

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.Random;
public class DiezHilos {

    /**
     * Ejemplo para observar reparto de trabajo
     * Adaptado de "Introduction to Java Threads", Brian Goetz, ibm
     * completado en getGranmatriz
     */
    static Random alea=new Random();
    private static class HiloTrabajador extends Thread{
        int maxi=Integer.MIN_VALUE;
        int [] elarreglo;
        public HiloTrabajador(int[] mipedazo){
            elarreglo = mipedazo;
        }
        public void run(){
            for (int i=0; i<elarreglo.length; i++)
                maxi = Math.max(maxi, elarreglo[i]);
        }
        public int getMaxi(){return maxi;}
    }

    public static int[][] getGranMatriz(int mi, int mj){
        int[][] a=new int[mi][mj];
        for (int ix=0; ix<mi;ix++)
            for (int jx=0; jx<mj;jx++)
                a[ix][jx]=alea.nextInt();
        return a;
    }
}

```

```

public static void main(String[] args) {
    HiloTrabajador[] hilos=new HiloTrabajador[10];
    int[][]granmatriz= getGranMatriz(10,100000);
    int maxx =Integer.MIN_VALUE;

    for (int j=0; j<10; j++){
        hilos[j]=new HiloTrabajador(granmatriz[j]);
        hilos[j].start();
    }

    try{
        for (int k=0; k<10; k++){
            hilos[k].join();
            maxx=Math.max(maxx, hilos[k].getMaxi());
            System.out.println("Hilo "+k+" tiene máximo
"+hilos[k].getMaxi());
        }
        }catch (InterruptedException e){System.out.println("Problema
de insomnio");
    }

    System.out.println("El máximo es "+maxx);

}
}

```