

SUMADOR - MiniMANUAL DE USUARIO

Este es un programa "Sumador", el cual realiza la función de adición de números presentados en diferentes formatos que provengan tanto de línea de comando como de archivos de texto.

Los formatos que analiza y representa son seis: números reales, decimales, romanos, octales, hexadecimales y de alta precisión.

Al introducir los valores numéricos, éstos pueden encontrarse en cualquier formato (ejemplo, sumar números romanos con decimales). El resultado será desplegado en el formato del primer valor que fue introducido.

Modularidad.

El programa fue dividido en cinco módulos para su desarrollo:

- **Leer de consola.** Este módulo (clase) tiene la función de recibir los datos (ya sea desde la consola o de un archivo de texto) y los almacena en un vector.
- **Analizador.** La función de este módulo es la de verificar que los parámetros de entrada sean válidos, es decir, que sea alguno de los formatos aceptados por el sumador para poder llevar a cabo la suma correctamente. Una vez hecho esto, en el vector llena un campo reservado para colocar el identificador de tipo de dato del que se trata.
- **Convertidor.** El módulo convertidor se encarga de transformar los valores introducidos en cualquier formato a uno en común que permite hacer la suma. Almacena el número transformado en un campo reservado que tiene cada dato dentro del vector.
- **Sumador.** Este otro módulo suma todos los elementos del vector en el formato en común, que es el que se encuentra en el campo alterado por el módulo "Convertidor".
- **Desconvertidor.** Por último, este módulo transforma el resultado de la suma al formato del primer número introducido por el usuario.

Instrucciones de uso:

1.- Para compilar el programa, basta con compilar el archivo "Ejecuta.java" (*javac Ejecuta.java*). La ejecución será a partir de ese archivo: *java Ejecuta*.

2.- Inmediatamente después aparecerá el mensaje siguiente "Dame los números o nombre de archivo a sumar"

3.- Ahora se puede ingresar *n* cantidad de números, de cualquier formato (entero, romano, octal, hexadecimal, flotante y de alta precisión), en cualquier orden y no importa si están en mayúsculas o minúsculas. Un *<enter>* proporciona una nueva línea para agregar uno más.

```
C:\Archivos de programa\Java\jdk1.5.0_06\bin>java Ejecuta
Dame los numeros o nombre de archivo a sumar:
2
XU
o24
H26
```

Figura 1

De igual forma se pueden introducir nombres de archivos de texto que contengan números enlistados por renglones, con la condición de que estén en el formato “*nombre_archivo.txt*”

```
Dame los numeros o nombre de archivo a sumar:
X
10
datosprueba.txt
H24
20
```

Figura 2

4.- Cuando se termine de teclear el último número a sumar se debe teclear <enter> y escribir la palabra “end” de esta manera, el sistema podrá realizar la suma total de los número ingresados.

```
Dame los numeros o nombre de archivo a sumar:
X
10
datosprueba.txt
H24
20
end
```

Figura 3

```
Dame los numeros o nombre de archivo a sumar:
X
10
datosprueba.txt
H24
20
end
Resultado: CXXVI
```

Figura 4

Es muy importante señalar que el resultado de la suma será presentado en el formato del primer número ingresado, como se demuestra en la Figura 9, el primer valor fue el X(10) en formato romano, por lo tanto el resultado final se presentará en este mismo formato.

Conclusión:

Este programa pretende ilustrar sobre la facilidad que es trabajar con módulos, y sobre todo la importancia que tiene el separar las funciones para hacerlas transparentes a los demás procesos. Las fallas pueden detectarse y repararse de forma más sencilla.

Cuando se trata de programas de miles de líneas de código, la modulación disminuye tiempos y costos, y para las empresas y los clientes esto resulta de vital importancia.

Notas de Compilación:

En caso de que la compilación marque error, se tiene que ajustar el *path* agregando una instrucción a la compilación. La siguiente *wildcard* compilará todas las clases:

```
javac -classpath . *.java
```

De igual manera, la ejecución es de manera similar:

```
Javac -classpath . Ejecuta
```