

- **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

PILA

- **DEFINICIÓN**

Una pila es una secuencia de elementos en la que todas las operaciones se realizan por un extremo de la misma. Dicho extremo recibe el nombre de tope o cima.

- **ESPECIFICACIÓN**

- Pila()
Post: Crea una pila vacía
- Bool vacia()const
Post: Devuelve true si la pila está vacía
- Const tElemento& tope() const
Pre: La pila no está vacía
Post: Devuelve el elemento del tope de la pila
- Void pop()
Pre: La pila no está vacía
Post: Elimina el elemento del tope de la pila y el siguiente se convierte en el nuevo tope
- Void push(const tElemento& x)
Post: Inserta el element x en el tope de la pila y el antiguo tope pasa a ser el siguiente.

- **IMPLEMENTACIÓN VECTORIAL ESTÁTICA**

Exige fijar el tamaño máximo de la pila en tiempo de compilación, por lo que el tamaño tendrá que ser una constante y el mismo para todas las pilas.

- **IMPLEMENTACIÓN VECTORIAL PSEUDOESTÁTICA**

Cuando queremos tener pilas de distintos tamaños máximos en un programa. Es importante señalar que este tamaño será constante durante el tiempo de vida de cada pila.

Almacena dos campos del vector y la posición del tope, requiere un campo para almacenar el tamaño máximo(LMAX)

- **IMPLEMENTACIÓN VECTORIAL MEDIANTE ESTRUCTURAS ENLAZADAS(DINÁMICA)**

El tamaño es variable y siempre ocupará el espacio justo para almacenar los elementos que contenga la pila en cada momento.

Consistirá en una secuencia de registros enlazados con punteros, en la que cada registro almacenará un elemento y la pila será un puntero al primer elemento de la secuencia(tope), por donde se realizan las inserciones y borrados. Se hace por referencia. El espacio de memoria ocupado es menor a la estática o pseudoestática.

- **PREGUNTAS**

- En el TAD Pila, puedo guardar un elemento menos debido a que en el nodo cabecera no se guarda nada.

FALSO

- El TAD Pila no incorpora una operación concreta para acceder a un elemento cualquiera de la misma, pero sí podemos acceder a cualquier elemento de la Pila, siempre y cuando la implementación del TAD sea utilizando una representación vectorial.

FALSO

- En la estructura vectorial del TAD Pila, no es posible aprovechar todos los elementos del vector ya que no nos permite distinguir entre pila llena y pila vacía.

FALSO

- No tiene sentido implementar una pila mediante un estructura doblemente enlazada porque, aunque el doble puntero es útil, resulta muy costoso en términos de espacio.

FALSO

- Si he de escoger entre una estructura simplemente enlaza y doblemente enlazada a la hora de implementar una pila, la decisión dependerá de si alguna operación en concreto se va a realizar con mucha frecuencia o no.

FALSO

- El TAD Pila es útil para resolver, entre otros, problemas en los que es necesario procesar los elementos en orden inverso al que se proporcionan.

VERDADERO

- Aunque desconozcamos el tamaño máximo del problema, es posible, aunque ineficiente, usar una estructura vectorial pseudoestática.

FALSO

- Si conozco a priori el tamaño máximo del problema, desde el punto de vista de la eficiencia espacial, no debo utilizar nunca una estructura enlazada, siempre mejor una vectorial.

VERDADERO

- La representación vectorial circular del TAD Pila deja un hueco libre para distinguir cuando está llena o vacía.

FALSO

- En la representación dinámica del TAD Pila, no es necesario añadir por un nodo cabecera, pero si lo ponemos, nos facilita el acceso al elemento que está en el tope.

FALSO

- El TAD Pila no incorpora una operación concreta para acceder a un elemento cualquiera de la misma, pero sí podemos hacer el cambio elemento de la Pila, implementando la operación a partir de las operaciones del TAD.

VERDADERO