Java Assignment 2

Vaibhav Sharma

AIML-B2

22070126125

2022-26

Q1 Write a Java program that declares two arrays named 'even' and 'odd'. Accept numbers from the user and move them to respective arrays depending on whether they are even or odd.

Code:

```
// UserInput.java
package Assign2;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class UserInput {
    // Method to get an array input from the user
    public static double[] inputArray(int size) throws IOException {
        // Create a BufferedReader to read user input
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        // Prompt the user to enter the array elements
        System.out.println("Enter the array elements separated by spaces: ");

        // Read the array input as a string
        String array = br.readLine();

        // Initialize an array to store the input elements
```

```java
        double[] arrayInput = new double[size];

        // Split the input string and convert each element to double
        String[] input = array.trim().split("\\s+");

        // Populate the array with the converted elements
        for (int i = 0; i < size; i++) {
            arrayInput[i] = Double.parseDouble(input[i]);
        }

        // Return the array containing user-input elements
        return arrayInput;
    }
}

// OddEven.java
//Vaibhav Sharma
//AIML-B2
//2022-26
//22070126125
package Assign2;

import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;

public class OddEven {
    public static void main(String[] args) throws IOException {
        // Create a Scanner object to read user input
        Scanner sc = new Scanner(System.in);

        // Declare arrays for even and odd numbers
```

```java
double[] even;
int j = 0;
int k = 0;
double[] odd;

// Prompt the user to enter the number of elements
System.out.println("Enter the number of elements to enter:");

// Read the size from user input
int size = sc.nextInt();

// Initialize arrays for even and odd numbers based on the user-defined size
even = new double[size];
odd = new double[size];

// Get the array input from the user using the UserInput class
double[] array = UserInput.inputArray(size);

// Close the Scanner to avoid resource leaks
sc.close();

// Separate even and odd numbers into their respective arrays
for (int i = 0; i < size; i++) {
  if (array[i] % 2 == 0) {
    even[j] = array[i];
    j++;
  } else {
    odd[k] = array[i];
    k++;
  }
}
```

```
        // Print even elements
        System.out.println("Even elements:");
        System.out.println(Arrays.toString(even));

        // Print odd elements
        System.out.println("Odd elements:");
        System.out.print(Arrays.toString(odd));
    }

    // Method to print elements of an array
    public static void print(double[] array, int size) {
        for (int i = 0; i < size; i++) {
            System.out.println(array[i]);
        }
    }
}
```

Q2 Implement a Java function that finds two neighbouring numbers in an array with the smallest distance to each. The function should return the index of the 1st number.

Code:

```
// Neighbours.java
//Vaibhav Sharma
//AIML-B2
//2022-26
//22070126125
package Assign2;

import java.io.IOException;
import java.util.Scanner;

public class Neighbours {
    public static void main(String[] args) throws IOException {
        // Create a Scanner object to read user input
```

```java
        Scanner sc = new Scanner(System.in);

        // Prompt the user to enter the size of the array
        System.out.println("Enter the size of the array: ");

        // Read the size of the array from user input
        int size = sc.nextInt();

        // Get the array input from the user using the UserInput class
        double[] array = UserInput.inputArray(size);
        sc.close();

        // Find and print the index of the nearest neighbours in the array
        System.out.println("Index of Nearest Neighbours: " + findNearestNeighbours(array));
    }

    // Method to find the index of nearest neighbours in the array
    public static int findNearestNeighbours(double[] arr) {
        double minDistance = Double.MAX_VALUE;
        int index = -1;

        // Iterate through the array and calculate distances between adjacent elements
        for (int i = 0; i < arr.length - 1; i++) {
            double distance = Math.abs(arr[i] - arr[i + 1]);

            // Update the index if the current distance is smaller than the minimum distance
            if (distance < minDistance) {
                minDistance = distance;
                index = i;
            }
        }
```

```
        // Return the index of the nearest neighbours
        return index;
    }
}
```

Q3 Write a Java program to convert an array into ArrayList and vice versa.

Code:

```
// ArrayAndArrayList.java
//Vaibhav Sharma
//AIML-B2
//2022-26
//22070126125
package Assign2;

import java.io.IOException;
import java.util.*;

public class ArrayAndArrayList {
    public static void main(String[] args) throws IOException {
        // Create a Scanner object to read user input
        Scanner sc = new Scanner(System.in);

        // Prompt the user to enter the size of the array
        System.out.println("Enter the size of array: ");

        // Read the size from user input
        int size = sc.nextInt();

        // Get the primitive double array from user input using UserInput class
        double[] array = UserInput.inputArray(size);

        // Close the Scanner to avoid resource leaks
```

```java
        sc.close();

        // Print the primitive double array using Arrays.toString
        System.out.println(Arrays.toString(array));

        // Convert the primitive double array to an ArrayList<Double>
        List<Double> arrlist = Arrays.asList(Arrays.stream(array).boxed().toArray(Double[]::new));

        // Print the elements of the ArrayList using a for-each loop
        for (Double element : arrlist) {
            System.out.println(element);
        }
    }
}
```

Check my repo for all the assignments organized:
https://github.com/vaibhav7766/PIJ/tree/main/Assign2