[*https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/console/collection/CollectionsSynchronizedList.java*](https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/console/collection/CollectionsSynchronizedList.java)

*public class* CollectionsSynchronizedList {

 *public static void* main(String[] args) {
 *List*<String> list = Collections.*synchronizedList*(*new* ArrayList<>());

 *// Adding elements* list.add("One");
 list.add("Two");
 list.add("Three");

 *// Synchronizing iteration over the synchronized list
 synchronized* (list) {
 *for* (String item : list) {
 System.*out*.println(item);
 }
 }
 }
}

Choosing Between Collections.synchronizedList and CopyOnWriteArrayList

Collections.synchronizedList:

Use when you need a thread-safe list and the list is frequently modified.

All the methods are synchronized, which may cause contention among threads.

CopyOnWriteArrayList:

Use when the list is mostly read and infrequently modified.

Each mutative operation (like add, remove, set) results in a new copy of the entire array, which can be costly for large lists.

Provides an iterator that does not throw ConcurrentModificationException and reflects the state of the list at the point when the iterator was created.