

Tipos de datos abstractos

La idea fundamental de los tipos de datos abstractos (TDA) es separar de forma escrita la representación del uso de los datos de un tipo. Para empezar, hemos de tener en cuenta que un tipo no solamente consta de un conjunto de valores, sino también de un conjunto de operaciones para la creación, modificación y manipulación de dichos valores, a través de las cuales se realizará el uso de los datos.

Por una parte, tenemos la representación del tipo de datos en términos de tipos básicos o de otros tipos ya conocidos, así como la implementación sobre dicha representación de las operaciones asociadas al tipo de datos, que constituyen la interfaz de ese tipo. En los algoritmos que implementan esas operaciones se tiene pleno acceso a la representación concreta del tipo de datos. Así, en el ejemplo de las fechas se puede ver si el año se representa con dos o con cuatro dígitos, o de otra forma.

Por otra parte, la única forma de usar los datos del tipo abstracto es invocado adecuadamente las operaciones de la interfaz, de manera que nunca se accede a su representación interna, de ahí la abstracción. Así podemos tener diferentes representaciones para un mismo tipo de datos, pero cuando se usa el tipo nunca se sabe cuál es la representación que se está utilizando. Cuando se cambia la representación, siempre y cuando la interfaz se mantenga, los programas que usan el tipo deben continuar funcionando exactamente de la misma forma que antes.

En particular, una vez se ha fijado la interfaz del tipo, la programación de la representación del tipo e implementación de las operaciones por un lado, y la programación de los algoritmos que usan el tipo por otro, se pueden proceder de manera completamente independiente la una de la otra. Sin embargo, para que esto sea posible que los programadores de los segundos conozcan la representación concreta que manejan los programadores de las primeras, hace falta que intercambien no solamente la interfaz sino también una descripción apropiada del comportamiento de las operaciones que constituyen dicha interfaz.

Ejemplos de tipos de datos abstractos:

- **Listas:**

Colección de elementos homogéneos 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.

- **Colas:**

Colección de elementos homogéneos ordenados cronológicamente y en el que sólo se pueden Añadir elementos por un extremo y Sacarlos sólo por el otro. Es una estructura FIFO (First In First Out).

- **Pilas:**

Colección de elementos homogéneos 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)

- **Arboles:**

Un árbol binario es un conjunto finito de elementos que está vacío o partido en tres subconjuntos disjuntos. El primer subconjunto contiene un único elemento llamado Raíz del árbol binario. Los otros dos subconjuntos son a su vez árboles binarios, llamados subárboles izquierdo y derecho del árbol binario original. Nótese que un subárbol izquierdo o derecho puede estar vacío. Cada elemento de un árbol binario se denomina Nodo. Si un subárbol o la raíz contiene a su vez dos subárboles vacíos, es un nodo hoja.

En otras palabras, un tipo de dato abstracto o tipo abstracto de dato es un conjunto de operaciones con una función específica que puede ser aplicada en un programa independientemente de su uso su función siempre será la misma utilizando ciertas variables ya definidas y otras que pueden variar dependiendo de la necesidad para al que se utilice en el programa todo esto para un mejor manejo y orden presentación y manejo de los datos mediante la encapsulación y el ocultamiento de la información para así agrupar y proteger los datos que se maneje para así tener un mejor control.

Bibliografías:

Oliet, N. M., Mallén, Y. O., & López, J. A. V. (2004). *Estructuras de datos y métodos algorítmicos: ejercicios resueltos*. Pearson educación.

5 Tipos Abstractos de Datos- LCC www.lcc.uma.es/~jlleivao/algoritmos/t5.pdf