It’s an exception when we work with Java collection classes. The error occurs when you are iterating and modifying the same code. Also, it is not permissible for one thread to modify a Collection while another thread is iterating over it. Iterators that do this are called *fail-fast* iterators.

Thus, the *iterator.next()* throws ***ConcurrentModificationException****..*

This exception can occur for both multithread and single thread.

**WHAT WE ARE DOIN WORNG?**

//Making changes on the ongoing thread while other is iterating over

for(String str : map.get(0)){

map.get(0).add("aparna");

}

**HOW TO FIX IT?**

Use ArrayList to avoid the exception.

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

for (String str : map.get(0)) {

list.add("aparna");

}

1. Convert the list to an array and then, you iterate on the array.
2. Avoid adding objects because it can be processed depending on the keyset.
3. If working on single-threaded environment and want your code to take care of the extra added objects in the list then you can do so using for loop rather than an iterator.

EXAMPLE

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

public class ConcurrentModificationExceptionExample {

public static void main(String args[]) {

List<String> myList = new ArrayList<String>();

myList.add("1");

myList.add("2");

myList.add("3");

myList.add("4");

myList.add("5");

Iterator<String> it = myList.iterator();

while (it.hasNext()) {

String value = it.next();

System.out.println("List Value:" + value);

if (value.equals("3"))

myList.remove(value);

}

Map<String, String> myMap = new HashMap<String, String>();

myMap.put("1", "1");

myMap.put("2", "2");

myMap.put("3", "3");

Iterator<String> it1 = myMap.keySet().iterator();

while (it1.hasNext()) {

String key = it1.next();

System.out.println("Map Value:" + myMap.get(key));

if (key.equals("2")) {

myMap.put("1", "4");

// myMap.put("4", "4");

}

}

}

}

**FIXED CODE**

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.concurrent.ConcurrentHashMap;

import java.util.concurrent.OnWriteArrayList;

public class AvoidConcurrentModificationException {

public static void main(String[] args) {

List<String> myList = new OnWriteArrayList<String>();

myList.add("1");

myList.add("2");

myList.add("3");

myList.add("4");

myList.add("5");

Iterator<String> it = myList.iterator();

while (it.hasNext()) {

String value = it.next();

System.out.println("List Value:" + value);

if (value.equals("3")) {

myList.remove("4");

myList.add("6");

myList.add("7");

}

}

System.out.println("List Size:" + myList.size());

Map<String, String> myMap = new ConcurrentHashMap<String,String>();

myMap.put("1", "1");

myMap.put("2", "2");

myMap.put("3", "3");

Iterator<String> it1 = myMap.keySet().iterator();

while (it1.hasNext()) {

String key = it1.next();

System.out.println("Map Value:" + myMap.get(key));

if (key.equals("1")) {

myMap.remove("3");

myMap.put("4", "4");

myMap.put("5", "5");

}

}

System.out.println("Map Size:" + myMap.size());

}

}