

COPYONWRITEARRAYSET

(1) Characteristic of copyonwriteArraySet?

- > Thread safe.
- > copy on write mechanism.
- > No duplicate elements
- > Iterators do not reflect modifications during iteration.

Ex –

```
public static void main(String[] args) {
    // Create a CopyOnWriteArraySet
    CopyOnWriteArraySet<Integer> copySet = new CopyOnWriteArraySet<>();

    // Adding elements to the CopyOnWriteArraySet
    copySet.add(10);
    copySet.add(20);
    copySet.add(30);
    copySet.add(40);
    System.out.println("Initial Set: " + copySet);

    // Attempting to add duplicate elements
    copySet.add(20);
    System.out.println("After trying to add duplicate (20): " + copySet);

    // Removing an element
    copySet.remove(30);
    System.out.println("After removing 30: " + copySet);

    // Checking if an element exists
    boolean contains40 = copySet.contains(40);
    System.out.println("Contains 40: " + contains40);

    // Iterating over the CopyOnWriteArraySet
    System.out.println("Iterating using iterator:");
    Iterator<Integer> iterator = copySet.iterator();
    while (iterator.hasNext()) {
        Integer value = iterator.next();
        System.out.println(value);

        // Trying to modify the set during iteration
        // This will not throw ConcurrentModificationException,
        // but changes won't reflect during this iteration.
        if (value == 20) {
            copySet.add(50); // Modification during iteration
        }
    }

    System.out.println("After modification during iteration: " + copySet);

    // Using enhanced for loop for iteration
```

```
System.out.println("Iterating using enhanced for loop:");  
for (Integer value : copySet) {  
    System.out.println(value);  
}  
  
// Checking the size of the CopyOnWriteArraySet  
int size = copySet.size();  
System.out.println("Size of the set: " + size);  
}
```