

## Programación Avanzada Curso 2011

Cargue cada una de las clases siguientes y ejecútelas varias veces; observe los resultados (anótelos)

```
/**
 * CalculatePrimes -- calculate as many primes as we can in ten seconds
 * Del tutorial de IBM
 */

public class CalculatePrimes extends Thread {

    public static final int MAX_PRIMES = 1000000;
    public static final int SECONDS = 5000;

    public volatile boolean finished = false;

    public void run() {
        int[] primes = new int[MAX_PRIMES];
        int count = 0;

        for (int i=2; count<MAX_PRIMES; i++) {

            // Check to see if the timer has expired
            if (finished) {
                break;
            }

            boolean prime = true;
            for (int j=0; j<count; j++) {
                if (i % primes[j] == 0) {
                    prime = false;
                    break;
                }
            }

            if (prime) {
                primes[count++] = i;
                System.out.println("Found prime: " + i);
            }
        }
    }

    public static void main(String[] args) {
        CalculatePrimes calculator = new CalculatePrimes();
        calculator.start();
        try {
            Thread.sleep(SECONDS);
        }
        catch (InterruptedException e) {
            // fall through
        }
        calculator.finished = true;
    }
}
```

```

/*
 * Del tutorial de IBM
 */

public class TwoThreads {

    public static class Thread1 extends Thread {
        public void run() {
            System.out.println("A");
            System.out.println("B");
        }
    }

    public static class Thread2 extends Thread {
        public void run() {
            System.out.println("1");
            System.out.println("2");
        }
    }

    public static void main(String[] args) {
        new Thread1().start();
        new Thread2().start();
    }
}

```

```

import java.util.Calendar;
import java.util.Random;

public class UsoDualCore {

    private double [][] matriz;
    private double []resulta;
    private Trabajador tra, tra2, tra3, tra4;

    class Trabajador extends Thread{
        int ini, fin, quan, quien;
        long tinicial, tfinal;

        public Trabajador(int a,int b, int c, int qu){
            ini = a; fin = b; quan = c; quien = qu;
        }

        public void run(){
            int ix, jx; double suma, mini, maxi;
            // agregar tiempo
            tinicial = Calendar.getInstance().getTimeInMillis();
            System.out.println("Comienza a trabajar el trabajador
"+quien+" a los "+tinicial);
            for (ix=ini; ix<fin; ix++){
                suma = 0; mini =Double.MAX_VALUE; maxi =
Double.MIN_VALUE;
                for (jx=0; jx<quan; jx++){

```

```

        suma += matriz[ix][jx]*matriz[ix][jx];
        if (matriz[ix][jx]<mini)mini =
matriz[ix][jx];
        if (matriz[ix][jx]>maxi)maxi =
matriz[ix][jx];
    }
    resulta[ix]=Math.sqrt((maxi -mini)/suma);
}
tfinal = Calendar.getInstance().getTimeInMillis();
System.out.println("Termina de trabajar el trabajador
"+quien+" a los "+tfinal);
    System.out.println("Diferencia trabajador "+quien+" =
"+(tfinal-tinicial));
    //agregar tiempo
}
}

public UsoDualCore(){
    // crear datos en matriz
    llenarMatriz(1000,1000);
    // lanzar hilos
    // Versión de un hilo
    tra = new Trabajador(0,1000,1000,1);
    /* Versión de dos trabajadores
    tra = new Trabajador(0, 500, 1000, 1);
    tra2 = new Trabajador(500,1000, 1000, 2);
    */

    /* Versión de cuatro trabajadores
    tra = new Trabajador(0, 250, 1000, 1);
    tra2 = new Trabajador(250,1000, 1000, 2);
    tra3 = new Trabajador(500, 750, 1000, 3);
    tra4 = new Trabajador(750,1000, 1000, 4);
    */
    tra.start();
    /*
    tra2.start();
    tra3.start();
    tra4.start();
    */
    //incluir total de tiempo
}

private void llenarMatriz(int a, int b){
    int ix, jx;
    matriz = new double[a][b];
    resulta = new double[a];
    Random r = new Random();
    for (ix=0; ix<a; ix++)
        for (jx=0; jx<b; jx++)
            matriz[ix][jx]= r.nextDouble();
}

/**
 * @param args
 */
public static void main(String[] args) {

```

```
// TODO Auto-generated method stub

new UsoDualCore();
System.out.println("Termina hilo principal");
}
}
```