FUNDACION UNIVERSITARIA TECNOLOGICO COMFENALCO

ALUMNO

EZEQUIEL DAVID PALENCIA OCAMPO

PROFESOR

RONALD CARRASCAL

TEMA

CLEAN CODE

Cartagena 10/08/2024

Beneficios en mi Proyecto de Aula según la lectura:

1. Organización de Clases:

- Beneficio: La lectura destaca la importancia de organizar las clases de manera clara y estructurada, comenzando por las constantes, variables y luego las funciones públicas y privadas. Esto me ayudará a mantener un código limpio y fácil de entender, lo cual es importante cuando se trabaja en un proyecto que maneja muchos datos.

- Aplicación: En el proyecto de aula, puedo aplicar esta organización para asegurarme de que mi código sea fácil de mantener y escalar. Esto es especialmente útil si la aplicación crece.

2. Encapsulación:

- Beneficio: Mantener la encapsulación adecuada me permite proteger los datos y funciones de mi aplicación, asegurando que solo las partes necesarias sean accesibles desde fuera de la clase. Esto mejora la seguridad y manejo del código.

- Aplicación: Puedo aplicar encapsulación en la aplicación para proteger datos sensibles de los pacientes y limitar el acceso a funciones críticas, garantizando que solo las partes necesarias del código interactúen con estos datos.

3. Tamaño Reducido de Clases:

- Beneficio: Mantener clases pequeñas y con responsabilidades bien definidas facilita la comprensión y el mantenimiento del código. Esto también promueve la reutilización de código y la reducción de errores.

- Aplicación: En mi proyecto, me aseguro de que cada clase tenga una responsabilidad clara y específica, como gestionar la base de datos, procesar los cálculos, o generar gráficos. Esto me permitirá construir un sistema más robusto y fácil de correr.

Implementación en el Proyecto de Aula:

- Estructura de Clases: Puedo Organizar las clases siguiendo las convenciones descritas en "Clean Code". Comenzaría por definir las constantes y variables, y luego las funciones. Esto hará que el código sea más intuitivo y fácil de seguir.

- Modularidad y Encapsulación: Necesito Mantener las funciones y variables lo más privadas posible. Requiero compartir funcionalidad entre clases, considero que usar el ámbito `protected` o protegido solo cuando sea necesario.

- División de Responsabilidades: Evalúo cada clase para asegurarme de que esté enfocada en una sola responsabilidad. Si alguna clase parece manejar demasiadas tareas, considero dividirla en clases más pequeñas.

Implementando estos principios, el proyecto no solo será más fácil de desarrollar y mantener, sino que también ofrecerá una solución más sólida y confiable para los médicos que lo utilicen.

https://github.com/EzequielPalencia