L19

Tuesday, January 25, 2022 3:12 PM

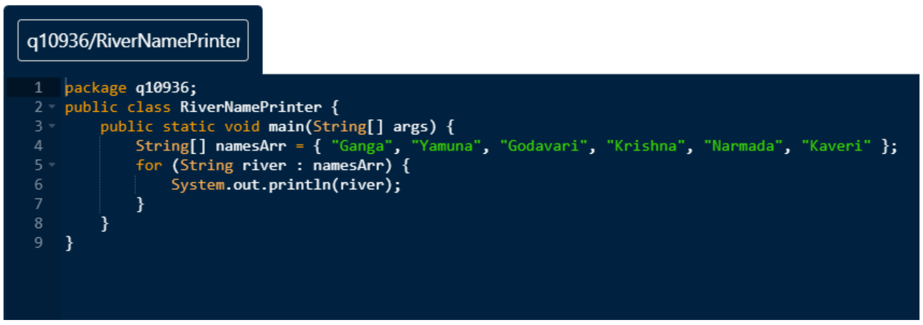
Array Access and Iterations
Saurav Hathi

https://www.youtube.com/channel/UCp6MFWao5vWRnyRCxBsKnfw

Print the elements in an array

Q1. Retype the code below. The class RiverNamePrinter prints an array containing the names of Indian rivers.

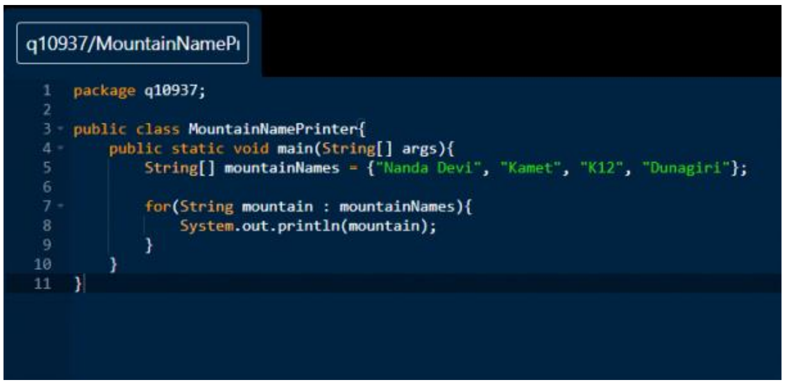
The code in the main method uses a for-each loop to iterate over the array namesArr and prints each name.


q10936/RiverNamePrinter

```java
package q10936;
public class RiverNamePrinter {
    public static void main(String[] args) {
        String[] namesArr = { "Ganga", "Yamuna", "Godavari", "Krishna", "Narmada", "Kaveri" };
        for (String river : namesArr) {
            System.out.println(river);
        }
    }
}
```

Q2. Create a class MountainNamePrinter with a main method. Create an array with the following names: **Nanda Devi**, **Kamet**, **K12**, **Dunagiri**.

Iterate over the array and print each name on a separate line.

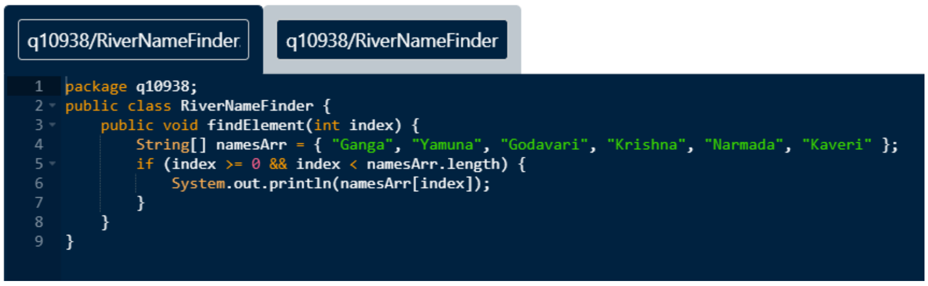

q10937/MountainNamePr

```java
package q10937;

public class MountainNamePrinter{
    public static void main(String[] args){
        String[] mountainNames = {"Nanda Devi", "Kamet", "K12", "Dunagiri"};

        for(String mountain : mountainNames){
            System.out.println(mountain);
        }
    }
}
```

Q3. Retype the code below. The class RiverNameFinder has a method findElement(int index). It accepts an integer argument index. The method prints the array element at that index.

The findElement method creates an array namesArr containing river names. It accesses the element of namesArr at index and prints the element.

Observe that before accessing the element, the method checks if the index is valid or not by checking if index is greater than zero and less than the size of namesArr. Otherwise, we get an ArrayIndexOutofBoundsException.



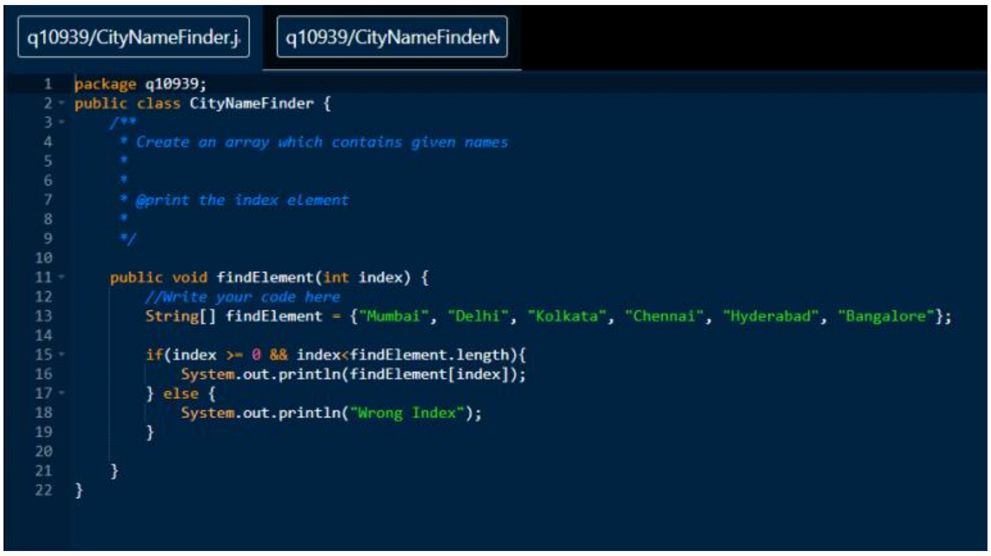
```java
package q10938;
public class RiverNameFinder {
    public void findElement(int index) {
        String[] namesArr = { "Ganga", "Yamuna", "Godavari", "Krishna", "Narmada", "Kaveri" };
        if (index >= 0 && index < namesArr.length) {
            System.out.println(namesArr[index]);
        }
    }
}
```

Q4. Create a class CityNameFinder with a **public** method findElement that takes one parameter index of type int.

The findElement method should create an array namesArr containing these city names: **Mumbai, Delhi, Kolkata, Chennai, Hyderabad, Bangalore**.
Write code in the method to print the element that is present at the **index** passed to the method.

The program should print **Wrong Index** if the value of index is not in the range of valid indices of the array.

For example:
Cmd Args : 99
Wrong Index



```java
package q10939;
public class CityNameFinder {
    /**
     * Create an array which contains given names
     *
     *
     * @print the index element
     *
     */

    public void findElement(int index) {
        //Write your code here
        String[] findElement = {"Mumbai", "Delhi", "Kolkata", "Chennai", "Hyderabad", "Bangalore"};

        if(index >= 0 && index<findElement.length){
            System.out.println(findElement[index]);
        } else {
            System.out.println("Wrong Index");
        }

    }
}
```

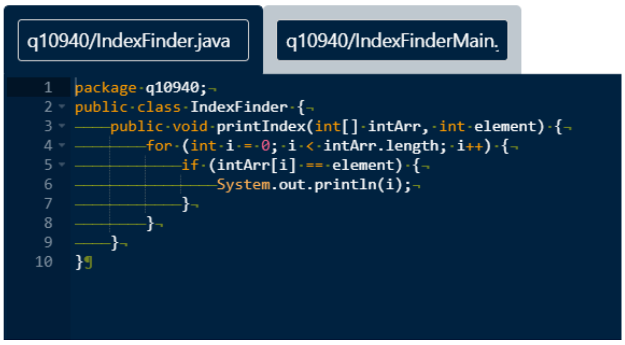
Q5. Retype and submit the code below.

Write a class IndexFinder with **public** method printIndex that takes two parameters one is intArr of type int[] and other is an element of type int and print all indices of the elements in intArr which are equal to the given element.

Assumptions:
1. arr is never null

Here is an example:

Cmd Args : 2 2 2 42 2
Indices of the elments matched with the given element 2 is :
0
1
2

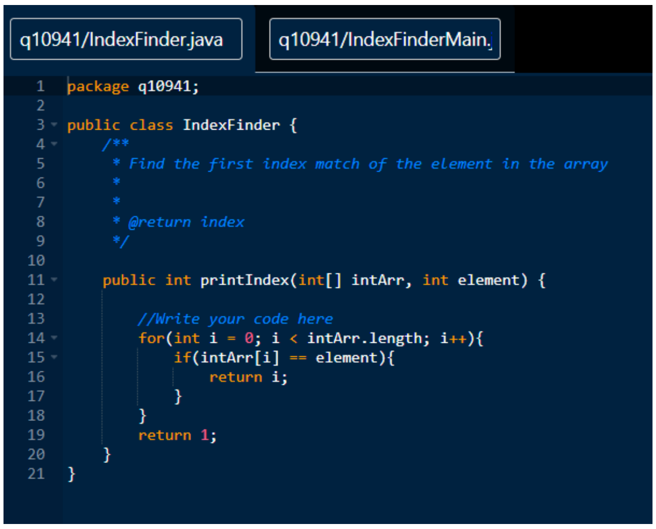


```java
package q10940;
public class IndexFinder {
	public void printIndex(int[] intArr, int element) {
		for (int i = 0; i < intArr.length; i++) {
			if (intArr[i] == element) {
				System.out.println(i);
			}
		}
	}
}
```

Q6. Create a class with name IndexFinder with public method printIndex that takes two parameters one is intArr of type int[] and second one is element of type int and returns only the first index match of the element in the intArr.

Assumptions:
1. arr is never null
2. arr may contain duplicate elements but returns the index of the first match of the element

Here is an example:
Cmd Args : 69 25 89 54 89
First match of the element 89 index is: 2

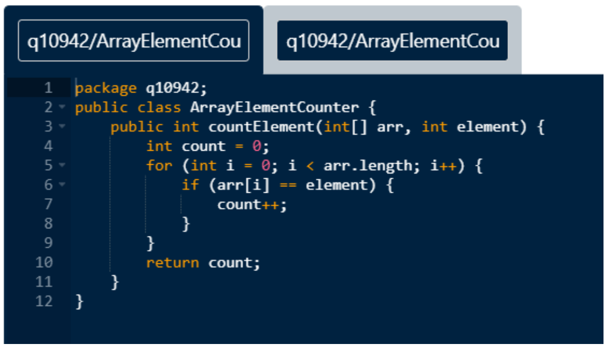

q10941/IndexFinder.java
q10941/IndexFinderMain.

```java
package q10941;

public class IndexFinder {
    /**
     * Find the first index match of the element in the array
     *
     *
     * @return index
     */

    public int printIndex(int[] intArr, int element) {

        //Write your code here
        for(int i = 0; i < intArr.length; i++){
            if(intArr[i] == element){
                return i;
            }
        }
        return 1;
    }
}
```

Q7. Retype the code below and submit.

Write a class ArrayElementCounter with a **public** method countElement that takes two parameters one is arr of type int[] and the other is element of type int and returns the count of the element that occures in the arr. The return type of countElement should be int.

Here is an example:
Cmd Args : 3 2 3 3 3 3
The element 3 presents 4 times in the arry

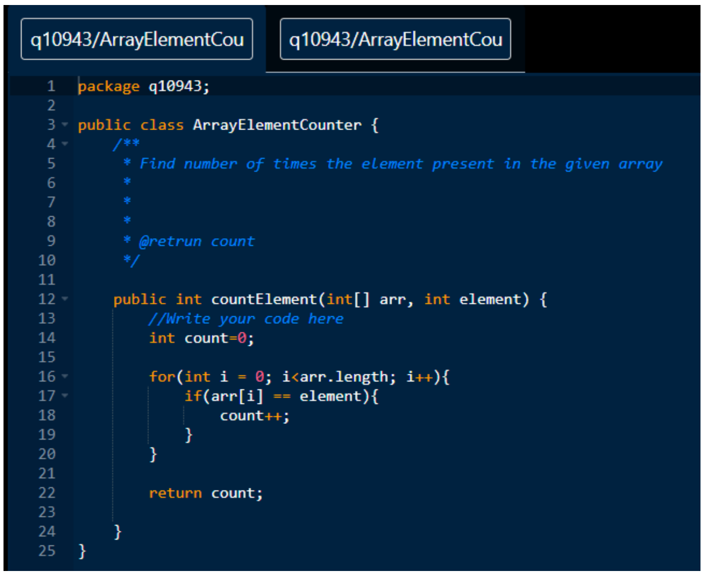

q10942/ArrayElementCou
q10942/ArrayElementCou

```java
package q10942;
public class ArrayElementCounter {
    public int countElement(int[] arr, int element) {
        int count = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == element) {
                count++;
            }
        }
        return count;
    }
}
```

Q8. Create a class ArrayElementCounter with a **public** method countElement that takes two parameters one is arr of type int[] and second one is element of type int and returns the count of element occures in the arr. The return type of countElement should be int.

Assumptions:
1. arr is never null

Here is an example:
Cmd Args : 1 1 2 3 1
2



```java
package q10943;

public class ArrayElementCounter {
    /**
     * Find number of times the element present in the given array
     *
     *
     *
     * @retrun count
     */

    public int countElement(int[] arr, int element) {
        //Write your code here
        int count=0;

        for(int i = 0; i<arr.length; i++){
            if(arr[i] == element){
                count++;
            }
        }

        return count;

    }
}
```

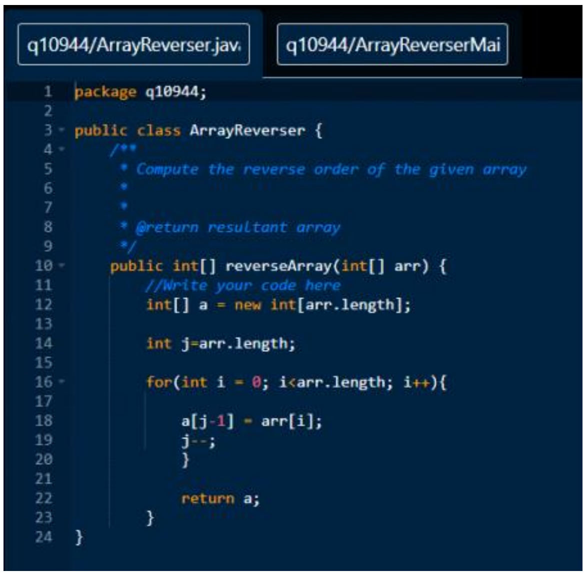
Q9. Create a class ArrayReverser with a **public** method reverseArray that takes one parameter arr of type int[] and returns all the elements in the arr in reverse order. The return type of reverseArray should be int.

Assumptions:

1. arr is never null

Here is an example:

```
Cmd Args : 1 2 3 4
4
3
2
1
```



```java
package q10944;

public class ArrayReverser {
    /**
     * Compute the reverse order of the given array
     *
     *
     * @return resultant array
     */
    public int[] reverseArray(int[] arr) {
        //Write your code here
        int[] a = new int[arr.length];

        int j=arr.length;

        for(int i = 0; i<arr.length; i++){

            a[j-1] = arr[i];
            j--;
            }

            return a;
        }
}
```