HowToDoInJava

Java Iterator interface example

Last Updated: April 10, 2020 By: Lokesh Gupta Java Collections Java Iterator

Java Iterator interface used to iterate over the elements in a collection (list, set or map). It helps to retrieve the specified collection elements one by one and perform operations over each element.

1. Java Iterator interface

All Java collection classes provide **iterator()** method which return the instance of **Iterator** to walk over the elements in that collection. For example, arraylist class **iterator()** method return an iterator over the elements in this list **in proper sequence**.

```
Iterator example

ArrayList<String> list = new ArrayList<>();

list.add("A");
list.add("B");
list.add("C");
list.add("D");

Iterator<String> iterator = list.iterator();

while(iterator.hasNext()) {
    System.out.println( iterator.next() );
}
```

Program Output.

```
Console
```

```
A
B
C
D
```

2. Java Iterator methods

2.1. Iterator hasNext()

- This method returns `true` if the iteration has more elements remaining in the collection.
- If iterator has gone over all elements then this method will return `false`.

2.2. Iterator next()

- This method returns the next element in the iteration.
- It throws **NoSuchElementException** if the iteration has no more elements.

2.3. Iterator remove()

- It removes from the underlying collection the last element returned by the iterator (optional operation).
- This method can be called only once per call to next().
- If the underlying collection is modified while the iteration is in progress in any way other than by calling `remove()` method, iterator will throw an **ConcurrentModificationException**.
- Iterators that do this are known as **fail-fast** iterators, as they fail quickly and cleanly, rather that risking arbitrary, non-deterministic behavior at an undetermined time in the future.

2.4. Iterator forEachRemaining()

- This method performs the given action for each remaining element until all elements have been processed or the action throws an exception.

- Actions are performed in the order of iteration, if that order is specified.
- It throws NullPointerException if the specified action is null.

3. Java Iterator example

3.1. ArrayList Iterate Example

Java example to iterate over ArrayList elements.

```
ArrayList Iterator example

ArrayList<String> list = new ArrayList<>();

list.add("A");
list.add("B");
list.add("C");
list.add("D");

System.out.println(list);

//Get iterator
Iterator<String> iterator = list.iterator();

//Iterate over all elements
while(iterator.hasNext())
{
    //Get current element
    String value = iterator.next();

    System.out.println( value );

    //Remove element
    if(value.equals("B")) {
        iterator.remove();
    }
}

System.out.println(list);
```

Program Output.

```
Console
```

```
[A, B, C, D]
A
B
C
D
[A, C, D]
```

3.2. HashSet Iterate Example

Iterating over a HashSet is very similar to iterate over a list. No visible differences.

```
HashSet Iterator example

HashSet<String> hashSet = new HashSet<>();

hashSet.add("A");
hashSet.add("B");
hashSet.add("C");
hashSet.add("D");

System.out.println(hashSet);

//Get iterator
Iterator<String> iterator = hashSet.iterator();

//Iterate over all elements
while(iterator.hasNext())
{
    //Get current element
    String value = iterator.next();

    System.out.println( value );

    //Remove element
    if(value.equals("B")) {
        iterator.remove();
    }
}

System.out.println(list);
```

Program Output.

```
Console

[A, B, C, D]
```

```
A
B
C
D
[A, C, D]
```

3.3. HashMap Keys Iterator Example

Java example to iterate over keys of a HashMap.

```
HashMap Keys Iterator example

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

map.put(1, "A");
map.put(2, "B");
map.put(3, "C");
map.put(4, "D");

System.out.println(map);

//Get iterator
Iterator<String> iterator = map.keys().iterator();

//Iterate over all keys
while(iterator.hasNext())
{
    String key = iterator.next();
    System.out.println( "Key : " + key + ", Value : " + map.get(key) );
}
```

Program Output.

```
Console

{1=A, 2=B, 3=C, 4=D}
Key : 1, Value : A
Key : 2, Value : B
Key : 3, Value : C
Key : 4, Value : D
```

3.4. HashMap Values Iterator Example

Java example to iterate over values of a HashMap.

```
HashMap Values Iterator example

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

map.put(1, "A");
map.put(2, "B");
map.put(3, "C");
map.put(4, "D");

System.out.println(map);

//Get iterator
Iterator<String> iterator = map.values().iterator();

//Iterate over all values
while(iterator.hasNext())
{
    System.out.println( "Value : " + iterator.next() );
}
```

Program Output.

```
Console

{1=A, 2=B, 3=C, 4=D}
Value : A
Value : B
Value : C
Value : D
```

3.5. Iterator forEachRemaining() Example

Java example to iterate over ArrayList elements and perform an action on them.

```
HashSet Iterator example

ArrayList<String> list = new ArrayList<>();

list.add("A");
list.add("B");
list.add("C");
list.add("D");

list.iterator().forEachRemaining( System.out::println );
```

Program Output.

```
Console

A
B
C
D
```

4. Convert Iterator to Stream

Convert iterator to stream by first converting iterator to `Spliterator` and then using `StreamSupport` to get stream from Spliterator.

```
IteratorToStream.java

import java.util.Arrays;
import java.util.Iterator;
import java.util.Spliterator;
import java.util.Spliterators;
import java.util.stream.Stream;
import java.util.stream.StreamSupport;

public class IteratorToStream
{
  public static void main(String[] args)
  {
    // Iterator
    Iterator<String> iterator = Arrays.asList("a", "b", "c")
                      .listIterator();
    //Extra step to get Spliterator
    Spliterator<String> splitItr = Spliterators
        .spliteratorUnknownSize(iterator, Spliterator.ORDERED);

    // Iterator -> Stream
    Stream<String> stream = StreamSupport.stream(splitItr, false);

    // Apply stream operations
    stream.forEach(System.out::println);
  }
}
```

Program output:

```
Console

a
b
c
```

5. Conclusion

In this tutorial, we learned the **Java Iterator interface**. We learned the iterator methods and simple examples to iterate over different collections such as `List`, `Set` and `Map`.

Drop me your questions in the comments section.

Happy Learning !!

References:

Iterator Interface Java Docs

Recommended Reading:

1. How Iterator works in java
2. Difference between enumerator and iterator?
3. Convert Iterable or Iterator to Stream in Java

4. Java Read XML with StAX Parser – Cursor & Iterator APIs
5. Iterator Design Pattern
6. Java ListIterator interface
7. Java Spliterator interface
8. Java Comparable Interface
9. Java Comparator Interface
0. Java Cloneable interface – Is it broken?

Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

** We do not spam !!*

2 thoughts on “Java Iterator interface example”

sai vijay

February 13, 2019 at 10:06 pm

Seems, HashMap Keys Iterator Example is having same example of HashSet, Please check

Reply

Lokesh Gupta

February 14, 2019 at 12:39 pm

Hi Vijay, Thanks for noticing and reporting.

Reply

Leave a Comment

Name *

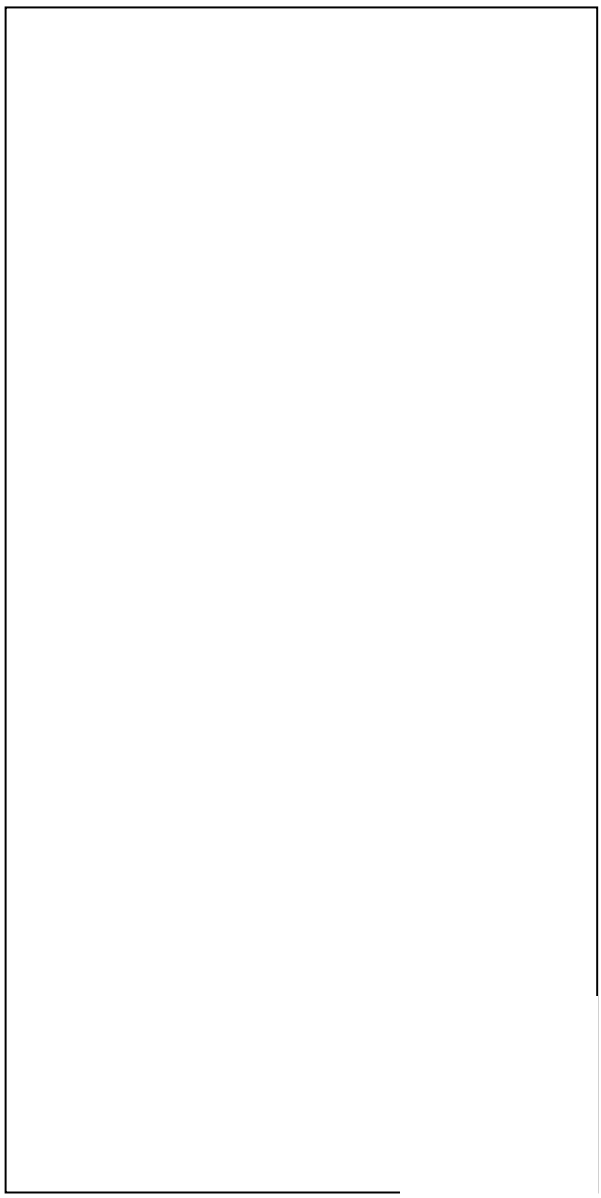
Email *

Website

Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment

Search …



HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- About Me
- Contact Us
- Privacy policy
- Advertise

Guest Posts

Blogs

REST API Tutorial