

Práctica 4 – Programación (1)

1. Introducción

El objetivo de esta práctica es realizar unos primeros ejercicios de programación, familiarizarse con el manejo del compilador de java y las funcionalidades básicas del lenguaje.

Esta práctica y las posteriores se pueden hacer fuera del laboratorio en cualquier sistema operativo donde se tenga instalado el JDK, como puede ser por ejemplo en los [escritorios virtualizados de la universidad¹](https://vdibroker.unavarra.es), en los PC SC de los armarios del laboratorio o en los PCs del otro lado del aula.

2. Parseo de argumentos

Guarde el siguiente programa en un fichero llamado P4.java :

```
public class P4 {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Uso: java P4 <n>");
            return;
        }

        int n = parse_n(args[0]);
        System.out.printf("He recibido el entero %d\n", n);
    }

    public static int parse_n(String arg) {
        return 0;
    }
}
```

Compílelo y ejecútelo con:

```
$ javac P4.java
$ java P4
```

Modifique el código de la función `parse_n` para que el número que devuelva sea el valor indicado por línea de comandos. Tenga en cuenta que deberá considerar los posibles errores si el usuario introduce algo que no es un número entero.

Verifique a su vez que el entero que le facilita el usuario es un número positivo (mayor o igual a 0), imprimiendo en su defecto un mensaje de error que le indique al usuario este requisito.

¹ <https://vdibroker.unavarra.es>

3. Primer bucle

Continuando con el programa anterior, realice las modificaciones necesarias para que el programa imprima por pantalla la serie {1, 0.1, 0.01, 0.001}, llegando al número de decimales indicado en el argumento del programa. Por ejemplo:

```
$ java P4 0
1

$ java P4 4
1
0.1
0.01
0.001
0.0001
```

Punto de control 1: Muestre al profesor el programa indicado.

Guarde el código del programa en algún sitio que le permita consultarlo en otro momento, pues le será útil más adelante.

4. ErlangB inversa

En este apartado partiremos de cualquiera de los ejemplos de programa de cálculo de la Erlang B y alteraremos su funcionamiento.

En concreto, los ejemplos vistos aplican la fórmula de Erlang B para calcular :

$B(A, k) = ?$

Lo que vamos a hacer es calcular el mínimo número de canales necesarios para cumplir con una probabilidad de bloqueo dada. Es decir, resolver la siguiente expresión:

$B(A, ?) = P_b$

La ejecución del programa deberá ser:

```
$ java P4 <a> <pb>
```

Los símbolos "<>" no son para ponerlos sino para indicar que se sustituya eso por un valor.

Para que puedan probar si el funcionamiento es el adecuado, la salida debe ser por ejemplo:

```
$ java P4 0.52 0.1
2
$ java P4 0.52 0.001
5
$ java P4 5 0.001
14
```

Punto de control 2: Muestre al profesor el programa indicado.

5. Modificaciones al cálculo de la ErlangB

Finalmente, a partir del número de canales y la probabilidad de bloqueo, calcule la máxima intensidad de tráfico que puede soportar el sistema antes de exceder la probabilidad de bloqueo indicada, con una resolución mínima de 6 decimales. Es decir:

$$B(a, k) = P_b$$

La ejecución del programa deberá ser:

```
$ java P4 <a> <pb>
```

Para este apartado, puede serle útil parte del código del primer punto de control.

Para que puedan probar si el funcionamiento es el adecuado, ejemplo de resultado de la ejecución son:

```
$ java P4 2 0.1  
0.595433  
$ java P4 5 0.001  
0.762115  
$ java P4 14 0.001  
5.446371
```

Punto de control 3: Muestre al profesor el programa indicado.