22.Implementation of Static Hashing Technique using java

```
import java.util.Scanner;
class HashTable {
    int[] arr;
    int capacity;

    /** constructor **/
    public HashTable(int capacity) {
        this.capacity = nextPrime(capacity);
        arr = new int[this.capacity];
    }

    /** function to insert **/
    public void insert(int ele) {
        arr[ele % capacity] = ele;
    }

    /** function to clear **/
    public void clear() {
        arr = new int[capacity];
    }

    /** function contains **/
    public boolean contains(int ele) {
        return arr[ele % capacity] == ele;
    }

    /** function to delete **/
    public void delete(int ele) {
        if (arr[ele % capacity] == ele)
            arr[ele % capacity] = 0;
        else
            System.out.println("\nError : Element not found\n");
    }

    /** Function to generate next prime number >= n **/
    private static int nextPrime(int n) {
        if (n % 2 == 0)
            n++;
        for (; !isPrime(n); n += 2) ;
        return n;
    }
```

```
    /** Function to check if given number is prime **/
    private static boolean isPrime(int n) {
        if (n == 2 || n == 3)
            return true;
        if (n == 1 || n % 2 == 0)
            return false;
        for (int i = 3; i * i <= n; i += 2)
            if (n % i == 0)
                return false;
        return true;
    }

    /** function to print hash table **/
    public void printTable() {
        System.out.print("\nHash Table = ");
        for (int i = 0; i < capacity; i++)
            System.out.print(arr[i] + " ");
        System.out.println();
    }
}

/** Class HashTableTest **/
 class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Hash Table Test\n\n");
        System.out.println("Enter size");
        /** Make object of HashTable **/
        HashTable ht = new HashTable(scan.nextInt());
        char ch;
        /** Perform HashTable operations **/
        do {
            System.out.println("\nHash Table Operations\n");
            System.out.println("1. insert ");
            System.out.println("2. remove");
            System.out.println("3. contains");
            System.out.println("4. clear");
            int choice = scan.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("Enter integer element to insert");
                    ht.insert(scan.nextInt());
                    break;
                case 2:
```

```
                System.out.println("Enter integer element to delete");
                ht.delete(scan.nextInt());
                break;
            case 3:
                System.out.println("Enter integer element to check if present");
                System.out.println("Contains : " + ht.contains(scan.nextInt()));
                break;
            case 4:
                ht.clear();
                System.out.println("Hash Table Cleared\n");
                break;
            default:
                System.out.println("Wrong Entry \n ");
                break;
          }
          /** Display hash table **/
          ht.printTable();
          System.out.println("\nDo you want to continue (Type y or n) \n");
          ch = scan.next().charAt(0);
        } while (ch == 'Y' || ch == 'y');
    }
}
```