ELISABET CALVO ESPAÑA

ED 04

Tarea

Índice de contenidos:

* Ejercicios de refactorización

(pág 2 – 7)

* Ejercicios de **GIT**

(pág 8 – 11)

* Ejercicios de **Javadoc**

(pág 12 – 14)

|  |
| --- |
| **Enunciado.** |
| En el proyecto **Java “Deposito”**, hay definida una Clase llamada *Ccuenta*, que tiene una serie de atributos y métodos. El proyecto cuenta asimismo con una Clase *Main*, donde se hace uso de la clase descrita.  Basándonos en ese proyecto, vamos a realizar las siguientes actividades.  **REFACTORIZACIÓN**   1. Las clases deberán formar parte del paquete cuentas.   Para ello, como ya había creado el archivo en el paquete por defecto, desde el menú contextual del paquete, accedemos a refactor, rename y cambiamos el nombre del mismo a “cuentas”     1. Cambiar el nombre de la variable “miCuenta” por “cuenta1”.   Para ello, nos situamos también en el menú contextual pero esta vez sobre el nombre de la variable, damos a **refactor** y **rename**, cambiando el nombre por “cuenta1”.    Una captura de pantalla de una computadora  Descripción generada automáticamente   1. Introducir el método operativa\_cuenta, que englobe las sentencias de la clase Main que operan con el objeto cuenta1.   Para ello agregamos en el método main una sentencia que se llame operativa\_cuenta() y luego creamos un nuevo método con el nombre operativa\_cuenta que englobe todas las sentencias del ejercicio.     1. Encapsular los atributos de la clase Ccuenta.   Para ello seleccionamos los atributos de la clase, vamos a refactor, encapsulate fields, y seleccionamos todos para que nos haga los métodos set y get.         1. Añadir un nuevo parámetro al método operativa\_cuenta, de nombre cantidad y de tipo float.   Para ello nos vamos al menú contextual del método (seleccionándolo con el ratón y botón derecho) y elegimos **refactor**, **change** **method** **parameters** para añadir el nuevo parámetro. |
| Pinchamos en add para añadir el parámetro y definimos el tipo, el nombre y el valor por defecto. |
|  |
|  |
| **GIT**   1. Configurar GIT para el proyecto. Crear un repositorio público en GitHub.   Nos damos de alta en github.com y creamos el repositorio público.     1. Realizar, al menos, una operación commit. Comentando el resultado de la ejecución.   Desde consola, después de instalar GIT con el comando git init desde la carpeta del proyecto, creamos un nuevo repositorio local. Con git status vemos que no hay commits todavía.    Con el comando git add . añadimos los archivos al repositorio. Con git status de nuevo, vemos lo que se ha añadido.      Añadimos el repositorio remoto con el comando git remote add    Ahora hacemos el push para subir el proyecto a la web. Autorizamos las credenciales que nos pide      Vemos el commit que hemos realizado con el comando git log     1. Mostrar el historial de versiones para el proyecto mediante un comando desde consola.   Podemos verlo con el comando git log que hemos mencionado en el anterior ejercicio.    La URL del repositorio github es la siguiente:  https://github.com/eliblacks/ED\_Tarea04.git  **JAVADOC**   1. Insertar comentarios JavaDoc en la clase CCuenta.   Insertamos los comentarios utilizando las marcas que le ayudan al API a reconocer las partes que queremos documentar y crear los archivos HTML.   1. Generar documentación JavaDoc para todo el proyecto y comprueba que abarca todos los métodos y atributos de la clase CCuenta.   Para generar el documento JavaDoc, vamos a Run, Generate JavaDoc.    Vemos en la parte de abajo que se ha generado correctamente, y también la ruta donde se ha guardado. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |