

TAREA PARA PROG04

Enunciado.

A continuación, se enumeran varios supuestos que el alumnado debe solventar a través de la realización un programa por cada uno de los supuestos propuestos, todos incluidos **en un mismo proyecto de NetBeans** (diferentes clases, que tendrán main aquellas que las necesiten, y cada supuesto en un paquete diferente dentro del mismo proyecto. Los paquetes se crean con botón derecho → New → JavaPackage dentro del Source Package del proyecto.)

Además, se debe **crear un documento**, que **acredite identidad**. Para ello basta con **algún pantallazo, del proyecto y ejercicios desplegados, ejecutados y completos**, y que en el fondo salga vuestro Nombre y Apellidos dentro del aula virtual. Algo breve y conciso, sólo se trata de mostrar la identidad junto con vuestro trabajo **ejecutado viendo resultados**, no de explicar paso a paso, ya que también me entregaréis el código para que pueda comprobar que funciona.

Consideraciones sobre el código que implementes:

* El código tiene que estar suficientemente comentado y bien indentado (también conocido como sangrado), si no, no se corregirá por ilegibilidad del mismo.

Como viste en el apartado de javadoc, si escribes `/**` y pulsas intro, automáticamente se crea una zona de comentarios en las que aparece el autor, y podrías añadir más datos, como la versión, fecha o lo que veas oportuno. Es conveniente hacerlo antes de cada clase, e incluso de cada método.

Como mínimo hay que poner un comentario al principio de cada clase, con tu nombre y apellidos, indicando el tema en el que estamos. Todo esto es requisito para continuar la corrección.

Supuesto 1 (1 punto):

Realizar una clase llamada `DeportesConIF_Nombre` (Nombre es vuestro nombre), que llevará un método `main()`, en el que se implementará un algoritmo que lea por teclado la temperatura y escriba en la salida cuál es el deporte apropiado para practicar según los siguientes datos (**no** se puede utilizar **switch case**):

Natación $T > 30^{\circ}$
Golf $15^{\circ} < T \leq 30^{\circ}$
Tenis $5^{\circ} < T \leq 15^{\circ}$
Esquí $-10^{\circ} < T \leq 5^{\circ}$
Damas $T \leq -10^{\circ}$

Si se introduce una temperatura no numérica, al intentar convertir a número saltará una excepción que tiene que controlarse (tal y como se hace en el Ejemplo_primer de Excepciones, del pdf que dejé en el apartado Contenidos del tema 4).

Supuesto 2 (1 punto):

Este es casi igual, pero con switch case.

Hay que realizar una clase llamada `DeportesConSwitchCase_Nombre` (Nombre es vuestro nombre), que llevará un método `main()`, en el que se implementará un algoritmo que lea por teclado la temperatura y escriba en la salida cuál es el deporte apropiado para practicar según los siguientes datos (en este caso **sí** hay que utilizar **switch case**):

Natación $T > 30^{\circ}$
Golf $15^{\circ} < T \leq 30^{\circ}$
Tenis $5^{\circ} < T \leq 15^{\circ}$
Esquí $-10^{\circ} < T \leq 5^{\circ}$
Damas $T \leq -10^{\circ}$

Si se introduce una temperatura no numérica, al intentar convertir a número saltará una excepción que tiene que controlarse (tal y como se hace en el Ejemplo_primer de Excepciones, del pdf que dejé en el apartado Contenidos del tema 4).

Supuesto 3 (1 punto):

Hay que realizar una clase llamada `SumaNaturalesConFor_Nombre` (Nombre es vuestro nombre), que llevará un método `main()` que calcule la suma de los N primeros números naturales. N se leerá por teclado. Se realizará con un bucle FOR. En cada iteración del bucle se debe ir mostrando la suma parcial.

En este ejercicio el control de las excepciones es voluntario.

Supuesto 4 (1 punto): (similar al anterior, pero con while y sin mostrar parciales)

Hay que realizar una clase llamada `SumaNaturalesConWhile_Nombre` (Nombre es vuestro nombre), que llevará un método `main()` que calcule la suma de los N primeros números naturales. N se leerá por teclado. Se realizará con un bucle WHILE. En cada iteración del bucle NO se debe ir mostrando la suma parcial, se mostrará la suma sólo al final.

En este ejercicio el control de las excepciones es voluntario.

Supuesto 5 (1 punto):

Hay que realizar una clase llamada `SumaNumeros(Nombre es vuestro nombre)`, que llevará un método `main()` en la que dada una secuencia de números leídos por teclado, que acabe con un `-1`, el programa tras leer dichos números muestre la suma de todos ellos (a excepción del número -1, el cual es escrito solo para finalizar el programa). Puedes utilizar la instrucción iterativa que veas más oportuna para realizar el ejercicio. (For, While o Do)

En este ejercicio el control de las excepciones es voluntario.

Supuesto 6 (5 puntos):

*Realizar una clase llamada **Ecuacion_Nombre** (Nombre es vuestro nombre), para calcular las soluciones de una ecuación de segundo grado $ax^2 + bx + c = 0$, cuyas soluciones se calculan así:

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

(una solución se calcula sumando la raíz del discriminante y la otra restando la raíz del discriminante)

Sea $(b^2 - 4 \cdot a \cdot c)$ el **discriminante**.

Cuando lo necesites, para calcular la raíz cuadrada utilizaremos **Math.sqrt** y para la potencia **Math.pow**.

La clase **Ecuacion_Nombre** tendrá los siguientes atributos:

a: int

b: int

c: int

Deberás crear además **2 constructores**, uno, el constructor por defecto, en el que se inicialicen todos los valores a cero, y otro en el que se inicialicen a, b y c, con los valores que introduzca el usuario al crear el objeto. (similar a como se vio en la tarea de la unidad anterior)

También crearás los **métodos** para **modificar** y **consultar** el valor de esos atributos (similar a la unidad anterior)

Además, habrá un método, cuya cabecera es así, que calculará una de las soluciones:

```
double calcularX1(int a, int b, int c) throws ExcepcionDiscriminanteNegativo
```

Este método calculará y devolverá la solución de la ecuación resultante de **sumar** la raíz del discriminante. (una de las soluciones de la ecuación).

Como ves, lanza una excepción que tendrás que crear y que viene explicada más adelante.

✚ Además, habrá otro método similar, cuya cabecera es así, que calculará la otra solución.

```
double calcularX2(int a, int b, int c) throws ExcepcionDiscriminanteNegativo
```

Este método calculará y devolverá la solución de la ecuación resultante de **restar** la raíz del discriminante. (la otra de las soluciones de la ecuación)

Como ves, lanza una excepción que tendrás que crear y que viene explicada más adelante.

✚ Ambos **métodos**, antes de intentar calcular la solución llamarán a otro método:

```
int comprobarDiscriminante (int a, int b, int c) throws ExcepcionDiscriminanteNegativo
```

Este método **calculará el discriminante** ($b^2 - 4 \cdot a \cdot c$)

→ Si el discriminante es mayor o igual a cero, devolverá el valor calculado.

Pista: utiliza para ese cálculo Math.pow y cuidado con los paréntesis.

→ Si el discriminante es negativo lanzará una nueva excepción, **ExcepcionDiscriminanteNegativo**, con el mensaje "EL DISCRIMINANTE ES NEGATIVO: NO HAY SOLUCIONES REALES"

✚ Para ello, previamente tienes que crear la clase **ExcepcionDiscriminanteNegativo** (se crea similar al ejemplo completo que hay en el pdf que os pasé y además os pasé los archivos .java).

Esa clase simplemente tiene un constructor que recibe un mensaje, que pasa al super (msg), como se vió en el PDF.

Con todo esto estamos aprendiendo a realizar un control de nuestras excepciones. Parece lioso, pero en cuando leas estos pasos dos o tres veces y lo entiendas como un funcionamiento conjunto, verás que bien está y la gran utilidad de las excepciones. No dudes en preguntarme por mensajería privada en este caso, si tienes dudas haciendo esta parte. Es la manera de aprender.

✚ Ya por fin, en el **main**, probaremos todo. Para ello, deberás pedir los valores de **a, b y c**, y crear un objeto de tipo **Ecuacion_Nombre**. (si no quieres pedirlos por pantalla me vale con que crees unas variables en el main, que asignes los valores, y con esos valores crees el objeto, utilizando el segundo constructor, de manera similar a como hicimos en la unidad anterior).

Posteriormente llamarás al método **calcularX1** y mostrarás el resultado de la primera solución, y al método **calcularX2** para mostrar la segunda.

Para una ecuación como esta: $3X^2 + 2X - 5$

Se mostrarían dos mensajes como este.

"La primera solución de la ecuación de segundo grado a=3, b=2 c=-5 es: 1"

"La segunda solución de la ecuación de segundo grado a=3, b=2 c=-5 es: 1.66666666666667"

Si al llamar a uno de estos métodos el discriminante fuese **negativo**, hay que controlar la **excepción** que habrán lanzado dichos métodos y se mostrará un mensaje desde un **primer catch**, que por consola indique ese hecho.

"EL DISCRIMINANTE ES NEGATIVO: NO HAY SOLUCIONES REALES."

Por ejemplo, para una ecuación como: $3X^2 + 2X + 5$

Se mostraría el mensaje:

"EL DISCRIMINANTE ES NEGATIVO: NO HAY SOLUCIONES REALES."

Además, en el main capturaremos cualquier otra **excepción**, capturando una **Excepción** general (Exception).

Aquí va el esqueleto de cómo os tiene que quedar el main

```
.....
int a,b,c;
a=3;
b=2;
c=-5;
try{
    Ecuacion_Nombre miEcuacion=new Ecuacion_Nombre(a,b,c)
    .....
    AQUÍ LLAMAS A LOS MÉTODOS QUE CALCULAN LAS 2 SOLUCIONES Y LAS MUESTRAS
    .....
}
catch(ExcepcionDiscriminanteNegativo.....){
    .....
    SI EL DISCRIMINANTE ES NEGATIVO LA EXCEPCION SALTARÁ AQUÍ Y MUESTRAS SU MENSAJE e.getMessage()
    ASÍ HEMOS CONTROLADO NUESTRA PROPIA EXCEPCIÓN
    .....
}
catch(Exception ..... ){
    .....
    SI SALTA OTRO TIPO DE EXCEPCION DEL SISTEMA SE CAPTURA AQUÍ Y SE MUESTRA SU MENSAJE
    .....
}
}
```

Puntuación de este ejercicio:

- 1 punto:** Creación de la clase, atributos, constructores, métodos de consulta y asignación...
- 1 punto:** Creación de métodos de cálculo con control de excepciones.
- 1 punto:** Creación de método que calcula el discriminante.
- 1 punto:** Creación de la Clase **ExcepcionDiscriminanteNegativo**.
- 1 punto:** Implementación correcta el main con control de excepciones.

ENTREGA

Criterios de puntuación.

Supuestos 1 al 5: 1 punto cada uno.

Supuesto 6: 5 puntos.

Total: 10 puntos

Recursos necesarios para realizar la Tarea.

Sistema Operativo.

Ordenador personal con un procesador de textos.

Entorno de desarrollo NetBeans.

Conexión a Internet.

Navegador Web.

Consejos y recomendaciones.

Dentro de la carpeta de NetBeans, se habrá creado automáticamente la carpeta del proyecto.

Comprueba que efectivamente está la carpeta y dentro todo el contenido que has creado.

Para entregar, comprime en un único archivo la carpeta del proyecto con todo su contenido, junto con el documento y pantallazos que demuestran la identidad.

Indicaciones de entrega.

Se enviará **un único archivo comprimido** que contendrá el **proyecto**, y un **documento** PDF, con los **pantallazos** que **muestran** los ejercicios **ejecutados y completos**, junto con vuestra identidad, poniendo de fondo por ejemplo la pantalla del aula virtual. (no se trata de explicar paso a paso, sino, ver que está todo hecho, y se ve vuestra identidad del aula virtual en los pantallazos).

El envío se realizará a través de la plataforma y el archivo se nombrará siguiendo las siguientes pautas:
apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la tercera unidad del MP de PROG**, debería nombrar esta tarea como...

sanchez_manas_begona_PROG04_Tarea.