**Breakdown:**

1. **Main Method:**

java

Copy code

public static void main(String[] args) {

String name1 = "Dawod";

String name2 = "woadD";

System.out.println(isAnagram(name1, name2));

}

* + Defines two strings, name1 and name2.
  + Calls the isAnagram method with these two strings and prints the result.

1. **Anagram Check Method:**

java

Copy code

public static boolean isAnagram(String str1, String str2) {

boolean res = false;

if (sort(str1).equals(sort(str2))) {

res = true;

}

return res;

}

* + Takes two strings, str1 and str2, as input.
  + Uses the sort method to sort the characters in each string.
  + Compares the sorted versions of the two strings.
  + If the sorted strings are equal, res is set to true, indicating that the strings are anagrams. Otherwise, res remains false.

1. **Sort Method:**

java

Copy code

public static String sort(String str) {

char[] strArr = str.toCharArray(); // Convert string to character array

Arrays.sort(strArr); // Sort the character array

str = String.valueOf(strArr); // Convert the sorted character array back to a string

return str;

}

* + Converts the input string str to a character array.
  + Sorts the character array using Arrays.sort.
  + Converts the sorted character array back to a string.
  + Returns the sorted string.

**Summary:**

* **Purpose:** The code checks if two strings are anagrams of each other. Two strings are anagrams if they contain the same characters but possibly in a different order.
* **Methodology:**
  1. The sort method sorts the characters of each string.
  2. The isAnagram method compares the sorted versions of the strings.
* **Output:** If the sorted versions of the two input strings are the same, it prints true, indicating the strings are anagrams. Otherwise, it prints false.

For example, given name1 = "Dawod" and name2 = "woadD", the output will be true because both strings contain the same characters.