Recording

Date : 07/07/2025

Problem Specification : Check if two arrays are equal (having the same elements in the same order).

Assumption : Assume both arrays contain integer values. Arrays can be null or empty.

Limitation : This program only works for integer arrays. Does not handle arrays of different data types.

Input : Two integer arrays

Processing : Compare the two arrays by first checking null conditions, then comparing lengths, and finally comparing each element at corresponding positions.

Output : Boolean value (true if arrays are equal, false otherwise) or throws IllegalArgumentException for invalid inputs.

Algorithm :

Step 1: Check if both arrays are null - if yes, throw IllegalArgumentException

Step 2: Check if either array is null - if yes, throw IllegalArgumentException

Step 3: Check if array lengths are different - if yes, throw IllegalArgumentException

Step 4: Initialize counter i = 0

Step 5: Compare array1[i] with array2[i]

Step 6: If elements are not equal, return false

Step 7: Increment counter: i = i + 1

Step 8: If (i < array1.length) go to Step5

Step 9: Return true (all elements are equal)

Step 10: End the Program

Program listing : Program file attached

Test data and expected output :

1. **Test data**: array1 = {1, 2, 3, 4, 5}, array2 = {1, 2, 3, 4, 5}
   Expected output: true

2. **Test data**: array1 = {1, 2, 3, 4, 5}, array2 = {1, 2, 3, 4, 6}
   Expected output: false

3. **Test data**: array1 = null, array2 = {1, 2, 3}
   Expected output: IllegalArgumentException

Output obtained for test data :

1. **Test data**: array1 = {1, 2, 3, 4, 5}, array2 = {1, 2, 3, 4, 5}
   Obtained output: Are array1 and array2 equal? True

2. **Test data**: array1 = {1, 2, 3, 4, 5}, array2 = {1, 2, 3, 4, 6}
   Obtained output: Are array1 and array2 equal? false

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

	+,-	/,*	%	</>/<=/>=	==	!=
For calculation	-	-	-	-	-	N
Null checks	-	-	-	-	3	-
Length comparison	-	-	-	-	1	-
For loop	*N*	-	-	*N*	-	-

Conclusion : This program checks if two integer arrays are equal by comparing their elements in the same order. The program uses proper null checking and length validation before element comparison. A for loop is used to iterate through array elements.

Discussion : The time complexity of this algorithm is O(n) where n is the length of the arrays, as in the worst case we need to compare all elements. The space complexity is O(1) as no additional space is used. The program handles edge cases like null arrays and different length arrays by throwing appropriate exceptions. To make it more robust, we could consider using Arrays.equals() method from Java's built-in library, or extend it to handle arrays of different data types using generics.