Java Programming Interview Questions for QA/SDET

Part -2

BHAVIN THUMAR

bhavin-thumar

Question 1: Write a program to check whether the given number is prime or not

The Input is: 7
Output should be: prime

```java
package coding;

/**
 * The CheckPrimeNumber class checks whether a given number is prime or not.
 *
 * @author Bhavin.Thumar
 */
public class CheckPrimeNumber {

    /**
     * Checks whether the given number is prime or not.
     *
     * @param n the number to be checked
     * @return true if the number is prime, false otherwise
     */
    public static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        } else {
            for (int i = 2; i < n; i++) {
                if (n % i == 0) {
                    return false;
                }
            }
        }
        return true;
    }

    /**
     * Main method to test the isPrime function with a sample input.
     *
     * @param args the command-line arguments
     */
    public static void main(String[] args) {
        if (isPrime(17)) {
            System.out.println("Prime");
        } else {
            System.out.println("Not Prime");
        }
    }
}
```

Question 2: Write a program to check whether the year is leap or not.

The Input is: 2020
Output should be: Leap

```java
package coding;

/**
 * The LeapYear class checks whether a given year is a leap year or not.
 *
 * @author Bhavin.Thumar
 */
public class LeapYear {

    /**
     * Checks whether the given year is a leap year or not.
     *
     * @param year the year to be checked
     * @return true if the year is a leap year, false otherwise
     */
    public static boolean isLeap(int year) {
        if (year % 4 == 0) {
            if (year % 100 == 0) {
                if (year % 400 == 0) {
                    return true;
                } else {
                    return false;
                }
            } else {
                return true;
            }
        } else {
            return false;
        }
    }

    /**
     * Main method to test the isLeap function with a sample input.
     *
     * @param args the command-line arguments
     */
    public static void main(String[] args) {
        if (isLeap(2020)) {
            System.out.println("Leap");
        } else {
            System.out.println("Not Leap");
        }
    }
}
```

Question 3: Write a program to duplicate elements in an integer array.

The Input is: 1, 3, 5, 7, 3, 1, 5
Output should be: 1 3 5

```java
package array;

/**
 * The FindDuplicates class finds and prints duplicate elements in an integer array.
 *
 * @author Bhavin.Thumar
 */
public class FindDuplicates {

    /**
     * Main method to find and print duplicate elements in the array.
     *
     * @param args the command-line arguments
     */
    public static void main(String[] args) {
        int arr[] = { 1, 3, 5, 7, 3, 1, 5 };

        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] == arr[j]) {
                    System.out.println(arr[i]);
                }
            }
        }
    }
}
```

Question 4: Write a program to merges two integer arrays into a single array

The Input is: 1 2 3 and 4 5 6
Output should be: 1 2 3 4 5 6

```java
package array;

/**
 * The MergeArray class merges two integer arrays into a single array.
 *
 * Bhavin.Thumar
 */
public class MergeArray {

	/**
	 * Main method to merge two arrays and print the merged array.
	 *
	 * @param args the command-line arguments
	 */
	public static void main(String[] args) {
		int firstArray[] = { 1, 2, 3 };
		int secondArray[] = { 4, 5, 6 };

		int firstArraySize = firstArray.length;
		int secondArraySize = secondArray.length;

		int mergeArraySize = firstArraySize + secondArraySize;

		int[] mergeArray = new int[mergeArraySize];

		for (int i = 0; i < firstArraySize; i++) {
			mergeArray[i] = firstArray[i];
		}

		for (int i = 0; i < secondArraySize; i++) {
			mergeArray[firstArraySize + i] = secondArray[i];
		}

		for (int i = 0; i < mergeArraySize; i++) {
			System.out.println(mergeArray[i]);
		}
	}
}
```

Question 5: Write a program to find the minimum and maximum elements in an integer.

The Input is: 3, 3, 9, 5, 9, 6, 4
Output should be: Minimum element: 3, Maximum element: 9

```
package array;

/**
 * The MinMaxElement class finds the minimum and maximum elements in an integer
 * array.
 *
 * @author Bhavin.Thumar
 */
public class MinMaxElement {

	/**
	 * Main method to find and print the minimum and maximum elements in the array.
	 *
	 * @param args the command-line arguments
	 */
	public static void main(String[] args) {
		int arr[] = { 3, 3, 9, 5, 9, 6, 4 };
		int min = arr[0];
		int max = arr[0];

		for (int i = 0; i < arr.length; i++) {
			if (arr[i] < min) {
				min = arr[i];
			}
			if (arr[i] > max) {
				max = arr[i];
			}
		}

		System.out.println("Minimum element: " + min);
		System.out.println("Maximum element: " + max);
	}
}
```

Question 6: Write a program to removes duplicate elements from an integer array.

The Input is: 1, 2, 4, 2, 5, 0, 1, 6, 5
Output should be: Unique elements: [1, 2, 4, 5, 0, 6]

```java
package array;

import java.util.LinkedHashSet;

/**
 * The RemoveDuplicates class removes duplicate elements from an integer array.
 *
 * @author Bhavin.Thumar
 */
public class RemoveDuplicates {

	/**
	 * Main method to remove duplicate elements from the array and print the unique
	 * elements.
	 *
	 * @param args the command-line arguments
	 */
	public static void main(String[] args) {
		int arr[] = { 1, 2, 4, 2, 5, 0, 1, 6, 5 };

		LinkedHashSet<Integer> uniqueArray = new LinkedHashSet<>();

		for (int i = 0; i < arr.length; i++) {
			uniqueArray.add(arr[i]);
		}

		System.out.println("Unique elements: " + uniqueArray);
	}
}
```

Question 7: Write a program to the elements of an integer array.

The Input is: 1, 2, 3, 4, 5
Output should be: Reversed Array: 5, 4, 3, 2, 1

```java
package array;

/**
 * The ReverseArray class reverses the elements of an integer array.
 *
 * @author Bhavin.Thumar
 */
public class ReverseArray {

    /**
     * Main method to reverse the elements of the array and print the reversed array.
     *
     * @param args the command-line arguments
     */
    public static void main(String[] args) {
        int[] a = { 1, 2, 3, 4, 5 };

        int right = a.length - 1;
        int left = 0;

        while (left < right) {
            int temp = a[left];
            a[left] = a[right];
            a[right] = temp;

            left++;
            right--;
        }

        System.out.println("Reversed Array:");
        for (int i = 0; i < a.length; i++) {
            System.out.println(a[i]);
        }
    }
}
```

Question 8: Write a program to sort the array and print the sorted array in ascending order.

The Input is: 4, 6, 1, 3, 8, 0, 2
Output should be: Sorted Array in Ascending Order:
0, 1, 2, 3, 4, 6, 8

```
package array;

/**
 * The SortingArray class sorts an array of integers in ascending order.
 *
 * @author Bhavin.Thumar
 */
public class SortingArray {

	/**
	 * Main method to sort the array and print the sorted array in ascending order.
	 *
	 * @param args the command-line arguments
	 */
	public static void main(String[] args) {
		int arr[] = { 4, 6, 1, 3, 8, 0, 2 };
		int temp = 0;

		for (int i = 0; i < arr.length; i++) {
			for (int j = i + 1; j < arr.length; j++) {
				if (arr[i] > arr[j]) {
					temp = arr[i];
					arr[i] = arr[j];
					arr[j] = temp;
				}
			}
		}

		System.out.println("Sorted Array in Ascending Order:");
		for (int i = 0; i < arr.length; i++) {
			System.out.println(arr[i]);
		}
	}
}
```