How to check if Java Array Contains specific Object/Element?

Check if Java Array Contains Specified Value/Element

Java Array is a collection of ordered items of similar type. We can check if this array contains a specific value of interest. An element can occur any number of times. We will consider that the element is present, if it occurs at least once in the array.

In this tutorial, we will learn how to check if a Java Array contains a given value or not.

To check if an element is present in the array or not, we have to traverse through the array elements one by one, and check if the element of array equals the search element. The logic to check the equality may change with datatype. For example, to check if two integers are equal, we can use equal to comparison operator, but to check if two strings are equal, we may have to use String.equals() function. Similarly, for user defined objects, the logic to prove the equality may vary based on the requirement.

We will go through examples with integer arrays, string arrays and arrays of user defined datatype.

Example 1 – Integer Array – Check if Array Contains given Integer

In this example, we will initialize an integer array, and check if the array contains a given integer number in the array.

We will use advanced for loop to traverse through the items, and equal to comparison operator to check if array element equals search item. When we get a hit, we will decide that the array contains search element, and we will break the loop, because, it is already proven that the search element is present in the array.

Program

```
public class Example {
    public static void main(String[] args) {
        int arr[] = {87, 41, 52, 63, 98};
        int search = 52;
        for(int item: arr) {
            if(item == search) {
                System.out.println("The array contains "+search);
                break;
            }
        }
    }
}
```

Output

```
The array contains banana
```

Example 2 – String Array – Check if Array Contains given String

In this example, we will initialize a string array, and check if the array contains a given string value in the array.

Program

```
public class Example {
    public static void main(String[] args) {
        String arr[] = new String[] {"apple", "banana", "cherry", "mango"};
        String search = "banana";
        for(String item: arr) {
            if(item.equals(search)) {
                System.out.println("The array contains "+search);
                break;
            }
        }
    }
}
```

Output

```
The array contains banana
```

Example 3 – Object Array – Check if Array Contains given Object

In this example, we will initialize an array of user defined objects, and check if the array contains a given object in the array.

Our user defined objects would be of type Color. And we have written a method, Color.equals(), to check if two color objects are equal. We are overriding the inbuilt equals() method.

By default, the hash value of objects is compared to check if two objects are same. You can use equal to comparison operator just like we have compared integers. But, in this example, we are establishing the condition that the two objects are equal if their properties are equal.

Program

```
public class Example {
    public static void main(String[] args) {
        Color colors[] = {new Color("blue", 0), new Color("green", 1), new Color("red", 2)};
        Color search = new Color("red", 2);
        for(Color color: colors) {
            if(color.equals(search)) {
                System.out.println("The array contains search element.");
                break;
            }
        }
    }
}

class Color {
    public String name;
    public int value;
    public Color(String name, int value) {
        this.name = name;
        this.value = value;
    }
    public boolean equals(Color color) {
        if(this.name.equals(color.name) &&
                this.value == color.value) {
            return true;
        }
        return false;
    }
}
```

Output

```
The array contains search element.
```

Conclusion

In this Java Tutorial, we learned how to check if Java Array contains specific Object/Element.

Related Tutorials

- How to fix ArrayIndexOutOfBoundsException in Java?
- How to fix ArrayStoreException in Java?
- JavaScript – Arrays
- How to Initialize Array in Java?
- Java – Iterate over Array Elements
- Java Concatenate Arrays
- Java Program – Find Smallest Number of an Array
- Java Program to Find Largest Number of an Array
- Java Program to Print Elements of Array
- Java Array – For Loop
- Java Array – While Loop
- Java Array of Integers
- How to find index of Element in Java Array?
- Java – Check if Array is Empty

- Java – Append to Array
- Array of Arrays in Java
- Array of Objects in Java
- Java Arrays asList() – Examples
- Java Arrays copyOf() – Examples
- Java Arrays sort() – Examples
- Java – How to Sort an Array only in Specific Index Range?
- Java System.arraycopy() – Syntax & Examples