

Tipos de datos abstractos

Un TDA es un tipo de dato definido por el programador que se puede manipular de un modo similar a los tipos de datos definidos por el sistema.

Está formado por un conjunto válido de elementos y un número de operaciones primitivas que se pueden realizar sobre ellos.

Ejemplo: - Definición del tipo Numero racional: Conjunto de pares de elementos (a,b) de tipo entero, con $b \neq 0$.

A la hora de utilizar el TDA, la representación debe permanecer oculta. Solo podremos utilizar las operaciones del tipo para trabajar con sus elementos.

- Para construir un tipo abstracto debemos:

1. Exponer una definición del tipo.
2. Definir las operaciones (funciones y procedimientos) que permitan operar con instancias de ese tipo.
3. Ocultar la representación de los elementos del tipo de modo que sólo se pueda actuar sobre ellos con las operaciones proporcionadas.
4. Poder hacer instancias múltiples del tipo.

♦ Tipos básicos de operaciones en un TDA

- Constructores: Crean una nueva instancia del tipo.
- Transformación: Cambian el valor de uno o más elementos de una instancia del tipo.
- Observación: Nos permiten observar el valor de uno o varios elementos de una instancia sin modificarlos.
- Iteradores: Nos permiten procesar todos los componentes en un TDA de forma secuencial.

Abstracción o encapsulamiento:

Separación de la especificación de un objeto o algoritmo de su implementación, en base a que su utilización dentro de un programa sólo debe depender de un interfaz explícitamente definido (la especificación) y no de los detalles de su representación física (la implementación), los cuales están ocultos.

- Ventajas de la abstracción:

Establece la independencia de QUÉ es el objeto (o QUÉ hace el algoritmo) y de CÓMO está implementado el objeto (o algoritmo), permitiendo la modificación del CÓMO sin afectar al QUÉ, y por lo tanto, sin afectar a los programas que utilizan este objeto o algoritmo.

- Tipos de abstracciones:

Abstracción de operaciones. Una serie de operaciones básicas se encapsulan para realizar una operación más compleja. En los lenguajes de programación este tipo de abstracción se logra mediante los subprogramas:

TipoLista

Colección de elementos homogéneos (del mismo tipo: TipoElemento) con una relación LINEAL establecida entre ellos. Pueden estar ordenadas o no con respecto a algún valor de los elementos y se puede acceder a cualquier elemento de la lista.

TipoCola

Colección de elementos homogéneos (del mismo tipo: TipoElemento) ordenados cronológicamente (por orden de inserción) y en el que sólo se pueden añadir elementos por un extremo (final) y sacarlos sólo por el otro (frente). Es una estructura FIFO (First In First Out).

TipoPila

Colección de elementos homogéneos (del mismo tipo: TipoElemento) ordenados cronológicamente (por orden de inserción) y en el que sólo se pueden añadir y extraer elementos por el mismo extremo, la cabeza. Es una estructura LIFO (Last In First Out).

TipoABin

Estructura de elementos homogéneos (del mismo tipo: TipoElemento) con una relación JERARQUICA establecida entre ellos a modo de árbol binario.

TipoABB

Estructura de elementos homogéneos (del mismo tipo: TipoElemento) con una relación JERARQUICA establecida entre ellos a modo de árbol binario en el que el subárbol izquierdo de cualquier nodo, si no está vacío, contiene valores menores, con respecto a algún campo clave de tipo TipoClave, que el valor que contiene dicho nodo, y el subárbol derecho, si no está vacío, contiene valores mayores.:

Referencias:

Joyanes Aguilar, L., & Zahonero Martínez, I. (1998a). Estructura de datos. Algoritmos, abstracción y objetos. (Ed. rev.). Madrid, España: McGraw-Hill.
Cairó, O., & Guardati, S. (2002). Estructuras de datos (3rd ed.). México: McGraw-Hill.