

Ciclo formativo y módulo: Desarrollo de Aplicaciones Web / Multiplataforma

Módulo: Entornos de Desarrollo (EDDE)

Unidad 4: Optimización y documentación

Cogeremos como fichero base aquel trabajado de la tarea 6 con vuestras modificaciones y comentarios; trabajaremos con el IDE NetBeans.

Tarea

DOCUMENTACIÓN JAVADOCS

1. Lanza el analizador de javadocs para el package creado. Genera automáticamente las opciones que JavaDocs te propone. Completa los comentarios generados y crea (si procede y si no están) los comentarios Javadocs siguientes:

@author y @version para la clase main.
@param y @return para cada método

Actualiza el análisis Javadocs para comprobar que no faltan.

2. Genera la documentación JavaDoc para el proyecto y comprueba que abarca todos los métodos y atributos de la clase main.

¿Cuál es la extensión del fichero generado?

¿A quién está destinado este documento? ¿De qué tipo de documentación se trata?

¿Todos tus comentarios de la tarea 6 han sido añadidos al fichero?

Adjunta el fichero generado en la entrega de la tarea.

REFACTORIZACIÓN

Usando las opciones de refactorización de Netbeans, realiza los siguientes cambios y comprueba los resultados obtenidos:

3. Cambia el nombre de la variable “pass” del método *main()* por “passIntroducido”.
¿Ha cambiado ese nombre en los argumentos de los otros métodos?

4. Para cada uno de los 4 métodos de comprobación de la contraseña, selecciona el texto de ayuda de la sentencia “*system.out.println()*”, aplica las refactorizaciones siguientes y comenta su resultado:

Texto del método *length()*: opción “introduce->variable”

Texto del método *containsDigits()*: opción “introduce->constant”

Texto del método *containsUpLowerCase()*: opción “introduce->field”

Texto del método *hasSpecial()*: opción “introduce->parameter”

5. Encapsula una de las variables creadas. ¿Qué cambios ves en el código?
6. Selecciona la palabra “false” del método *passwordIsGood()* y refactoriza con la opción “invertboolean”, indicando un nuevo nombre de método más coherente que este cambio.

¿Qué cambios ves en el código? ¿Hemos cambiado el funcionamiento de ese método?

7. Selecciona el código “*length(pass) &&containsDigits(pass)*” y refactoriza con la opción “introduce->metod”, dando nombre del nuevo método “*passIsMinimum()*”.

Selecciona el código “*containsUpLowerCase(pass) &&hasSpecial(pass)*” y refactoriza con la opción “introduce->metod”, dando nombre del nuevo método “*passIsMaximum()*”.

¿Qué cambios ves en el código?

ANALIZADOR DE CÓDIGO

8. Instala el plugin del analizador de código SonarLint:
<https://plugins.netbeans.apache.org/catalogue/?id=21>

9. Lanza el análisis para el proyecto.
<https://www.youtube.com/watch?v=f0KOFfe-EVJ>

I

Comenta tres de los errores o avisos generados por el analizador.

10. Deshabilita aquellos tres avisos en la configuración de reglas del analizador y lanza de nuevo el analizador.

Criterios de puntuación

Cada apartado se puntuará con 1 punto.

Consejos, recomendaciones e indicaciones de entrega

Una vez realizada la tarea elaborarás un único documento PDF donde figuren una portada, las respuestas y los pantallazos (recortados) necesarios. La redacción de las respuestas, su explicación o justificación es una condición “sine qua non” para que se puntúen. 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_ED08_Tarea

El fichero Javadocs generado tendrá el nombre: **apellido1_apellido2_nombre_**

ED08_JAVADOCS