

PROGRAMACIÓN ORIENTADA A OBJETOS

Nombre y Apellido: Alex Emiliano Agüero Recio

Carrera: LCC

Actividad N°1

A) Agregue nuevas cuestiones inherentes a los conceptos de abstracción y ocultamiento de información que a su criterio no están presentes en este documento.

Abstracción: (Consiste en aislar un elemento de su contexto o del resto de elementos que lo acompañan). Permite utilizar lo desarrollado por otros en nuestras aplicaciones o incluso lo desarrollado en casa para otros sistemas.

Se utiliza también para lo que se denomina implementación de tipos de datos abstractos (TDA), diremos que trabajamos con un TDA cuando manejamos unos tipos de datos que no está predefinido en el lenguaje utilizado, simulándolo de manera totalmente transparente, mediante un adecuado uso de una biblioteca de subprogramas. Esto permite manejar tipos de datos que el lenguaje no posee, ya testeados en otros contextos.

Puesto que los tipos de datos no son conjuntos sino álgebras, conjuntos más las operaciones definidas sobre los elementos de los mismos, se hace necesario especificar ambas cosas. Por ello, la implementación de un TDA implica:

- Determinación de las operaciones necesarias sobre el TDA.
- Elección de una estructura de datos sobre la cual se materializará la implementación.
- Implementación de la estructura de datos y las operaciones.

Ocultamiento: (Los datos de un objeto sólo se pueden cambiar mediante operaciones definidas para ese objeto). Con el principio del ocultamiento de datos permite el desarrollo de software de modo tal que el cliente no pueda usar las estructuras de datos y los algoritmos subyacentes. Es decir, pueden usar las interfaces de los TDA, pero no la implementación. De este modo, la interfaz se convierte en una especie de contrato entre las partes, que debe ser respetado.

Precisa de un par de normas:

- Al cliente, se le debe proveer sólo toda la información que necesita para el uso del módulo del módulo correctamente.
- Al que implemente el módulo, se le debe proveer toda la información necesaria para comprender el uso que se hará del mismo, pero ninguna información más.

Una especificación debe señalar cuáles son las funciones que el módulo realiza, describiendo el dominio de cada una, los valores iniciales, los parámetros que se deben utilizar y el efecto de dichas funciones.

El ocultamiento implica separar el qué del cómo, el cliente debe conocer el qué hace el módulo, pero no necesariamente cómo lo hace. Las ventajas:

- Permite cambios de implementación.
- Impide violaciones de restricciones entre datos internos.

B) Teniendo en cuenta que uno de los objetivos de la orientación a objetos es la calidad del software, defina calidad de software e indique factores para obtener dicha calidad.

La calidad de software es una evaluación que considera el nivel de satisfacción que alcanza el usuario o cliente a partir del momento que comienza a utilizar el software. Un mecanismo menos exigente evalúa la calidad del sistema con relación a los requerimientos acordados.

Los factores son:

- **Correctitud:** Un producto de software es correcto cuando actúa de acuerdo a los requerimientos especificados.
- **Eficiencia:** Un producto de software es eficiente si tiene una baja demanda de recursos de hardware, en particular el tiempo de la CPU, espacio de memoria y ancho de banda.
- **Portabilidad:** Un producto de software es portable si puede ejecutarse sobre diferentes plataformas de hardware y de software.
- **Simplicidad:** Un producto de software es simple si es fácil de usar, su interfaz es amigable, y no requiere demasiado entrenamiento ni capacitación por parte del usuario.
- **Robustez:** Un producto de software es robusto si reacciona adecuadamente en circunstancias imprevisibles.

- **Usabilidad:** Un producto de software es usable si está disponible en el momento que el usuario lo necesita y el rendimiento está dentro de los parámetros establecidos.
- C) **Cuando se habla de productividad del software, se habla de reducir los costos del software y el tiempo de desarrollo. Investigue sobre cualidades relacionadas a la productividad del software (página 7).**

La productividad implica reducir tiempos y costos en la concepción y construcción de un sistema y está fuertemente ligada a dos cualidades fundamentales, extensibilidad y reusabilidad.

Extensibilidad: se refiere a la flexibilidad de un sistema de software para adaptarse a cambios en los requerimientos. Un producto de software es extensible si es fácil adaptarlo a cambios en la especificación.

Reusabilidad: es la capacidad de aprovechar en la construcción de un nuevo sistema, componentes desarrolladas previamente. Un módulo de software o una colección de módulos es reusable si puede utilizarse para la construcción de diferentes aplicaciones.

La extensibilidad afecta fundamentalmente al mantenimiento, pero depende de las etapas anteriores. Un sistema con un diseño adecuado será más fácil de extender que una solución con un diseño pobre. La reusabilidad afecta a todas las etapas. Cuando el equipo de desarrollo resuelve un problema puede reusar requerimientos, módulos de diseño o código de sistemas anteriores. Ambas cualidades están ligadas a la legibilidad