

| |
|--|
| Algo ▾ |
| DS ▾ |
| Languages ▾ |
| Interview ▾ |
| Students ▾ |
| GATE ▾ |
| CS Subjects ▾ |
| Quizzes ▾ |
| GBlog |
| Puzzles |
| Practice |
| List Interface in Java with Examples |
| AbstractList in Java with Examples |
| ArrayList in Java |
| LinkedList in Java |
| Immutable List in Java |
| Custom ArrayList in Java |
| Java Collection Difference between Synchronized ArrayList and CopyOnWriteArrayList |
| How to remove a SubList from a List in Java |
| Randomly select items from a List in Java |
| Get first and last elements from ArrayList in Java Split a list into two halves in Java |
| How to Remove Duplicates from ArrayList in Java |
| How to get ArrayList from Stream in Java 8 |
| How to convert LinkedList to Array in Java? |
| How to make an ArrayList read only in Java |
| Join two ArrayLists in Java |
| Find common elements in two ArrayLists in Java |
| Find first and last element of ArrayList in java |
| Convert an Iterator to a List in Java |
| Polymorphism in Java |
| Maximum absolute difference in an array |
| What's the connection between Java and Blockchain? |
| Difference between static and non-static method in Java |
| Spring Dependency Injection with Example |
| Difference between static and non-static variables in Java |
| Count occurrences of a given character using Regex in Java |
| Message Passing in Java |
| File Handling in Java with CRUD operations |
| What is the Role of Java in the IT Industry? |
| BigInteger multiply() Method in Java with Examples |



Classroom program in MOOD Starting from 21st September 2019

Machine Learning

FOUNDATION WITH PYTHON

₹10,999

Register Now

2nd Batch

Mentored by Works in ADOR Industry Expert Dr. ANAND KUMAR

Good for foundations Machine Learning - Rohit Ray

The course was excellent. All basic topics of ML are covered. - ANMOL SHARMA

CopyOnWriteArrayList in java

CopyOnWriteArrayList: CopyOnWriteArrayList class is introduced in JDK 1.5, which implements List interface. It is enhanced version of ArrayList in which all modifications (add, set, remove, etc) are implemented by making a fresh copy.



Fig: CopyOnWriteArrayList

Here are few points about CopyOnWriteArrayList:

- As the name indicates, CopyOnWriteArrayList creates a Cloned copy of underlying ArrayList, for every update operation at certain point both will be synchronized automatically ,which is taken care by JVM. Therefore there is no effect for threads which are performing read operation CopyOnWriteArrayList is the best choice if our frequent operation is read operation.
- It extends Object and implements Serializable, Cloneable, Iterable<E>, Collection<E>, List<E> and RandomAccess
- The underlined data structure is grow-able array.
- It is thread-safe version of ArrayList.
- Insertion is preserved, duplicates are allowed and heterogeneous Objects are allowed.
- The main important point about CopyOnWriteArrayList is Iterator of CopyOnWriteArrayList can not perform remove operation otherwise we get Run-time exception saying UnsupportedOperationException.

Constructors Summary:

- CopyOnWriteArrayList c = new CopyOnWriteArrayList();** :Creates an empty list.
- CopyOnWriteArrayList c = new CopyOnWriteArrayList(Collection obj);**:Creates a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.
- CopyOnWriteArrayList c = new CopyOnWriteArrayList(Object[] obj);**:Creates a list holding a copy of the given array.

Methods Summary:

- add(int index, E element):** This method Inserts the specified element at the specified position in this list.
- add(E e):** This method Appends the specified element to the end of this list.
- addAll(int index, Collection<E> c):** This method Inserts all of the elements in the specified collection into this list, starting at the specified position.
- addAll(Collection<E> c):** This method Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator.
- addAllAbsent(Collection<E> c):** This method Appends all of the elements in the specified collection that are not already contained in this list, to the end of this list, in the order that they are returned by the specified collection's iterator.
- clearAbsent(E e):** This method Removes the element, if not present.
- clear():** This method Removes all of the elements from this list.
- clone():** This method Returns a shallow copy of this list.
- contains(Object o):** This method Returns true if this list contains the specified element.
- containsAll(Collection<E> c):** This method Returns true if this list contains all of the elements of the specified collection.
- equals(Object o):** This method Compares the specified object with this list for equality.
- forEach(Consumer<E> action):** This method Performs the given action for each element of the Iterable until all elements have been processed or the action throws an exception.
- get(int index):** This method Returns the element at the specified position in this list.
- hashCode():** This method Returns the hash code value for this list.
- indexOf(E e, int index):** This method Returns the index of the first occurrence of the specified element in this list, searching backwards from index, or returns -1 if the element is not found.
- indexOf(Object o):** This method Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
- isEmpty():** This method Returns true if this list contains no elements.
- iterator():** This method Returns an iterator over the elements in this list in proper sequence.
- lastIndexOf(E e, int index):** This method Returns the index of the last occurrence of the specified element in this list, searching backwards from index, or returns -1 if the element is not found.
- lastIndexOf(Object o):** This method Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
- listIterator():** This method Returns a list iterator over the elements in this list (in proper sequence).
- listIterator(int index):** This method Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list.
- remove(int index):** This method Removes the element at the specified position in this list.
- remove(Object o):** This method Removes the first occurrence of the specified element from this list, if it is present.
- removeAll(Collection<E> c):** This method Removes from this list all of its elements that are contained in the specified collection.
- removeIf(Predicate<E> filter):** This method Removes all of the elements of this collection that satisfy the given predicate.
- replaceAll(UnaryOperator<E> operator):** This method Replaces each element of this list with the result of applying the operator to that element.
- retainAll(Collection<E> c):** This method Retains only the elements in this list that are contained in the specified collection.
- set(int index, E element):** This method Replaces the element at the specified position in this list with the specified element.
- size():** This method Returns the number of elements in this list.
- sort(Comparator<E> c):** This method Sorts this list according to the order induced by the specified Comparator.
- splitIterator():** This method Returns a SplitIterator over the elements in this list.
- subList(int fromIndex, int toIndex):** This method Returns a view of the portion of this list between fromIndex, inclusive, and toIndex, exclusive.
- toArray():** This method Returns an array containing all of the elements in this list in proper sequence (from first to last element).
- toArray(T[] a):** This method Returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.
- toString():** This method Returns a string representation of this list.

```
// Java program to illustrate
// CopyOnWriteArrayList class
import java.util.concurrent.CopyOnWriteArrayList;
import java.util.*;

class ConcurrentDemo extends Thread {

    static CopyOnWriteArrayList l = new CopyOnWriteArrayList();

    public void run()
    {
        // Child thread trying to
        // add new element in the
        // Collection object
        l.add("D");
    }

    public static void main(String[] args)
        throws InterruptedException
    {
        l.add("A");
        l.add("B");
        l.add("C");

        // We create a child thread
        // that is going to modify
        // ArrayList l.
        ConcurrentDemo t = new ConcurrentDemo();
        t.run();

        Thread.sleep(1000);

        // Now we iterate through
        // the ArrayList and get
        // exception.
        Iterator itr = l.iterator();
        while (itr.hasNext()) {
            String s = (String)itr.next();
            System.out.println(s);
            Thread.sleep(1000);
        }
        System.out.println(1);
    }
}
```

Output:

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

Recommended Posts:

- CopyOnWriteArrayList get() method in Java
- CopyOnWriteArrayList add() method in Java
- CopyOnWriteArrayList contains() method in Java
- CopyOnWriteArrayList hashCode() method in Java
- CopyOnWriteArrayList clear() method in Java
- CopyOnWriteArrayList set() method in Java with Examples
- CopyOnWriteArrayList toArray() method in Java
- CopyOnWriteArrayList clone() method in Java
- CopyOnWriteArrayList containsAll() method in Java
- CopyOnWriteArrayList size() method in Java
- CopyOnWriteArrayList addIfAbsent() method in Java
- CopyOnWriteArrayList listIterator() method in Java
- CopyOnWriteArrayList iterator() method in Java
- CopyOnWriteArrayList lastIndexOf() method in Java
- CopyOnWriteArrayList toString() method in Java

Bishal Kumar Dubey
Hello everyone, I am Bishal KUMAR Dubey and most importantly an idea creator I just love Programming languages and love to know new concepts everyday, every minute, every second Here to help Other GEEKS

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : vinaykumar54, randhish79, rahulcool dude05

Transform Any Aspect Of Your Life

How To Get More Health, Wealth, Love, Inner Peace and Clarity.

FREE MASTERCLASS

Article Tags : [Java](#) [Java - util package](#) [Java-ArrayList](#) [Java-Collections](#) [Java-CopyOnWriteArrayList](#) [java-list](#)

Practice Tags : [Java](#) [Java-Collections](#)

2

☐ To-do ☐ Done

0

No votes yet.

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Previous [How to measure time taken by a function in java ?](#) Next [Difference between ArrayList and CopyOnWriteArrayList](#)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Google AdMob

अपने ऐप्लिकेशन के विज्ञापन से होने वाली आय को और बढ़ाएं.

अभी शुरू करें

1 IN EVERY 15 CS STUDENTS OPTED FOR DSA SELF-PACED COURSE

STUDENTS PLACED IN TOP PRODUCT BASED COMPANIES LIKE AMAZON MICROSOFT GOOGLE ADOBE SWIGGY

WHAT ARE YOU WAITING FOR?

₹2,499

Register Now

20000+ ENROLLMENTS SO FAR

Corporate Training

Corporate Trainers | Class Recordings | Lifetime Membership | 100% Money Back Guarantee Interview Gyan

| |
|--|
| Most popular in Java |
| Java program to check whether a string is a Palindrome |
| Recursion in Java |
| Differences between Interface and Class in Java |
| BigInteger add() Method in Java with Examples |
| Difference between concat() and + operator in Java |

IntelliPaat

Be a Certified CLOUD EXPERT

Accredited By IBM, MICROSOFT

ENROLL NOW

splunk>

| |
|--|
| More related articles in Java |
| Difference between String and Character array in Java |
| ClassLoader in Java |
| Count the pairs of vowels in the given string |
| Print characters and their frequencies in order of occurrence using a Linked-hashMap in Java |
| Difference between while and do-while loop in C, C++, Java |

Starting from 1st September 2019

SDE TEST SERIES

₹599 ₹199

Register Now

ONE TIME OFFER

Test Series Features

5 Contest Video Solutions Job Opportunities