

- **OCULTACIÓN DE INFORMACIÓN**

- El uso de la abstracción provoca ocultación de información.
- Mediante la abstracción operacional, el usuario de una función solo necesita conocer su especificación (qué hace), pero no los detalles internos de su implementación (cómo lo hace).
- En la abstracción de datos, la ocultación de información permite separar la interfaz de un TAD (características y comportamiento visibles desde el exterior) de los detalles internos de la implementación (invisibles desde el exterior).
- La interfaz de un TAD es un “contrato” de servicios con el usuario.
- Disminuye las interdependencias entre los componentes de un sistema, es decir, reduce el acoplamiento entre ellos.

- **INDEPENDENCIA DE LA REPRESENTACIÓN**

- La abstracción de datos se desarrolla en dos niveles:
 - Nivel conceptual o de especificación:
Definición de
 - Dominio (colección de datos)
 - Interfaz (operaciones y sus propiedades)
 - Nivel de representación o implementación:
Definición de
 - Estructura de datos
 - Implementación de las operaciones (exhibiendo las propiedades y comportamientos especificados)
- La especificación de un TAD es independiente de cualquier representación del mismo que se pueda diseñar.
- El uso de un TAD se basa en su especificación y, por tanto, también es independiente de la representación subyacente.

- La independencia de la representación implica:
 - Un mismo TAD se puede implementar con distintas representaciones.
 - Los requisitos del problema o aplicación determinarán la representación más adecuada a utilizar.

- **RELACIÓN ENTRE OCULTACIÓN DE INFORMACIÓN E INDEPENDENCIA DE LA REPRESENTACIÓN**

En programación, el concepto de abstracción de un problema se consigue mediante la abstracción operacional y la abstracción de datos. Ambas, implican ocultación de información, ya que al usuario no le interesa saber cómo se están implementadas las operaciones ni cómo se almacenan los datos. La abstracción de datos se consigue mediante la utilización de tipos abstractos de datos (TADs), los cuales, siguen el principio de independencia de la representación, que asegura al usuario que las operaciones y el dominio del TAD descritos en la especificación, van a comportarse tal y como se especifica independientemente de la representación que se use.

- **ESPECIFICACIÓN DE UN TAD**

Comprende toda la información que necesitar conocer el usuario del TAD.

- Definición del dominio o conjunto de datos del TAD.
- Especificación de operaciones:
 - Especificación sintáctica: Indica cómo usar las operaciones.
 - Especificación semántica: Indica qué hacen y qué propiedades tienen cada una de las operaciones.

Para implementarla usamos precondiciones y postcondiciones. Bajo el cumplimiento de las precondiciones se puede realizar la operación y, al finalizar la misma, se garantiza que se cumplirán las postcondiciones.

- **TIPO ABSTRACTO DE DATOS VS ESTRUCTURA DE DATOS**

El adjetivo “abstracto” expresa que un TAD es un modelo conceptual descrito por su especificación, que sólo existe en la mente del programador; mientras que una estructura de datos es la representación de esa abstracción en un lenguaje de programación.

- **PREGUNTAS**

- Si una operación está bien implementada, deben cumplirse obligatoriamente las postcondiciones al finalizar la misma.

FALSO

- Si no se cumplen las postcondiciones de una operación, implica obligatoriamente que dicha operación está mal implementada.

FALSO

- Es siempre obligación del diseñador del TAD, al implementar las operaciones del mismo, comprobar que se cumplen las precondiciones, y en caso de no ser así, enviar un mensaje de error indicándolo y abortar la ejecución de las mismas.

FALSO

- La especificación del TAD es útil solamente para el usuario, para saber el tipo y las operaciones del mismo para implementar sus programas, haciendo uso de él.

FALSO

- Un usuario de un TAD puede acceder a estructuras del mismo, siempre y cuando respete el principio de independencia de la representación.

FALSO

- Una vez ejecutada una operación de un TAD, si no se cumple la precondition hay que enviar un mensaje de error al usuario indicando lo ocurrido.

FALSO

- Bajo ninguna condición es posible hacer la especificación de un TAD con posterioridad a su implementación.

VERDADERO

- La independencia de la representación implica que podemos acceder directamente a la estructura del TAD siempre que se haga mediante funciones externas convenientemente especificadas.

FALSO

- La especificación del TAD debe realizarse siempre después de la implementación, para asegurarse de que no se comenten errores.

FALSO