Fail-Fast Iterator

Before understanding Fail-Fast Iterator you need to understand [ConcurrentModificationException](http://data-structure-learning.blogspot.com/2015/05/concurrentmodificationexception.html).

If you are at this line that I am assuming that you have read about the ConcurrentModificationException. Fail-Fast iterators throw ConcurrentModificationException if the [Collection](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-4collection.html) is modified structurally. Fail-Fast iterators fail quickly and cleanly rather than risking the arbitrary non-deterministic behavior at undetermined time in future.

Now Fail-Fast iterators can throw ConcurrentModificationException in two situations:

Single threaded environment: Once iterator is created and you don’t use remove method of iterator to remove element from Collection. Also if you use add method while iterating over collection then also ConcurrentModificationException is thrown.

Multithreaded Environment: If one thread is iterating over the collection and another thread modifies the collection structurally.

This leads to one more question: How does the iterator comes to know that internal structure is modified?

There is a flag value called modCount. Whenever the Collection is modified structurally the modCount is incremented. next() method and remove() method of Iterator calls checkForComodification() method which checks modCount and expectedModCount are same or not. If not then ConcurrentModificationException is thrown.

Below program throws ConcurrentModificationException.

**package** org.collections;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Iterator;

**public** **class** ArrayListExample {

**private** ArrayList<Integer> list;

**public** ArrayListExample() {

list = **new** ArrayList<Integer>(Arrays.*asList*(1, 2, 3, 4, 5));

}

**public** **void** throwConcurrentModificationException() {

Iterator<Integer> iter = list.iterator();

**while**(iter.hasNext()){

iter.next();

list.remove(2);

}

}

**public** **static** **void** main(String[] args) {

ArrayListExample a=**new** ArrayListExample();

a.throwConcurrentModificationException();

}

}

Exception in thread "main" java.util.**ConcurrentModificationException**

at java.util.ArrayList$Itr.checkForComodification(Unknown Source)

at java.util.ArrayList$Itr.next(Unknown Source)

at org.collections.ArrayListExample.throwConcurrentModificationException(ArrayListExample.java:19)

at org.collections.ArrayListExample.main(ArrayListExample.java:27)