	It It returns an immutable list containing two elements.
static <E> List<E>	of(E e1, E e2, E e3)	It It returns an immutable list containing three elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4)	It It returns an immutable list containing four elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5)	It It returns an immutable list containing five elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5, E e6)	It It returns an immutable list containing six elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7)	It It returns an immutable list containing seven elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8)	It It returns an immutable list containing eight elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8, E e9)	It It returns an immutable list containing nine elements.
static <E> List<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8, E e9, E e10)	It It returns an immutable list containing ten elements.

Java 9 List Factory Method Example

In Java 9, we can write this code in very simple manner with the help of **List.of() factory method.**

```
import java.util.List;
public class FactoryMethodsExample {
    public static void main(String[] args) {
        List<String> list = List.of("Java","JavaFX","Spring","Hibernate","JSP");
        for(String l:list) {
            System.out.println(l);
        }
    }
}
```

Output:

```
Java
JavaFX
Spring
Hibernate
JSP
```

Java 9 Set Interface

Java Set interface provides a **Set.of() static factory method** which is used to create immutable set. The set instance created by this method has the following characteristics.

- It is immutable
- No null elements
- It is serializable if all elements are serializable.
- No duplicate elements.
- The iteration order of set elements is unspecified and is subject to change.

Java 9 Set Interface Factory Methods

The following table contains the factory methods for Set interface.

Modifier and Type	Method	Description
static <E> Set<E>	of()	It returns an immutable set containing zero elements.
static <E> Set<E>	of(E e1)	It It returns an immutable set containing one element.
static <E> Set<E>	of(E... elements)	It It returns an immutable set containing an arbitrary number of elements.
static <E> Set<E>	of(E e1, E e2)	It It returns an immutable set containing two elements.
static <E> Set<E>	of(E e1, E e2, E e3)	It It returns an immutable set containing three elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4)	It It returns an immutable set containing four elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5)	It It returns an immutable set containing five elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5, E e6)	It It returns an immutable set containing six elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7)	It It returns an immutable set containing seven elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8)	It It returns an immutable set containing eight elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8, E e9)	It It returns an immutable set containing nine elements.
static <E> Set<E>	of(E e1, E e2, E e3, E e4, E e5, E e6, E e7, E e8, E e9, E e10)	It It returns an immutable set containing ten elements.

Java 9 Set Interface Factory Methods Example

```
import java.util.Set;
public class FactoryMethodsExample {
    public static void main(String[] args) {
        Set<String> set = Set.of("Java","JavaFX","Spring","Hibernate","JSP");
        for(String l:set) {
            System.out.println(l);
        }
    }
}
```

Output:

```
Spring
JavaFX
JSP
Java
Hibernate
```

Java 9 Map Interface Factory Methods

In Java 9, Map includes Map.of() and Map.ofEntries() static factory methods that provide a convenient way to create immutable maps.

Map created by these methods has the following characteristics.

- It is immutable
- It does not allow null keys and values
- It is serializable if all keys and values are serializable
- It rejects duplicate keys at creation time
- The iteration order of mappings is unspecified and is subject to change.

Java 9 Map Interface Factory Methods

The following table contains the factory methods for Map interface.

Modifier and Type	Method	Description
static <K,V> Map<K,V>	of()	It returns an immutable map containing zero mappings.
static <K,V> Map<K,V>	of(K k1, V v1)	It returns an immutable map containing a single mapping.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2)	It returns an immutable map containing two mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3)	It returns an immutable map containing three mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4)	It returns an immutable map containing four mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5)	It returns an immutable map containing five mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5, K k6, V v6)	It returns an immutable map containing six mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5, K k6, V v6, K k7, V v7)	It returns an immutable map containing seven mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5, K k6, V v6, K k7, V v7, K k8, V v8)	It returns an immutable map containing eight mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5, K k6, V v6, K k7, V v7, K k8, V v8, K k9, V v9)	It returns an immutable map containing nine mappings.
static <K,V> Map<K,V>	of(K k1, V v1, K k2, V v2, K k3, V v3, K k4, V v4, K k5, V v5, K k6, V v6, K k7, V v7, K k8, V v8, K k9, V v9, K k10, V v10)	It returns an immutable map containing ten mappings.
static <K,V> Map<K,V>	ofEntries(Map.Entry<? extends K,? extends V>... entries)	It returns an immutable map containing keys and values extracted from the given entries.

Java 9 Map Interface Factory Methods Example

```
import java.util.Map;
public class FactoryMethodsExample {
    public static void main(String[] args) {
        Map<Integer,String> map = Map.of(101,"JavaFX",102,"Hibernate",103,"Spring MVC");
        for(Map.Entry<Integer, String> m : map.entrySet()){
            System.out.println(m.getKey()+" "+m.getValue());
        }
    }
}
```

Output:

```
101 Hibernate
103 Spring MVC
101 JavaFX
```

Java 9 Map Interface ofEntries() Method Example

In Java 9, apart from static **Map.of()** methods, Map interface includes one more static method **Map.ofEntries()**. This method is used to create a map of **Map.Entry** instances.

In the following example, we are creating map instance with the help of multiple map.entry instances.

```
import java.util.Map;
public class FactoryMethodsExample {
    public static void main(String[] args) {
        // Creating Map Entry
        Map.Entry<Integer, String> e1 = Map.entry(101, "Java");
        Map.Entry<Integer, String> e2 = Map.entry(102, "Spring");
        // Creating Map using map entries
        Map<Integer, String> map = Map.ofEntries(e1,e2);
        // Iterating Map
        for(Map.Entry<Integer, String> m : map.entrySet()){
            System.out.println(m.getKey()+" "+m.getValue());
        }
    }
}
```

Output:

```
101 Spring
101 Java
```