

```

/**
 * Clase MVA para la implementación del algoritmo Mean Value Analysis (MVA).
 */
package MVA;

/**
 * @author Alberto Pérez Ancín
 */
public class MVA {

    /**
     * Método principal para ejecutar el algoritmo MVA.
     * @param args Los argumentos de la línea de comandos.
     */
    public static void main(String[] args) {
        // Número total de trabajos
        int N = 10;

        // Tiempo de reflexión
        double thinkTime = 8.0;

        // Tiempo de servicio para cada dispositivo
        double[] serviceTime = {0.03, 0.1};

        // Proporción de visitas para cada dispositivo
        int[] visitRatios = {8, 7};

        // Número de dispositivos
        int K = serviceTime.length;

        // Matriz de tiempos de respuesta para cada dispositivo y cada número de
        // trabajos
        double[][] responseTime = new double[K][N + 1];

        // Array de tiempos de respuesta del sistema para cada número de
        // trabajos
        double[] systemResponseTime = new double[N + 1];

        // Array de productividad para cada número de trabajos
        double[] throughput = new double[N + 1];

        // Matriz del número de trabajos en cada dispositivo para cada número de
        // trabajos
        double[][] numJobs = new double[K][N + 1];

        System.out.println("Trabajos R1 R2 R X0 N1 N2");

        // Para cada número de trabajos...
    }
}

```

```

for (int n = 1; n <= N; n++) {
    // Para cada dispositivo...
    for (int i = 0; i < K; i++) {
        // Calcula el tiempo de respuesta para este dispositivo y este
        // número de trabajos
        responseTime[i][n] = (numJobs[i][n - 1] + 1) * serviceTime[i];

        // Agrega al tiempo de respuesta del sistema la contribución de
        // este dispositivo
        systemResponseTime[n] += visitRatios[i] * responseTime[i][n];
    }

    // Calcula la tasa de rendimiento para este número de trabajos
    throughput[n] = n / (thinkTime + systemResponseTime[n]);

    // Para cada dispositivo...
    for (int i = 0; i < K; i++) {
        // Calcula el número de trabajos en este dispositivo para este
        // número de trabajos
        numJobs[i][n] = throughput[n] * visitRatios[i] *
            responseTime[i][n];
    }

    // Imprime los resultados para este número de trabajos
    System.out.printf("%d %.4f %.4f %.4f %.4f %.4f %.4f%n", n,
        responseTime[0][n], responseTime[1][n],
        systemResponseTime[n], throughput[n], numJobs[0][n],
        numJobs[1][n]);
    }
}
}

```