

TAREA PARA ED03.

ENUNCIADO.

En este tema se trata aprenderemos a depurar código fuente para ir evaluando el valor de las variables y así detectar posibles errores.

También crearemos casos de pruebas automáticos con JUnit.

En esta tarea se adjunta el código fuente de dos clases Java:

- **Calculador.java**

contiene una clase con dos métodos.

- **Tarea3Main.java**

contiene una pequeña aplicación de ejemplo, que hace uso de los métodos anteriores.

El propósito de la tarea es realizar tareas de **depuración y prueba**.

En el caso de las pruebas, se trata de realizar las **pruebas unitarias** de la clase Calculador, usando el framework **JUnit**.

Importante: Tienes que ir dando **pantallazos** de **todos** los pasos que vienen a continuación. Opcionalmente, y sustituyendo a la elaboración del documento en pdf. En ambos (elijas el que elijas) se tiene que ver tu identificación del aula virtual de fondo, pantalla completa viéndose tu escritorio. Si optas por el vídeo tienes que ir hablando, comentando lo que realizas. Si algún apartado no se realiza, hay que indicarlo en el documento o en el vídeo correspondiente.

1. Crear un proyecto en Apache NetBeans cuyo nombre sea "**Tarea3_TuNombre**" e incorpora las dos clases proporcionadas en el paquete **entornos.tema3** (**renombra el paquete que viene en las clases que te he pasado, para que funcione**)

Ejecución normal

2. Compilar y ejecutar la aplicación, viendo la salida que da por consola.
 - a. Realiza **dos o tres** ejecuciones, en las que pruebes diferentes números para ver que se **cuentan las cifras correctamente**, y se realizan bien las **potencias** (la aplicación no controla excepciones, no puedes poner un número no entero ni mayor del máximo)

Depuración

3. Establecer dos **breakpoints**:
 - a) uno en la línea 31 de Tarea3Main.java (la que pone "INICIO DE EJECUCIÓN").
 - b) otro en la línea 53 de Tarea3Main.java (la que pone "FIN DE EJECUCIÓN--").
4. Iniciar una sesión con el **depurador** para ejecutar paso a paso la aplicación.
 - c) La ejecución se **detendrá** en el primer breakpoint (línea 31).
 - d) Una vez se empieza a depurar, se tienen que tener **visibles** las ventanas de depuración (**Variables y Watches** sobre todo).
 - e) Añade a **Watches**, la siguiente **expresión**: base+expo (no es que tenga sentido sumar base y exponente, pero más adelante se utilizará para un ejemplo)
 - f) Avanza **paso a paso** hasta la línea 36 (la llamada al método cuentaCifras). En uno de los pasos, habrás tenido que introducir un número por teclado.
 - g) **Inspecciona** las variables en la ventana Variables (**num** tiene que tener el número introducido).
 - h) Avanza paso a paso **sin entrar en el método** hasta la línea 38.

- i) Verás que el valor de la variable **numDigitos** ha cambiado, ya que se ha ejecutado el método que ha contado las cifras y que van apareciendo poco a poco mensajes por consola.
- j) Avanza hasta la línea 41. El valor de la variable **num** ha cambiado (tiene que tener un valor muy grande, el correspondiente a Integer.MAX_VALUE).
- k) Ahora avanza paso a paso **entrando en el método** cuentaCifras.
- l) Verás cómo ha cambiado el **contexto** de las variables. (ahora se ven las del método en el que estás).
- m) Avanza paso a paso dentro del método hasta que hayas ejecutado el bucle **while** todas las veces. Verás cómo cambió la variable num, y en cada iteración del while, van cambiando el valor de la variable **cifras** en el que se van contando las cifras del número y el de la variable **aux**.
- n) Sigue paso a paso. Volverás al **contexto** del **main**. Avanza paso a paso hasta la **línea 44**, y verás cómo han aparecido mensajes por consola y las variables del contexto del main, ya se han actualizado con las que han obtenido del cuentaCifras.
- o) **Reanuda** la ejecución hasta llegar al 2º breakpoint (ya no vayas paso a paso, pusta en la opción Continue (f5 o circulito verde con triángulo interior blanco).
- p) Verás que por consola se te pide el valor de la **base** y del **exponente**, y una vez los introduzcas continúa la ejecución, deteniéndose en el otro punto de ruptura que pusimos, no has tenido que ir ya paso a paso. (eso muchas veces es útil, para avanzar en la ejecución si estás evaluando variables, pero no quieres salirte del todo de la depuración)
- q) **Entre tanto**, se habrá calculado ya, el valor de la potencia y lo habrá mostrado por consola.
- r) Ahora **cambiaremos** el valor de las **variables** en tiempo de ejecución.
 - No significa que estés cambiando el código, pero sí el valor en esa ejecución para hacer pruebas.
 - Para ello observa el valor que tiene la expresión que añadimos a Watches (base + exponente).
 - Ahora ve a la ventana Variables, y cambia el valor de la variable base.
 - Verás ahora que el valor de la expresión en Watches ha cambiado.
- s) Ya **estamos llegando al final**. Para terminar una depuración, podrías seguir paso a paso, hasta que termine, darle a continuar, o directamente darle a finalizar la depuración (cuadradito rojo con cuadradito blanco interno). Según sea el caso, es más útil hacer una u otra cosa, dependiendo de si queremos seguir inspeccionando, o si ya no nos interesa en cuyo caso lo mejor es finalizar.
- t) **Finaliza la sesión de depuración** de la manera que desees de las explicadas anteriormente.

Pruebas

5. Crear una Aplicación JUnit para la clase Calculador. Utiliza la estrategia de **caja blanca** para diseñar los casos de prueba. (para este ejercicio, además de los CONTENIDOS interactivos de la plataforma, conviene estudiar bien el anexo que se ha puesto en el apartado CONTENIDOS de la unidad)

Piensa en casos de prueba significativos.

- a) Escribe **cuatro casos de prueba** para el método **cuentaCifras**. En el que pruebes distintas cosas (rangos, negativos, cero etc.)
- b) Escribe **tres casos de prueba** para el método potencia. (igualmente que haya variedad).
- c) **Ejecuta la aplicación de prueba**, visualizando la consola y el **Panel de Resultados**. No deberían aparecer defectos en la implementación de la clase Calculador. (todo **verde**)
- d) Ahora **fuerza algún error** en el método **cuentaCifras** y en el método **potencia** para que den mal el resultado.

- e) **Ejecuta de nuevo los casos** de prueba en el que se vea que detecta el error. (saldrá rojo e indica el mensaje si despliegas el panel del error)

Importante: El caso de prueba tiene que ser correcto, no es ahí donde tienes que forzar el error. Es en el propio código donde has tenido que forzar el error para realizar simular la realidad: El probador diseña un caso de prueba con unas entradas y esperando unos resultados correctos, y si el método está mal, no genera los resultados esperados, por lo tanto, JUnit nos lo indica, y habría que pasar una incidencia al responsable que programó este método que lo revise. (en este caso eres tú mismo, pero en el caso de una empresa, quien prueba no suele ser la persona que ha realizado el código).

ENTREGA

Tendrás que entregar a tu tutor un documento con las **suficientes capturas de pantalla** y explicaciones, que justifiquen que se han realizado correctamente todos los apartados. Como se decía anteriormente se puede **sustituir por un vídeo**.

Si algún apartado no se realiza, hay que indicarlo en el documento o en el vídeo correspondiente.

Las capturas para el documento o bien el vídeo tienen que ser del **escritorio completo**, y deben mostrar la **identidad** de la persona que las realiza. Para ello, se tendrá **abierto el aula virtual** y saldrá la **identificación** personal del autor en parte del pantallazo. Comprueba que la **calidad de la captura es óptima**. No se corregirán capturas borrosas.

Si optas por realizar un **documento en PDF**, se tiene que añadir una **portada con tu nombre** y alguna imagen y el formato de todo el documento debe ser correcto. No es necesario que haya muchos colores ni cosas estridentes, pero sí que se cumplan unos mínimos de alienaciones, fuentes, saltos de página, distinguir bien los apartados y diferenciarlos de las respuestas etc. Es muy difícil comprender algunos documentos que se reciben, y por eso se pide que se cuide la entrega.

CRITERIOS DE PUNTUACIÓN. TOTAL 10 PUNTOS.

Cada error u omisión importante penalizará entre con 0.25, 0.5, 0.75 o 1 punto. Depende de la importancia del mismo a criterio del profesor.

Si las imágenes se ven borrosas, se detecta plagio, no se demuestra la identificación, el archivo está corrupto o en blanco, o no se cumple el formato la práctica será calificada con cero puntos.

RECURSOS NECESARIOS PARA REALIZAR LA TAREA.

JDK (última versión): Descarga del JDK

NetBeans (preferiblemente últimas versiones): Descarga de Netbeans

Ordenador con Windows instalado.

Conexión a internet.

INDICACIONES DE ENTREGA.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre.

En este ejercicio no hay que entregar código fuente. Sólo el documento y las pruebas de haberlo realizado.