Recommended Courses

- best python courses
- best java courses
- system design courses
- best spring courses
- best hibernate courses
- best design pattern courses
- best Linux courses
- best JavaScript courses
- best data structure and algorithms courses
- Best Multithreading Courses
- best MERN stack courses
- Best Git courses
- Best Microservice Courses
- Best DevOps Courses
- best MEAN stack Courses
- free Java courses
- free DSA courses
- free sql courses
- free Linux courses
- Free Docker courses
- free JUnit courses

OOP Tutorials

- oop - polymorphism
- oop - object
- oop - inheritance
- oop - free courses
- oop - class vs interface
- oop - inheritance questions
- oop- interview questions
- oop - class vs object
- oop - overriding

How to Remove Objects From ArrayList while Iterating in Java - Example Tutorial

One of the common problems many Java Programmers face is to **remove elements while iterating over ArrayList in Java** because the intuitive solution doesn't work like you just cannot go through an ArrayList using a for loop and remove an element depending upon some condition. Even though `java.util.ArrayList` provides the `remove()` methods, like `remove (int index)` and `remove (Object element)`, you cannot use them to remove items while iterating over ArrayList in Java because they will throw ConcurrentModificationException if called during iteration. The right way to remove objects from `ArrayList` while iterating over it is by using the *Iterator's* `remove()` method.

When you use iterator's `remove()` method, `ConcurrentModfiicationException` is not thrown. Because it also updates the counters and variables used by the Iterator like `modCount`, which indicates that modification is done by the Iterator itself and not somewhere around.

In this article, I'll show you an example of both ways and how they work in Java. You'll also learn a little bit about `java.util.ConcurrentModificationException`, which is a common problem for non-concurrent collection classes like `ArrayList` or `HashMap`.

Though, if you are entirely new to Java and coming from a non-programming background, I suggest you to first go through a comprehensive Java course like **The Complete Java MasterClass** instead of learning from arbitrary articles.

The course provides structured learning, which is both efficient and works great for beginners. Once you know the fundamentals, you can learn any topic by reading a blog post or individual tutorial, they will make more sense by then also.

- oop - abstraction
- oop - encpasulation
- oop - abstract class vs interface
- oop - method
- oop - constructor chaining
- oop - static method
- oop - java
- oop - abstract class
- oop - overloading vs overriding
- oop - overloading
- oop - interface
- oop - constructor

Array Tutorials

- array - copy
- array - tutorial
- array - add/remove element
- array - linked list
- array - reverse
- array - sorting
- array - sum
- array - binary search
- array - vector
- array - remove
- array - reverse in place
- array - to list
- array - initialization
- array - insertion sort
- array - to string
- array - example
- array - data structure
- array - compare
- array - liner search

ArrayList Tutorials

- arraylist - sort
- arraylist - stream
- arraylist - remove
- arraylist - replace
- arraylist - join
- arraylist - size
- arraylist - removeAll
- arraylist - iterator
- arraylist - to set
- arraylist - hashmap
- arraylist - sort descending
- arraylist - guide
- arraylist - with values
- arraylist - concurrent

1. ArrayList remove() method Example

Now, let's see an example of removing elements from ArrayList while looping using `for()` loop and `ArrayList.remove()` method, which is wrong, and the program will throw `ConcurrentModificationExcetpion` upon execution.

```
import java.util.ArrayList;
import java.util.List;

/*
 * Java Program to remove an element while iterating over ArrayList
 */

public class Main {

  public static void main(String[] args) throws Exception {

    List<String> loans = new ArrayList<>();
    loans.add("personal loan");
    loans.add("home loan");
    loans.add("auto loan");
    loans.add("credit line loan");
    loans.add("mortgage loan");
    loans.add("gold loan");

    // printing ArrayList before removing any element
    System.out.println(loans);

    // removing element using ArrayList's remove method during iteration
    // This will throw ConcurrentModification

    for (String loan : loans) {
      if (loan.equals("personal loan")) {
        loans.remove(loan);
      }
    }

    // printing ArrayList after removing an element
    System.out.println(loans);
  }
```


Interv

- co
- SC
- da
- co
- jav
- jav qu
- thr
- hib
- j2e
- Sp
- ob qu

Recor

- 5 I Be
- 5 I Bc
- 1C Be
- 1C Sh
- 1C Sh
- 1C Fr
- 1C Le

modification
- arraylist - to array
- arraylist - first element
- arraylist - reverse
- arraylist - synchronized
- arraylist - copyOnWriteArrayList

Blog Archive

▶ 2023 (417)
▶ 2022 (374)
▼ 2021 (446)
▶ December (43)
▶ November (41)
▶ October (26)
▶ September (34)
▼ August (65)
Java 8 Stream filter() + findFirst Example Tutorial
Top 22 Array Concepts Interview Questions Answers ...
How to add and remove Elements from an Dynamic Arr...
How to Remove Objects From ArrayList while Iterati...
9 JSP Implicit Objects and When to Use Them
How to implement PreOrder traversal of Binary Tree...
How to reverse a singly linked list in Java withou...
How to use Randam vs ThreadLocalRandom vs SecureRa...

```
}
```

Output

```
Exception in thread "main" [personal loan, home loan, auto loan,
credit line loan, mortgage loan, gold loan]
java.util.ConcurrentModificationException
at
java.util.ArrayList$Itr.checkForComodification(ArrayList.java:859)
```



```
at java.util.ArrayList$Itr.next(ArrayList.java:831)
at Main.main(Main.java:26)
```

Some of you may wonder that we are getting `ConcurrentModificationException` because we are not using the Iterator, but that's not true, even if you use Iterator you will get `java.util.ConcurrentModificationException` as long as you will use ArrayList's remove() method for removing element while iterating as shown in the following example:

```
Iterator<String> itr = loans.iterator();
    while (itr.hasNext()) {
      String loan = itr.next();
      if (loan.equals("personal loan")) {
        loans.remove(loan);
      }
    }
```

```
Exception in thread "main" java.util.ConcurrentModificationException
at
java.util.ArrayList$Itr.checkForComodification(ArrayList.java:859)
at java.util.ArrayList$Itr.next(ArrayList.java:831)
at Main.main(Main.java:29)
```

In order to fix the above code, **you just need to remove the `loans.remove(loan)` with the `itr.remove()` method**, which is explained in the next example. Though, if you want to know more about Iterator and in general Java Collection Framework, which can feel daunting sometimes, I suggest you go through **Java Fundamentals: Collections** course on

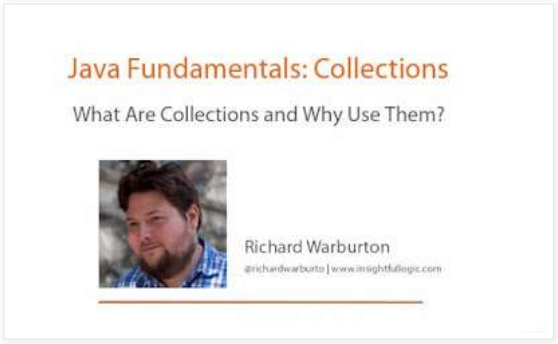
- 10 Sh
- 5
- 5 Qu
- 10 Al

Books
- Be Be
- 5 ar
- 2 Ja
- 12 Pr Ex
- 5 do
- 5 Sk
- Bo Sh
- To Bo
- 10 Bo
- 5
- Bo Sh
- To Bo

Follov
Follov
Foll
Privac

How to implement Linear Search Algorithm in Java? ...
How to Convert String to LocalDateTime in Java 8 -...
How to use Stream.filter method in Java 8? Example...
How to reverse bits of an integer in Java? [LeetCo...
10 Examples of Stream API in Java 8 - count + filt...
How to debug Java 8 Stream Pipeline - peek() metho...
Can you make an Abstract Class or Method Final in ...
How to Calculate Next Date and Year in Java | Loca...
How to Reverse an Array in place in Java? Example ...
3 ways to ignore null fields while converting Java...
5 Differences between an array and linked list in ...
What is difference between final vs finally and fi...
How to convert ArrayList to HashMap and

Pluralsight.



It's a perfect course to both learn and master the Java Collection framework and I highly recommend you to join this course.

2. Iterator's remove() method Example

Now, let's try the other approach, which uses Iterator's `remove()` method to remove an element from ArrayList while iterating over it.

```
import java.util.ArrayList;

import java.util.Iterator;
import java.util.List;

/*
 * Java Program to remove an element while iterating over ArrayList
 */

public class Main {

  public static void main(String[] args) throws Exception {

    List<String> loans = new ArrayList<>();
    loans.add("personal loan");
    loans.add("home loan");
    loans.add("auto loan");
    loans.add("credit line loan");
    loans.add("mortgage loan");
    loans.add("gold loan");

    // printing ArrayList before removing any element
    System.out.println(loans);
```

LinkedHash ...
How to code Binary Search Algorithm using Recursio...
Post Order Binary Tree Traversal in Java Without R...
7 Examples to Sort One and Two Dimensiona l String ...
How to prepare Oracle Certified Master Java EE Ent...
How to use Record in Java? Example
How to Check is given String is a Palindrome in Ja...
Top 5 Free Apache Maven eBooks for Java Developers
Top 4 Free Microsoft SQL Server Books - PDF Downlo...
How to check if strings are rotations of each othe...
4 Best Books to Learn Web Service in Java - SOAP a...
What is the cost of Oracle Java Certification s - O...
Top 3 Free Struts Books for Java EE developers - L...
Java Program to find Armstrong numbers

```
    // removing element using ArrayList's remove method during iteration
    // This will throw ConcurrentModification

    Iterator<String> itr = loans.iterator();
    while (itr.hasNext()) {
      String loan = itr.next();
      if (loan.equals("personal loan")) {
        itr.remove();
      }
    }

    // printing ArrayList after removing an element
    System.out.println(loans);
  }

}
```

Output

```
[personal loan, home loan, auto loan, credit line loan, mortgage loan, gold loan]
[home loan, auto loan, credit line loan, mortgage loan, gold loan]
```

From the output, you can see that the "personal loan" element is removed from the ArrayList. The size of the ArrayList is also reduced by one, and there is no `ConcurrentModficiationException` in the code.

That's all about **how to remove elements while iterating over ArrayList in Java**. As I have said that if you use ArrayList's `remove()` method like `remove(int index)` or `remove(Object obj)` while iterating over `ArrayList`, then a `ConcurrentModfiicationException` will be thrown. You can avoid that by using Iterator's `remove()` method, which removes the current object in the iteration.

Other **ArrayList tutorials** for Java Programmers

- How to remove duplicate elements from ArrayList in Java? (tutorial)
- 10 Courses to learn Java Programming for beginners (courses)
- How to create and initialize ArrayList in the same line? (example)
- 10 Free Spring Boot Courses for Java developers (courses)
- Difference between ArrayList and HashSet in Java? (answer)
- 10 Free Courses to learn Spring Framework (courses)
- Difference between an Array and ArrayList in Java? (answer)
- How to reverse an ArrayList in Java? (example)
- Top 5 Courses to learn Hibernate for Java developers (courses)
- How to loop through an ArrayList in Java? (tutorial)
- How to synchronize an ArrayList in Java? (read)
- When to use ArrayList over LinkedList in Java? (answer)
- Difference between ArrayList and HashMap in Java? (answer)
- Difference between ArrayList and Vector in Java? (answer)
- How to sort an ArrayList in descending order in Java? (read)
- How to get a sublist from ArrayList in Java? (example)
- How to convert CSV String to ArrayList in Java? (tutorial)

with Example

Insertion Sort Algorithm in Java with Example

Difference between VARCHAR and NVARCHAR in SQL Server

How to Rotate an Array to Left or Right in Java? S...

How to implement Merge Sort Algorithm in Java [So...

What is Variable and Method Hiding in Java - Examp...

How to implement Comparator and Comparable in Java...

Difference between List <?> and List<Object> in J...

Java - Difference between getClass() and instanceo...

How to Read or Parse CSV files with Header in Java...

Java 8 StringJoiner Example - How to join multiple...

Can You Run Java Program Without a Main Method? [I...

How to Remove an Element from Array in Java with E...

Thanks for reading this article so far. If you like this article, then please share it with your friends and colleagues. If you have any questions or feedback, then please drop a note

P. S. - If you are new to the Java world and looking for a free online training course to learn Java then you can also check out this **Java Tutorial for Complete Beginners** course on Udemy. It's completely free and you just need a Udemy account to join this online course.

Labels: ArrayList, core java, Java collection tutorial

5 comments:

Anonymous May 14, 2020 at 5:41 PM

```
while (itr.hasNext()) {
String loan = itr.next();
if (loan.equals("personal loan")) {
itr.remove();
}
}
```

Doesn't this while loop fail to check the first element in the sequence, since it's always checking the next?

Reply

Replies

javin paul May 15, 2020 at 4:48 AM

When @Annonymous, did you tested that? hasNext() return true if there is more element and itr.next() is required to move the pointer. I suggest you test and then we can discuss.

Reply

AA July 17, 2020 at 11:20 AM

Why I am not getting the exception in both cases....
Still wondering why, what I have coded wrong that I am missing the exception.

Reply

Unknown December 7, 2020 at 9:03 AM

```
public class Main {

public static void main(String[] args) {
Collection students = new ArrayList();

for (int i=0; i< 4;i++){
Student student = new Student();
student.name = "Lavanya"+i;
student.age = 20 + i;

students.add(student);
}

System.out.println("Before performing operations.....");
for (Student student : students) {
System.out.println(String.format("%s -> %d",student.name, student.age));
}
```

How to remove duplicates from Collections or Strea...

How to Reverse String in Java with or without Stri...

What is double colon (::) operator in Java 8 - Exa...

How to copy Array in Java? Arrays copyOf and copyO...

How to Join Multiple Strings in Java 8 - String jo...

How to Find Nth Fibonacci Number in Java [Solved] ...

How to Find Square Root of a Number in Java [Solve...

How to implement Binary Tree PreOrder Traversal in...

Java 8 Stream map() function Example with Explanation

Top 21 Java HashMap Interview Questions and Answers

Top 15 Java 8 Stream and Functional Programming In...

Bubble sort in Java - Program to sort an Integer A...

How to Reverse an Integer in Java

```
Iterator iterator = students.iterator();
if (iterator.hasNext()){
var student = iterator.next();
if ( student.age == 21){
iterator.remove();
}
}
System.out.println();
System.out.println("After performing remove operations.....");
for (Student student : students) {
System.out.println(String.format("%s -> %d",student.name, student.age));
}
}
```

Whrere Student class has two properties name and age.
In output i am able to see the student who has age 21.

output:

Before performing operations.....
Lavanya0 -> 20
Lavanya1 -> 21
Lavanya2 -> 22
Lavanya3 -> 23

After performing remove operations.....
Lavanya0 -> 20
Lavanya1 -> 21
Lavanya2 -> 22
Lavanya3 -> 23

Could you please explain.....

Reply

Replies

javin paul December 7, 2020 at 7:26 PM

You are using if instead of while, this means you are only checking for first student which has age 20 that's why no student is get deleted. change the code like this and it should work

```
while (iterator.hasNext()){
var student = iterator.next();
if ( student.age == 21){
iterator.remove();
}
```

Reply

Enter Comment

Feel free to comment, ask questions if you have any doubt.

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)