***Primera Clase***

package ejercicios;

import java.util.Random;

import java.util.Scanner;

public class sumaFilasTotal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Ingrese el número de filas");

int f = sc.nextInt();

System.out.println("Ingrese el número de columnas");

int c = sc.nextInt();

int [][] matriz = llenarMatriz(f,c);

int [] sumas = sumarFilasTotal(matriz);

int total = sumarTotal(sumas);

System.out.println("Matriz >> Sumas por filas");

for (int i = 0; i < f; i++) {

for (int j = 0; j < c; j++) {

System.out.print(matriz[i][j] + " ");

}

System.out.println(" >> " + sumas[i]);

}

/\*

System.out.println("Sumas por Filas");

for (int xd: sumas) {

System.out.println(">> " + xd);

}

\*/

System.out.println("Suma total:" + total);

}

public static int[] [] llenarMatriz(int f, int c) {

Random r = new Random();

int [][] matriz = new int[f][c];

for (int i = 0; i < f; i++) {

for (int j = 0; j < c; j++) {

matriz[i][j] = r.nextInt(9) + 1;

}

}

return matriz;

}

public static int[] sumarFilasTotal(int[][] matriz) {

int[] sumasFilas = new int[matriz.length];

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

for (int j = 0; j < matriz[0].length; j++) {

sumasFilas[i] += matriz[i][j];

}

}

return sumasFilas;

}

public static int sumarTotal(int[] vector) {

int total = 0;

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

total += vector[i];

}

return total;

}

}

***Segunda Clase***

package ejercicios;

import java.util.Scanner;

public class sumaFilasTotalHilos2 implements Runnable {

private int[] row;

private static int[] sumaTotal;

private int index = 0;

public sumaFilasTotalHilos2(int[] row, int index) {

this.row = row;

this.index = index;

}

public int[] getRow() {

return row;

}

public static int[] getSumaTotal() {

return sumaTotal;

}

@Override

public void run() {

int sum = 0;

for (int num : row) {

sum += num;

}

/\* ESTO ES PARA SINCRONIZAR EL ACCESO AL VECTOR Y EVITAR INCONSISTENCIA

synchronized (sumaFilasTotalHilos2.class) {

sumaTotal[index] = sum;

}

\*/

sumaTotal[index] = sum;

System.out.printf("%s trabajando. Total: %d\n", Thread.currentThread().getName(), sum);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Ingrese el número de filas");

int f = sc.nextInt();

System.out.println("Ingrese el número de columnas");

int c = sc.nextInt();

int[][] matriz = sumaFilasTotal.llenarMatriz(f, c);

sumaTotal = new int[f]; // para no estar inicializando el suma total por cada hilo

// simplemente se hace que este sumaTotal sea static y se lo inicializa con el tamaño de filas

// que también será la cantidad de hilos generados

Thread[] threads = new Thread[f];

for (int i = 0; i < f; i++) {

threads[i] = new Thread(new sumaFilasTotalHilos2(matriz[i], i)); // mando el índice para evitar

// crear un contador static e ir sumándolo

threads[i].start();

}

for (int i = 0; i < f; i++) {

try {

threads[i].join();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

int sT = 0;

for (int xd : getSumaTotal()) {

sT += xd;

}

System.out.println("Suma total: " + sT);

presentarMatriz(matriz);

}

public static void presentarMatriz(int[][] matriz) {

System.out.println(">>> PRESENTAR MATRIZ <<<");

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

for (int j = 0; j < matriz[0].length; j++) {

System.out.print(matriz[i][j] + " ");

}

System.out.println();

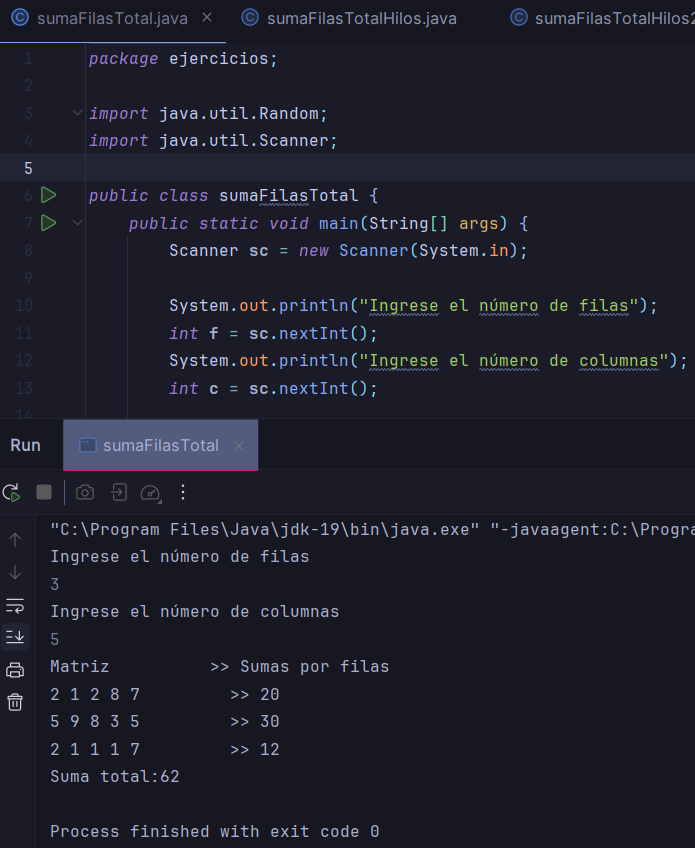
}

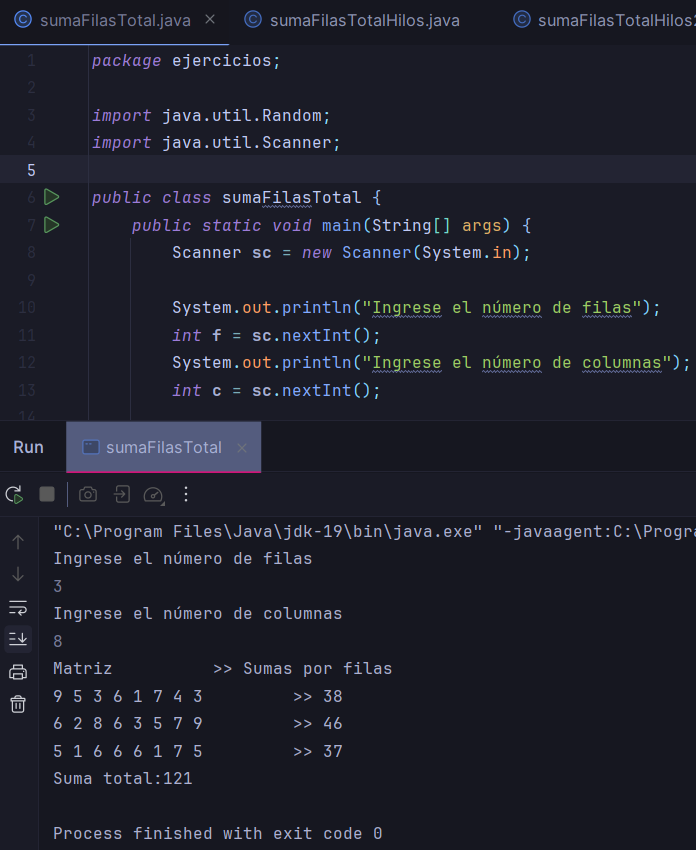
}

}

***Capturas:***

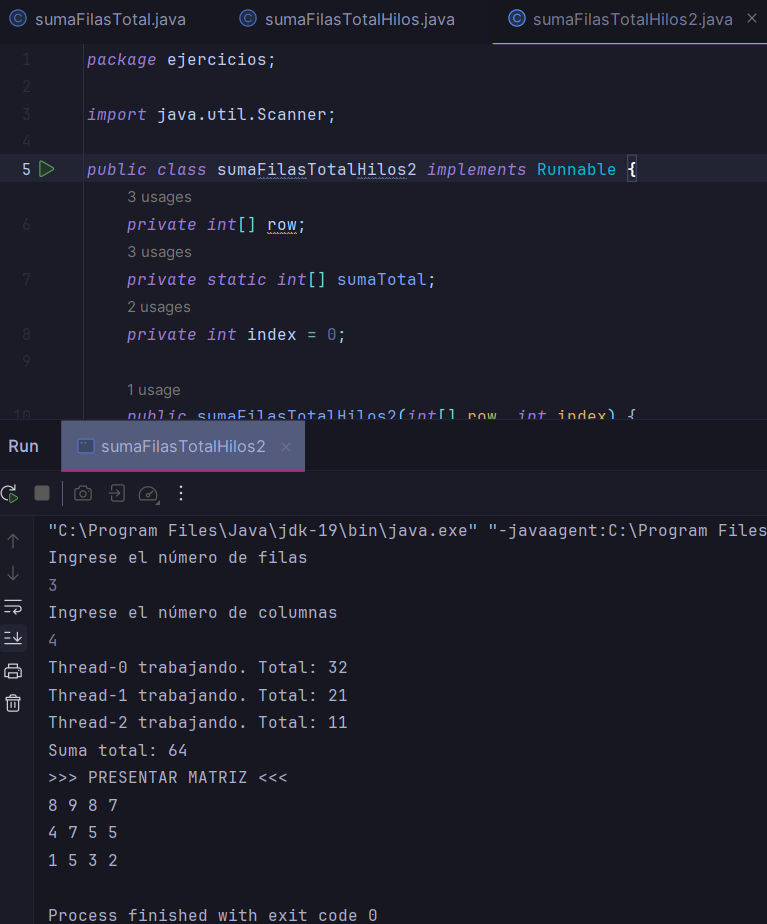
***Primera clase***

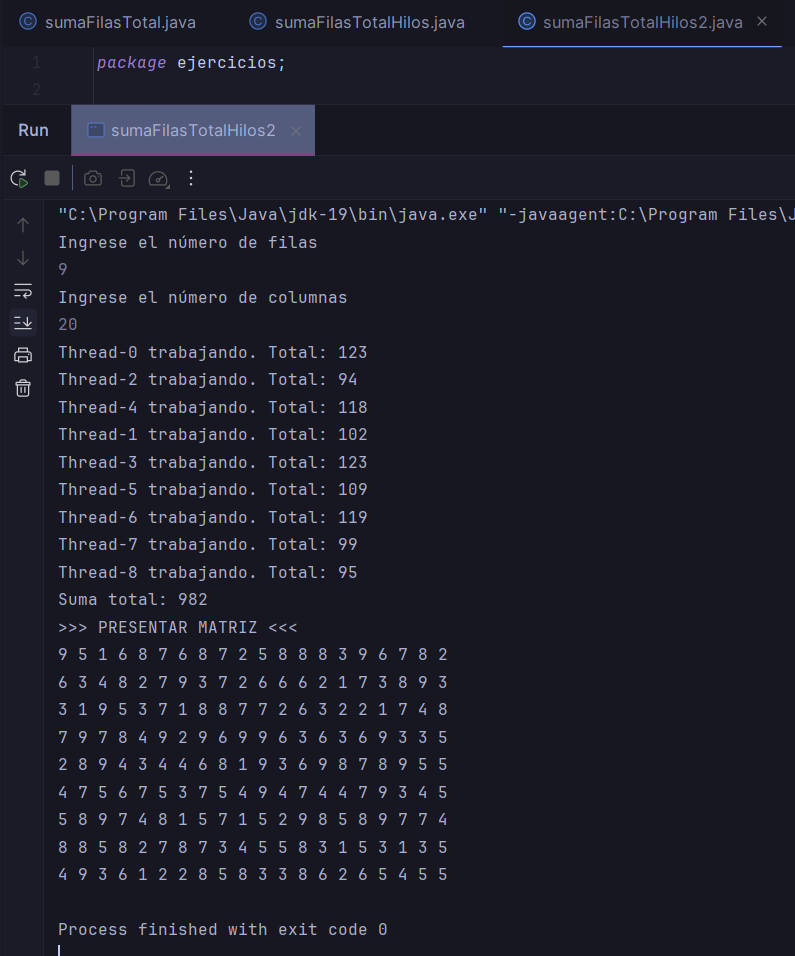
******

******

***Segunda clase***

***Matriz Secuencial***

******

******