

1. Con la ayuda del **IDE NetBeans** crear un proyecto de tipo **Aplicación Java**, llamado **OperacionNumericaArgumento**, que contenga su punto de entrada en la clase **Main** que se encuentra en el paquete **org.japo.java.main**, que llame al método **launchApp** de la clase **App** que se encuentra en el paquete **org.japo.java.app**.

El programa debe de:

1. Definir las **constantes ENTERAS** de clase **N1** y **N2**.
2. Inicializadas en una sección **static** con sendos valores **aleatorios** situados entre un **MÍNIMO** y un **MÁXIMO** elegidos de forma **arbitraria** (1 .. 10).
3. Definir las **Constantes de Operación**:

```
public static final int OP_SUM = 0;      // Suma
public static final int OP_RES = 1;      // Resta
public static final int OP_MUL = 2;      // Producto
public static final int OP_DIV = 3;      // Cociente
public static final int OP_MOD = 4;      // Resto
public static final int OP_MED = 5;      // Media
public static final int OP_MAY = 6;      // Mayor
public static final int OP_MEN = 7;      // Menor
```

4. Definir un método llamado **operar**, que debe **recibir** dos argumentos llamados **n1** y **n2**, y dependiendo del tercero que indica la operación a realizar con los argumentos, **analizar** la relación de magnitud entre ellos y **devolver** el **mayor** de ellos, con la siguiente **signatura**:

```
public static final double operar(double n1, double n2, int op);
```

5. Llamar al método **analizar** con los **parámetros** correspondientes.
6. **Mostrar** por pantalla el proceso en el siguiente **formato**:

```
OPERACIÓN NUMÉRICA
=====
Número 1 ....: 3
Número 2 ....: 7
---
El MAYOR es .: 7
```

2. Con la ayuda del **IDE NetBeans** crear un proyecto de tipo **Aplicación Java**, llamado **EntradaEnteroArgumento**, que contenga su punto de entrada en la clase **Main** que se encuentra en el paquete **org.japo.java.main**, que llame al método **launchApp** de la clase **App** que se encuentra en el paquete **org.japo.java.app**.

El programa debe de:

1. **Definir** la **constante de clase MSG_USR** inicializada con el mensaje que debe indicarse al usuario para invitarle a que introduzca un número por teclado.
2. **Definir** la **constante de clase MSG_ERR** inicializada con el mensaje que debe indicarse al usuario para informarle de que ha habido un error en el proceso de introducción de un número por teclado.
3. **Definir** un método llamado **obtener**, que debe gestionar la **petición al usuario** de un **número** y su posterior **devolución**, en el bien entendido de que en caso de error se debe de repetir la petición, con la siguiente **signatura**:

```
public static final double obtener(String msgUsr, String msgErr);
```
4. **Llamar** al método **obtener** para acceder al **número introducido por el usuario**.
5. **Informar** finalmente del proceso en el siguiente formato:

```
INTRODUCCIÓN ENTERO
=====
Introducir número ....: sirtopwe
ERROR: Entrada incorrecta
---
Introducir número ....: 6
---
Número introducido ...: 6
```

```
INTRODUCCIÓN ENTERO
=====
Introducir número ....: 5.6
---
Número introducido ...: 5
```

3. Con la ayuda del **IDE NetBeans** crear un proyecto de tipo **Aplicación Java**, llamado **AnálisisRangoArgumento**, que contenga su punto de entrada en la clase **Main** que se encuentra en el paquete **org.japo.java.main**, que llame al método **launchApp** de la clase **App** que se encuentra en el paquete **org.japo.java.app**.

El programa debe de:

1. **Definir** la **constante de clase NUM** inicializada con el valor correspondiente a un número arbitrario.
2. **Definir** las **constantes de clase MIN y MAX** inicializadas con los valores correspondientes a un rango arbitrario.
3. **Definir** un método llamado **analizarRango**, que recibe un número y los límites de un rango, y que debe de devolver un valor numérico que codifique:
 - Número **DENTRO** del rango: 0
 - Número **FUERA (Debajo)** del rango: 1
 - Número **FUERA (Encima)** del rango: 2con la siguiente **signatura**:

```
public static final int analizarRango(double num, double min, double max);
```

4. **Llamar** al método **analizarRango** con las constantes definidas previamente para el **número en el rango**.
5. **Informar** del estado de pertenencia del número al rango, en el formato:

```
ANÁLISIS DE RANGO
=====
Número .....: -3
---
Mínimo .....: 1
Máximo .....: 10
---
Análisis ....: FUERA ( Abajo )

ANÁLISIS DE RANGO
=====
Número .....: 5
---
Mínimo .....: 1
Máximo .....: 10
---
Análisis ....: DENTRO

ANÁLISIS DE RANGO
=====
Número .....: 50
---
Mínimo .....: 1
Máximo .....: 10
---
Análisis ....: FUERA ( Arriba )
```

4. Con la ayuda del **IDE NetBeans** crear un proyecto de tipo **Aplicación Java**, llamado **ValidacionDNIArgumento**, que contenga su punto de entrada en la clase **Main** que se encuentra en el paquete **org.japo.java.main**, que llame al método **launchApp** de la clase **App** que se encuentra en el paquete **org.japo.java.app**.

El programa debe de:

1. **Definir** la **constante de clase LETRAS** inicializada con la **secuencia de letras**:

```
public static final String LETRAS = "TRWAGMYFPDXBNJZSQVHLCKE";
```
2. **Definir** la **constante de clase DNI_NUM** inicializada con un **número entero** correspondiente a un DNI arbitrario.
3. **Definir** la **constante de clase DNI_CTR** inicializada con un carácter alfabético correspondiente a un DNI arbitrario.
4. **Definir** un método llamado **validar**, que debe analizar y devolver el estado de correspondencia del numero y la letra suministrados a un **DNI correcto** [true] o **DNI incorrecto** [false], con la siguiente **signatura**:

```
public static final boolean validar(int num, char ctr);
```
5. **Definir** un método llamado **calcular**, que debe calcular y devolver la letra de control asociada al número de DNI suministrado, con la siguiente **signatura**:

```
public static final char calcular(int num);
```
6. **Llamar** al método **validar** con las constantes **DNI_NUM** y **DNI_CTR** para averiguar si se corresponden con un DNI correcto o no.
7. **Mostrar** el resultado del análisis por pantalla en el **formato**:

```
VALIDACIÓN DE DNI
=====
DNI .....: 12345678Z
---
Validez ..: Correcto

VALIDACIÓN DE DNI
=====
DNI .....: 12345678X
---
Validez ..: Incorrecto
```