

shuffle loses/duplicates objects in LinkedList

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.help/2005-01/0698.html>

From: Adrian Collister (*aac27_at_cam.ac.uk*)

Date: 01/19/05

Date: Wed, 19 Jan 2005 19:51:55 +0000

I'm trying to use the `Collections.shuffle()` method on a `LinkedList`. After the `shuffle()` the `LinkedList` is totally mangled, with several of the original Objects being replaced by duplicates of others. If I use an `ArrayList` instead of the `LinkedList` the behaviour is as expected.

Am I doing something wrong, or is this a bug?

The appended code demonstrates the problem. Example output when I execute it:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
0 3 7 0 6 13 4 1 12 15 11 6 14 2 7 3 17 14 17 8
0 0 1 2 3 3 4 6 6 7 7 8 11 12 13 14 14 15 17 17
```

I'm using the GCC java compiler – might be interesting to see what happens if it's compiled using Sun's compiler.

Adrian

```
/*----- CODE -----*/
```

```
import java.util.List;
import java.util.LinkedList;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Collections;

public class ShuffleTest
{

    public static void main(String[] args)
    {
        // Fill a list with "Int" objects (see below)
        LinkedList x = new LinkedList();
        for (int i = 0; i < 20; ++i) {
            x.add(new Int(i));
        }
    }
}
```

comp.lang.java.help: shuffle loses/duplicates objects in LinkedList

```
// Shuffle then re-sort...
printList(x);
Collections.shuffle(x);
printList(x);
Collections.sort(x);
printList(x);
}

// Dump list to STDOUT.
public static void printList(List l)
{
    Iterator it = l.iterator();
    while (it.hasNext()) {
        System.out.print(((Int)it.next()).value + " ");
    }
    System.out.println();
}

// Simple Class which just contains an int.
public class Int implements Comparable
{
    int value;

    // Constructor
    public Int(int v)
    {
        this.value = v;
    }

    public int compareTo(Object o)
```