

# Tarea para PROG04.

Detalles de la tarea de esta unidad.

## **Enunciado.**

Esta cuarta unidad se ha centrado en el estudio de las estructuras de control de flujo que Java pone a disposición del programador/a. Posteriormente, la unidad se ha enfocado hacia el ámbito de las excepciones en el código fuente. Se tratan cuestiones sobre cómo capturar dichas excepciones, su manejo y posible delegación.

En la misma línea, se ha continuado más tarde con la prueba de programas, detallando los tipos de pruebas a que pueden someterse y cerrando esta parte con la estandarización de las pruebas que puede realizarse a través del Framework JUnit.

A continuación se ha visto la importancia de la depuración de programas para la detección y corrección de errores y para finalizar la unidad se ha atendido a una cuestión tan importante como es la documentación del código y el uso de herramientas estandarizadas para ello.

Para la realización de esta tarea se han propuesto 3 ejercicios en los que deberéis de utilizar las distintas estructuras de control vistas en la unidad. Además se deberá tratar los posibles errores que se puedan producir en la entrada de datos.

## **Ejercicio 1**

Escribe un programa que lea una secuencia de notas que se van introduciendo por el teclado (las notas serán valores numéricos que pueden contener decimales). La secuencia de notas finalizará cuando se introduzca un valor negativo. El programa informará al final del número de aprobados, el número de suspensos y la nota media. El programa deberá controlar los posibles errores que se puedan producir a la hora de introducir las notas por teclado de tal forma que si alguna nota introducida fuera errónea la vuelva a solicitar.

## **Ejercicio 2**

Diseña un programa que imite a una calculadora capaz de realizar las operaciones básicas (suma, resta, producto y multiplicación) y dos operaciones complejas como la potencia y la raíz cuadrada, la aplicación debe mostrar inicialmente un menú similar al siguiente:

## **Emulador Calculadora**

Elige una opción:

- 1.- Operaciones básicas.
- 2.- Operaciones complejas.
- 3.- Salir.

En el caso de la opción 1 se deberá mostrar:

- a. Suma.
- b. Resta.
- c. Producto.

d. División.

En el caso de la opción 2 se deberá mostrar:

a. Potencia.

b. Raíz cuadrada.

Una vez elegida la opción correspondiente el programa deberá solicitar los datos necesarios y mostrar por pantalla el resultado obtenido en cada caso. El programa deberá controlar los posibles errores que se puedan producir a la hora de elegir la opción y de introducir los datos por teclado de tal forma que si los datos introducidos fueran erróneos los vuelva a solicitar.

### Ejercicio 3

Crea un proyecto que contenga una clase `Libro` y una clase `Principal` en un paquete llamado `biblioteca` y pruebe todos sus métodos.

- La clase `Libro` contendrá tres atributos `private` de tipo `String`:

- `titulo`: el título del libro.
- `autor`: el autor del libro.
- `isbn`: el código ISBN del libro.

Y los siguientes métodos:

- (0.5 puntos) `public Libro(String titulo, String autor, String isbn)`: El constructor de la clase inicializa los atributos del objeto con los datos pasados por parámetro. Es importante comprobar que el objeto se genera adecuadamente por lo que si el ISBN pasado por parámetro no es válido, se lanzará la excepción `IllegalArgumentException`. Así se genera la excepción:  

```
throw new IllegalArgumentException("isbn no válido");
```
- (1 punto) Métodos `consulta_XXX()` y `cambia_XXX()`: se creará un método `consulta/cambia` para cada uno de los atributos. El método `cambia_isbn()` deberá comprobar la validez y en caso de que no sea válido generará la excepción `IllegalArgumentException` (similar al constructor).
- (1 punto) `private boolean compruebaIsbn10(String isbn)`: Este método se utilizará en el constructor y en `cambia_isbn()` para comprobar la validez del ISBN. Devolverá `true` si el parámetro es un ISBN-10 válido, y `false` en caso contrario.
- (1 punto) `private boolean compruebaIsbn13(String isbn)`: Este método se utilizará en el constructor de la clase y en `cambia_isbn()` para comprobar la validez del ISBN. Devolverá `true` si el parámetro es un ISBN-13 válido, y `false` en caso contrario.
- (0.75 puntos) La clase `Principal` contendrá la función `main` que hará uso de los métodos públicos de la clase `Libro`. Para comprobar si un isbn introducido es válido, habrá que capturar la excepción `IllegalArgumentException` en la llamada al constructor (bloque `try-catch`).
- (0.75 puntos) Añade las etiquetas `javadoc` necesarias para documentar la clase `Libro` adecuadamente.

### Cómo comprobar la validez de un código ISBN

El código ISBN de un libro puede contener 10 o 13 caracteres (desde 2007). El ISBN es un código de detección de errores por lo que es posible comprobar la validez del mismo. A continuación se explica como:

**ISBN-10:** Debemos multiplicar cada dígito (si el último carácter fuera una X, su valor es 10) por la posición que ocupa y calcular la suma de todos ellos. Si la suma obtenida es múltiplo de 11, el isbn es válido.

Ejemplo: Comprueba la validez del siguiente isbn-10: 84-481-2231-3

- Suma:  $8*1 + 4*2 + 4*3 + 8*4 + 1*5 + 2*6 + 2*7 + 3*8 + 1*9 + 3*10 = 154$
- Comprobación de que la suma es múltiplo de 11:  $154 / 11 = 14$  (el resto es 0)

**ISBN-13:** Debemos multiplicar cada número por 1 si la posición que ocupa es impar o por 3 si la posición que ocupa es par y calcular la suma de todos salvo del último que es el dígito de control. Dividimos el resultado de la suma anterior entre 10 y obtenemos el resto de la división. Restamos a 10 el resto obtenido. El isbn será válido si el valor obtenido es igual al dígito de control (último dígito del isbn).

Ejemplo: Comprueba la validez del siguiente isbn-13: 978-84-415-2682-2

- Suma:  $9*1 + 7*3 + 8*1 + 8*3 + 4*1 + 4*3 + 1*1 + 5*3 + 2*1 + 6*3 + 8*1 + 2*3 = 128$
- Cálculo del dígito de control:  $128 \bmod 10 = 8$  (el resto).  $10 - 8 = 2$  (coincide con el dígito de control)

**Recursos:**

<http://gaussianos.com/detectar-si-un-isbn-es-erroneo/>

<http://www.monografias.com/trabajos42/codigo-de-barras/codigo-de-barras3.shtml>

### **Criterios de puntuación. Total 10 puntos.**

- Ejercicio 1: 2 puntos.
- Ejercicio 2: 3 puntos.
- Ejercicio 3: 5 puntos.

Total: 10 puntos

### **Recursos necesarios para realizar la Tarea.**

- Ordenador personal con un procesador de textos que pueda generar archivos pdf.
- Entorno de desarrollo NetBeans.
- Conexión a Internet.
- Navegador Web.
- Programa de impresión en formato pdf.

### **Consejos y recomendaciones.**

El orden de realización de los supuestos no importa.

El código fuente de cada uno de los supuestos debe estar debidamente identificado y separado de los demás.

Se recomienda que antes de enviar el código fuente hayas realizado la compilación y ejecución de los programas, de este modo evitarás posibles fallos.

Utiliza comentarios a lo largo de tu código.

Utiliza tabulaciones y estructura tu código para que sea lo más legible posible.