

Difference between List and Vector

Contents

Overview	2
Difference between List and Vector	3
✓1. Thread Safety	3
✓2. Synchronization.....	3
✓3. Performance.....	3
Summary Table:	4
Example:.....	4

Overview

This document is intended to provide difference between List and Vector in java.

Difference between List and Vector

✓1. Thread Safety

- **List** (specifically `ArrayList` and other non-thread-safe implementations):
 - Not thread-safe by default.
 - If multiple threads access it concurrently, and at least one modifies it, external synchronization is needed.
- **Vector**:
 - Thread-safe.
 - All methods are synchronized, making it safe for use in multithreaded environments **without additional synchronization**.

✓2. Synchronization

- **List**:
 - Not synchronized.
 - If thread safety is required, developers must wrap the list using:

```
List<String> syncList = Collections.synchronizedList(new ArrayList<>());
```
- **Vector**:
 - Synchronized internally.
 - Every method is synchronized, meaning only one thread can access the method at a time.

✓3. Performance

- **List** (like `ArrayList`):
 - Generally **faster** than `Vector` in single-threaded or read-heavy environments.
 - No synchronization overhead.
- **Vector**:
 - Slower compared to `ArrayList` due to the cost of synchronizing each method.
 - Can become a performance bottleneck in high-concurrency scenarios.

Summary Table:

Feature	List(ArrayList)	Set
Thread-Safety	Not thread-safe	Thread-safe
Synchronization	Requires external sync	Built-in (method level)
Performance	Faster (no sync overhead)	Slower (due to sync)

Example:

Java Code

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

public class ListVectorExample {

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

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

        Vector<String> vector = new Vector<>();

        List<String> syncArrayList = Collections.synchronizedList(new ArrayList<>());

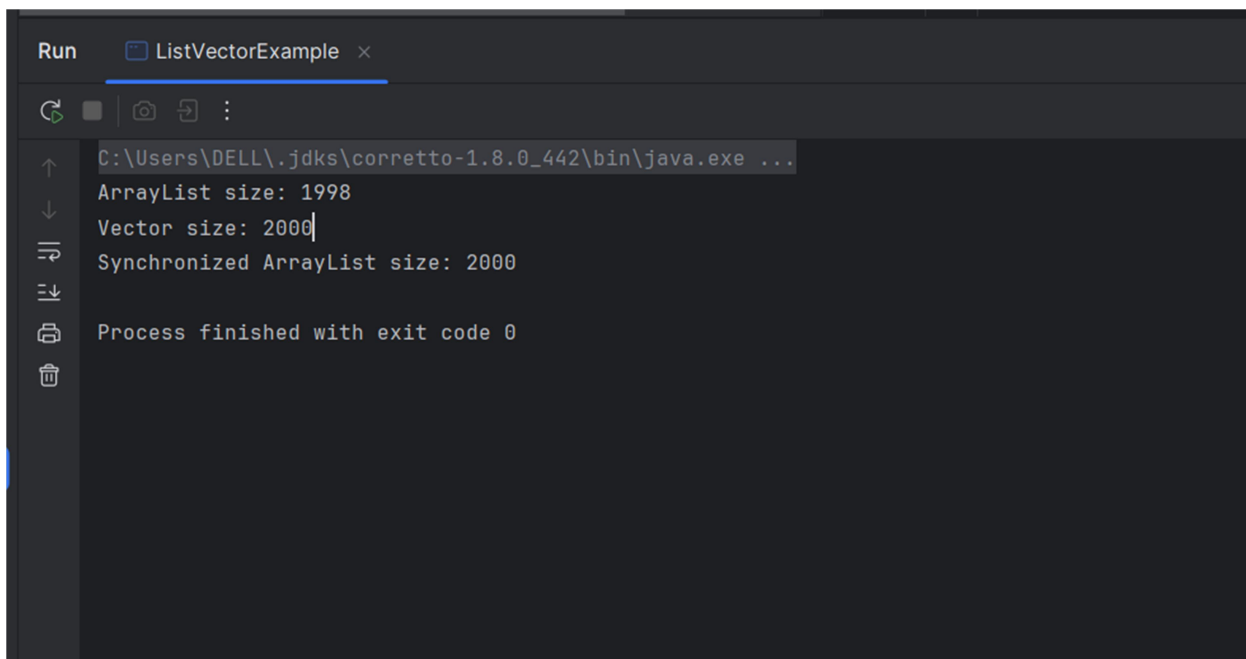
        Runnable task = () -> {
            for (int i = 0; i < 1000; i++) {
                arrayList.add("A" + i);
                vector.add("V" + i);
                syncArrayList.add("S" + i);
            }
        };

        Thread t1 = new Thread(task);
        Thread t2 = new Thread(task);
```

```
t1.start();
t2.start();
t1.join();
t2.join();

System.out.println("ArrayList size: " + arrayList.size());
System.out.println("Vector size: " + vector.size());
System.out.println("Synchronized ArrayList size: " + syncArrayList.size());
}
}
```

Output:



```
Run ListVectorExample x
C:\Users\DELL\.jdk\corretto-1.8.0_442\bin\java.exe ...
ArrayList size: 1998
Vector size: 2000
Synchronized ArrayList size: 2000

Process finished with exit code 0
```