MÓDULO PROFESIONAL PROGRAMACIÓN

UD 6 – Arrays
Línea de comandos

ARGUMENTOS DE LÍNEA DE COMANDOS

¿Qué es un argumento de línea de comandos?

ARGUMENTOS DE LÍNEA DE COMANDOS

Argumento de línea de comando es la información que se pasa al programa cuando ejecuta.

La información pasada se almacena como un **vector de cadenas** en el método **main**. Más tarde, podemos usar los argumentos de línea de comandos en nuestro programa.

ARGUMENTOS DE LÍNEA DE COMANDOS

En las aplicaciones Java, al ejecutarlas, se pueden especificar argumentos en la línea de comandos (línea de llamada).

Ejemplo si tenemos un programa que se llama Ordenar que ordena 5 números enteros, pero en lugar de leerlos por teclado los números se le pasan al programa como argumentos, este programa lo ejecutaríamos así:

C:\> java Ordenar 4 3 1 9 -1

Los valores que se envían se deben escribir a continuación del nombre del programa y separados por un espacio en blanco.

ARGUMENTOS DE LÍNEA DE COMANDOS

- No hay restricciones en el número de argumentos de la línea de comandos de Java. Puede especificar **cualquier cantidad de argumentos**
- La información se pasa como **cadenas**, pero ¿en dónde?

ARGUMENTOS DE LÍNEA DE COMANDOS

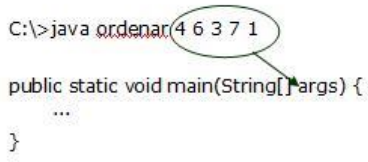
En el argumento args del método main. El array **array args** que aparece como argumento del método **main** es el encargado de recoger y almacenar estos valores.

```
public static void main( String args[] ) {
. . .
}
```

Aunque se le suele dar el nombre args, no es obligatorio que este parámetro se llame así, podemos darle el nombre que queramos. Por ejemplo sería valido un método main escrito así:

```
public static void main(String[] argumentos){
    ...
}
```

ARGUMENTOS DE LÍNEA DE COMANDOS



Como args es un **array de objetos String** contendrá cada uno de estos valores como un String:

args[0]	"4"
args[1]	"6"
args[2]	"3"
args[3]	"7"
args[4]	"1"

ARGUMENTOS DE LÍNEA DE COMANDOS. EJEMPLO

```
public class HelloCommandLineArguments {

    public static void main( String[] args ){

        // Print the string "Hello, " on screen
        System.out.println("I am saying Hello to the people below.. ");

        // Check if a command line argument exists
        if(args.length == 0)
            return;

        // Display the arguments from the command line
        for(int counter = 0; counter < args.length; counter++){
            System.out.println("argument index " + counter + ": " + args[counter]);
        }
    }

}
```

Siempre que trabajemos con valores recibidos desde la línea de comandos debemos controlar el número de valores recibidos para evitar que se produzcan errores al procesar el array.

ARGUMENTOS DE LÍNEA DE COMANDOS. EJEMPLO

Al ejecutar el programa **HelloCommandLineArguments** usando el **java** pasando argumentos en la línea de comandos.

C:\> java HelloCommandLineArguments Pepe Luis Gaby

El resultado será

```
I am saying Hello to the people below..
argument index 0: Pepe
argument index 1: Luis
argument index 2: Gaby
```

ACTIVIDAD

1. Completa el siguiente código y ejecútalo.

```
public class Arithmetic {
  public static void main (String[] args) {
    int operand1, operand2;
    char theOperator;

    // Check if there are 3 command-line arguments in the
    //  String[] args by using length variable of array.
    if (args.length != 3) {
      System.err.println("Usage: java Arithmetic int1 int2 op");
      return;
    }

    // Convert the 3 Strings args[0], args[1], args[2] to int and char.
    // Use the Integer.parseInt(aStr) to convert a String to an int.
    operand1 = Integer.parseInt(args[0]);
    operand2 = ......

    // Get the operator, assumed to be the first character of
    //  the 3rd string. Use method charAt() of String.
    theOperator = args[2].charAt(0);
    System.out.print(args[0] + args[2] + args[1] + "=");

    switch(theOperator) {
      case ('-'): System.out.println(operand1 - operand2); break;
      case ('+'): ......
      case ('*'): ......
      case ('/'): ......
      default:
        System.err.println("Error: invalid operator!");
    }
  }
}
```

ACTIVIDAD. LEER ARGUMENTOS USANDO NETBEANS

2. Construye un proyecto en NetBeans/Visual Studio con la siguiente clase y ejecútalo.

```
public class MyCompute {

  public static void main(String[] args) {
    System.out.println("I am reading numbers as command line arguments.. ");

  // Check if a command line argument exists
    if(args.length != 2){
        System.out.println("Please enter two numbers!");
        return;
    }

   // Display the addition of the two numbers
    int int1 = Integer.parseInt(args[0]);
    int int2 = Integer.parseInt(args[1]);
    int additionResult = int1 + int2;
    System.out.println("Result of addition = " + additionResult);

   // Display the multiplication of the two numbers
    int multiResult = int1 * int2;
    System.out.println("Result of addition = " + multiResult);
   }

}
```

ACTIVIDAD

3. Escribe un programa llamado **SumDigits.java** para resumir los dígitos individuales de un entero positivo, dado en la línea de comando. La salida debe ser:

   java SumDigits 1 2 3 4 5

   The sum of digits = 1 + 2 + 3 + 4 + 5 = 15

ACTIVIDAD

4. Escribe un programa llamado **MediaEdades.java** que reciba los nombres y edades de varias personas como argumentos en la línea de comandos de la siguiente forma:

    java MediaEdades Mónica 12 Daniel 34 Carlos 23

Y que calcule la media de las edades recibidas y la muestre por pantalla

RECURSOS

- Curso youtube : Java desde 0
- Libro Java 9. Manual imprescindible. F. Javier Moldes Teo. Editorial Anaya
- App SoloLearn: Aprende a Programar. Curso Java