IT5512- WEB TECHNOLOGY LAB-SESSION-9

DATE: 01/11/2021

NAME: A.S. PRUTHIEV

REG NO.2019506067

IMPLEMENT MULTITHREADED PROGRAMS

1)AIM:

To write a Java multithreaded Program to perform matrix addition

PROGRAM CODE:

```
package Java.Lab.lab9;


import java.util.Scanner;


public class MatrixAddition {
   private static final Scanner input = new Scanner(System.in);
   public static void readInput(int [][]mat,int row,int col){
      System.out.println("Enter the elements in the matrix :");
      for(int i = 0 ; i < row ; i ++){
         for(int j = 0 ; j < col ; j++){
            mat[i][j] = input.nextInt();
         }
      }
   }
   public static void add(int [][]res,int [][]mat1,int [][]mat2,int row,int col){
      for(int i = 0 ; i < col ; i++){
         res[row][i] = mat1[row][i] + mat2[row][i];
```

```
        }
        System.out.println("The " + row + " row's sum is calculated using " +
Thread.currentThread().getName());
    }
    public static void printArray(int [][]res,int row,int col){
        System.out.println("The matrix after addition :");
        for(int i = 0 ; i < row ; i++){
            for(int j = 0 ; j < col; j ++){
                System.out.print(res[i][j] + " ");
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        int row,col;
        System.out.print("Enter the number of rows :");
        row = input.nextInt();
        System.out.print("Enter the number of cols :");
        col = input.nextInt();
        int [][]mat1 = new int[row][col];
        int [][]mat2 = new int[row][col];
        int [][]res = new int[row][col];
        Thread [] threads = new Thread[row];
        readInput(mat1,row,col);
        readInput(mat2,row,col);
        for(int i = 0 ; i < row ; i++){
```

```
            int finalI = i;
            threads[i] = new Thread(() -> {
                add(res,mat1,mat2,finalI,col);
            });
            threads[i].setName("Thread " + i);
            threads[i].start();
        }
        for(int i = 0 ; i < row ; i++){
          try{
              threads[i].join();
          }
          catch(InterruptedException exception){
              exception.printStackTrace();
          }
        }
        printArray(res,row,col);
    }
}
```

OUTPUT :

```
Run:  MatrixAddition
"C:\Users\Hema InduKan\.jdks\openjdk-17.0.1\bin\java.exe" "-j
Enter the number of rows :2
Enter the number of cols :2
Enter the elements in the matrix :
1 2 3 4
Enter the elements in the matrix :
5 6 7 8
Thread 0 Perform the 0 th addition
The 0 row's sum is calculated using Thread 0
Thread 1 Perform the 1 th addition
The 1 row's sum is calculated using Thread 1
The matrix after addition :
6 8
10 12

Process finished with exit code 0
```

RESULT :

Thus the output of the program has been successfully executed

2)AIM:

To write a Java multithreaded Program to perform matrix subtraction

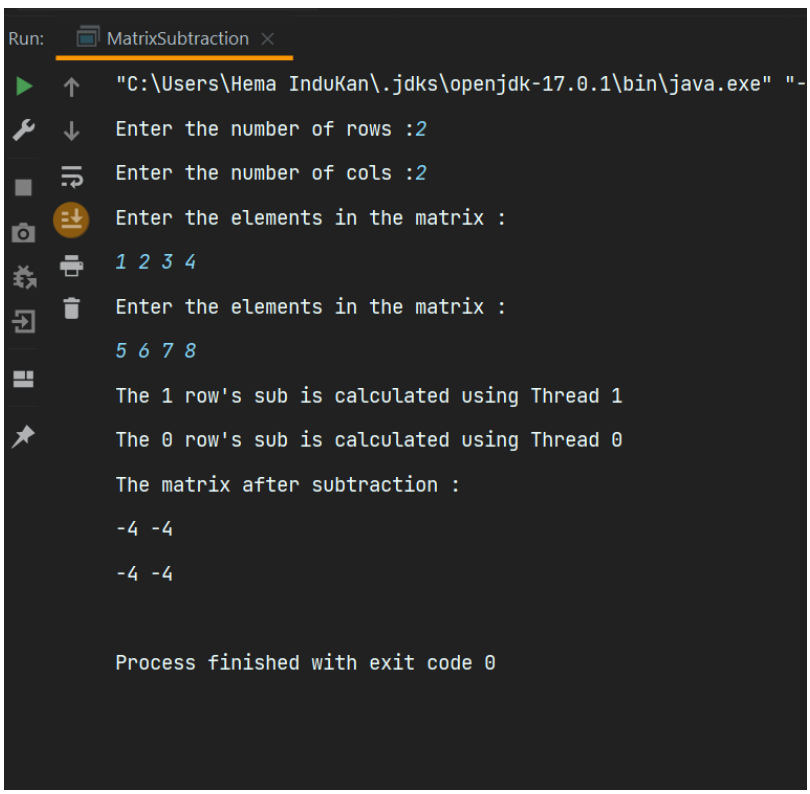
PROGRAM CODE:

```
package Java.Lab.lab9;


import java.util.Scanner;


public class MatrixSubtraction{
   private static final Scanner input = new Scanner(System.in);
   public static void readInput(int [][]mat,int row,int col){
      System.out.println("Enter the elements in the matrix :");
      for(int i = 0 ; i < row ; i ++){
         for(int j = 0 ; j < col ; j++){
            mat[i][j] = input.nextInt();
         }
      }
   }
   public static void add(int [][]res,int [][]mat1,int [][]mat2,int row,int col){
      for(int i = 0 ; i < col ; i++){
         res[row][i] = mat1[row][i] - mat2[row][i];
      }
      System.out.println("The " + row + " row's sub is calculated using " +
Thread.currentThread().getName());
   }
   public static void printArray(int [][]res,int row,int col){
      System.out.println("The matrix after subtraction :");
```

```
    for(int i = 0 ; i < row ; i++){
        for(int j = 0 ; j < col; j ++){
            System.out.print(res[i][j] + " ");
        }
        System.out.println();
    }
}
public static void main(String[] args) {
    int row,col;
    System.out.print("Enter the number of rows :");
    row = input.nextInt();
    System.out.print("Enter the number of cols :");
    col = input.nextInt();
    int [][]mat1 = new int[row][col];
    int [][]mat2 = new int[row][col];
    int [][]res = new int[row][col];
    Thread [] threads = new Thread[row];
    readInput(mat1,row,col);
    readInput(mat2,row,col);
    for(int i = 0 ; i < row ; i++){
        int finalI = i;
        threads[i] = new Thread(() -> {
            add(res,mat1,mat2,finalI,col);
        });
        threads[i].setName("Thread " + i);
        threads[i].start();
```

```
        }
        for(int i = 0 ; i < row ; i++){
            try{
                threads[i].join();
            }
            catch(InterruptedException exception){
                exception.printStackTrace();
            }
        }
        printArray(res,row,col);
    }
}
```

OUTPUT :



RESULT : Thus the output of the program has been successfully executed

3)AIM:

To write a Java multithreaded Program to sort the even elements in one thread,odd elements in second thread , find the even average using third thread and find the odd average using fourth thread

PROGRAM CODE:

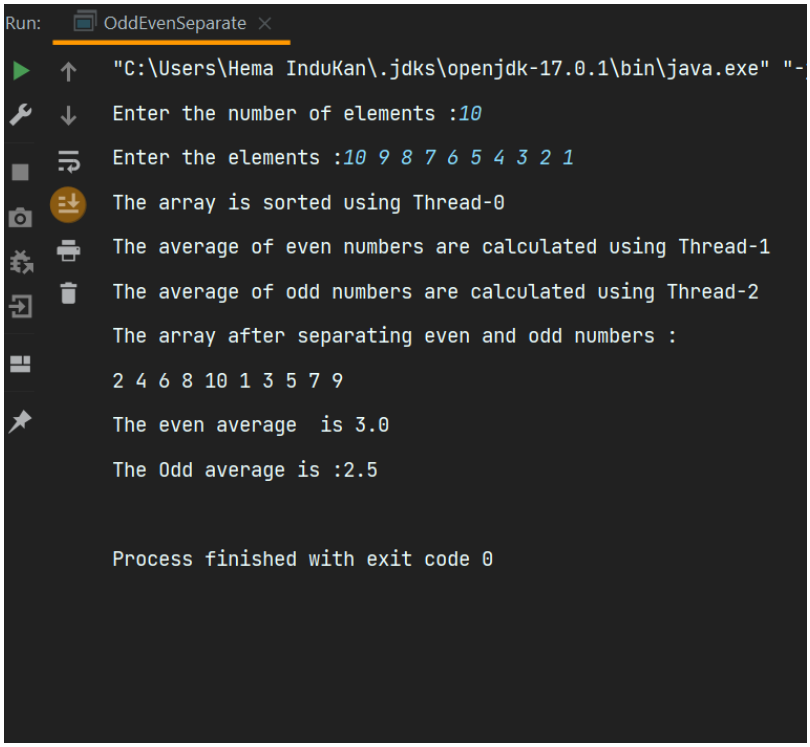
```
package Java.Lab.lab9;


import java.util.*;


class SortArrayEven implements Comparator<Integer>{
   @Override
   public int compare(Integer o1, Integer o2) {
      if (o1 % 2 == 0 && o2 % 2 == 0) {
         return o1 - o2;
      }
      else if(o1 % 2 == 0 && o2 % 2 == 1)return -1;
      else if(o1 % 2 == 1 && o2 % 2 == 0)return 1;
      else return o1 - o2;
   }
}


public class OddEvenSeparate {
   private static final Scanner input = new Scanner(System.in);
   public static void readInput(Integer []arr,int n){
      System.out.print("Enter the elements :");
      for(int i = 0 ; i < n ; i++){
```

```
            arr[i] = input.nextInt();
        }
    }
    public static void sortEvenOddNumbers(Integer []arr,int n){
        Arrays.sort(arr,new SortArrayEven());
        System.out.println("The array is sorted using " + Thread.currentThread().getName());
    }
    public static void calAvgEvenNumbers(Integer []arr,double []avg,int n){
        double evenAvg = 0.00;
        for(int i = 0 ; i < n ; i ++){
            if(arr[i] % 2 == 0)evenAvg += arr[i];
        }
        avg[0] =  evenAvg / n;
        System.out.println("The average of even numbers are calculated using " +
Thread.currentThread().getName());
    }
    public static void calAvgOddNumbers(Integer []arr,double []avg , int n){
        double oddAvg = 0.00;
        for(int i = 0 ; i < n ; i ++){
            if(arr[i] % 2 == 1)oddAvg += arr[i];
        }
        avg[1] =  oddAvg / n;
        System.out.println("The average of odd numbers are calculated using " +
Thread.currentThread().getName());
    }
    public static void main(String[] args) {
        int n;
```

```
System.out.print("Enter the number of elements :");
n = input.nextInt();
Integer []arr = new Integer[n];
readInput(arr,n);
double [] avg = new double[2];
Thread []thread = new Thread[3];
(thread[0] = new Thread(() -> {
   sortEvenOddNumbers(arr,n);
})).start();
(thread[1] = new Thread(() -> {
  calAvgEvenNumbers(arr,avg,n);
})).start();
(thread[2] = new Thread(() -> {
  calAvgOddNumbers(arr,avg,n);
})).start();
for(int i = 0 ; i < 3 ; i ++){
   try{
      thread[i].join();
   }
   catch(InterruptedException exception){
      exception.printStackTrace();
   }
}
System.out.println("The array after separating even and odd numbers :");
for(int i = 0 ; i < n ; i++){
   System.out.print(arr[i] + " ");
```

```
        }
        System.out.println("\nThe even average  is " + avg[0]);
        System.out.println("The Odd average is :" + avg[1]);


    }
}
```

OUTPUT :



RESULT : Thus the output of the program has been successfully executed