Recording

Date : 7/07/2025

Problem Specification : Write a program to check whether two arrays are equal or not.

Assumption : Assume capacity is always greater than 0 and real number.

Limitation :
it does not handle null elements.
Program performs a linear comparison and its not good for large array sizes.
Only work on integer ArrayADT instances.
It does not handle null arrays.

Input :
capacity of the array.
Two arrays (Array ADT instances containing integer elements).

Processing :
check the two arrays lengths.
If array lengths are not equal then return false.
Initialize i=0 (I is index).
Compare elements at position i in both two arrays.
If pair of elements found unequal then return false.
After checking the whole array if no different elements found return true.

.

`

Output :
true: if arrays are equal.
false: is arrays are not equal.

Algorithm :
Step1: get 2 arrays.
Step2: Call the method checkEqual(array1,array2).
Step3: if array1.getsize() and array2.getsize() are not equal return false.
Step4: For each index i from 0 to array1.getSize() - 1:
    If array1.get(i) is not equal to array2.get(i):
    Return false.;
Step5: repeat step 4 for all elements .
Step7: if no difference found then return true.
.

Programme listing **:** Programme file attached

Test data and expected output **:** 1. Test data:

```
ArrayADT arr1 = new ArrayADT(8);
ArrayADT arr2 = new ArrayADT(8);
arr1.insert(0, 10);
arr1.insert(1, 20);
arr1.insert(2, 30);
arr1.insert(3, 40);
arr1.insert(4, 50);
arr1.insert(5, 60);
arr1.insert(6, 70);
arr1.insert(7, 80);

arr2.insert(0, 10);
arr2.insert(1, 20);
arr2.insert(2, 30);
arr2.insert(3, 40);
arr2.insert(4, 50);
arr2.insert(5, 60);
arr2.insert(6, 70);
arr2.insert(7, 80);
```

Expected output: true

”

2. Test data:

```
ArrayADT arr3 = new ArrayADT(8);
ArrayADT arr4 = new ArrayADT(8);

arr3.insert(0, 10);
arr3.insert(1, 20);
arr3.insert(2, 30);
arr3.insert(3, 40);
arr3.insert(4, 50);
arr3.insert(5, 60);
arr3.insert(6, 70);
arr3.insert(7, 80);
```

```
arr4.insert(0, 10);
arr4.insert(1, 20);
arr4.insert(2, 30);
arr4.insert(3, 90);
arr4.insert(4, 50);
arr4.insert(5, 45);
arr4.insert(6, 70);
arr4.insert(7, 80);
```

Expected output: false

Output obtained for test data :

1. Test data:

```
ArrayADT arr1 = new ArrayADT(8);
ArrayADT arr2 = new ArrayADT(8);
arr1.insert(0, 10);
arr1.insert(1, 20);
arr1.insert(2, 30);
arr1.insert(3, 40);
arr1.insert(4, 50);
arr1.insert(5, 60);
arr1.insert(6, 70);
arr1.insert(7, 80);

arr2.insert(0, 10);
arr2.insert(1, 20);
arr2.insert(2, 30);
arr2.insert(3, 40);
arr2.insert(4, 50);
arr2.insert(5, 60);
arr2.insert(6, 70);
arr2.insert(7, 80);
```

Obtained output: true

2. Test data:

```
ArrayADT arr3 = new ArrayADT(8);
ArrayADT arr4 = new ArrayADT(8);
```

```
arr3.insert(0, 10);
arr3.insert(1, 20);
arr3.insert(2, 30);
arr3.insert(3, 40);
arr3.insert(4, 50);
arr3.insert(5, 60);
arr3.insert(6, 70);
 arr3.insert(7, 80);

arr4.insert(0, 10);
arr4.insert(1, 20);
arr4.insert(2, 30);
arr4.insert(3, 90);
arr4.insert(4, 50);
arr4.insert(5, 45);
arr4.insert(6, 70);
arr4.insert(7, 80);
```

Obtained output: false

Analysis **:** The numbers of operation required in performing the algorithm.

	+,-	/,*	%	</>/<=/>=
For calculation	Size of array	-	-	Size of array

Conclusion : This checkEqual method accept two Array ADT instances and check whether they are equal or not by comparing their sizes and elements. .

Discussion **:**This program performs a linear comparison between two arrays.it performs O(n) comparisons where n is size of array.
If array size is large then time to check whether arrays are equal will increase linearly.