

Tipos de datos abstratos

Tipo de Dato Abstracto (TDA). Es un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de **datos** para el modelo.

Caracterización

Un TDA se caracteriza por:

1. Exportar un tipo (una **clase** lo es)
2. Definir un conjunto de operaciones para la manipulación (la interface, el conjunto de **métodos** tanto públicos como protegidos que pueden ser utilizados).
3. Utilizar la interface como único mecanismo de acceso a la **estructura de datos** (encapsular, esconder el cómo).

Ejemplos

- TDA: entero es.
- Datos: una secuencia de dígitos que ocasionalmente presentan como **prefijo** un **signo** más(+)o menos(-). A esta secuencia le damos el nombre de n.

Operaciones

- Suma (entero K): crea un nuevo entero resultado de la **suma** de n y k.
- Resta (entero K): crea un nuevo entero resultado de la resta de n y k.
- Multiplicar(entero K): crea un nuevo entero resultado de la **multiplicación** n y k.
- Dividir(entero K): crea un nuevo entero resultado de la **división** n y k.
- Resto(entero K): crea un nuevo entero, el resto la división de n y k.
- Obtener: devuelve a n.
- Poner(entero K): pone en n el valor de k.

Lista TDA

Una lista se define como una n-tupla de elementos (donde 'Ln es el n-ésimo elemento de la lista L) ordenados de forma consecutiva, o sea, el elemento Ln le precede al elemento Ln+1.

- $L = (L1, L2, \dots, Ln)$.

El valor de n representa la posición del elemento en la lista. Si la lista contiene 0 elementos se denomina lista vacía.

Operaciones

- Constructor: crea una nueva lista, la crea vacía.
- Insertar(Tipo x, Entero n): insertar un nuevo elemento x, en la posición n de la lista.
- Adicionar (Tipo x): adiciona un elemento al final de la lista.
- Tipo Obtener (Entero n): devuelve el elemento que está en la posición n de la lista.
- Eliminar (Entero n): elimina el elemento que está en la posición n de la lista. Los elementos posteriores a partir de la posición n+1 pasan a tener la posición anterior inmediata.
- Entero Longitud: devuelve la cantidad de elementos de la lista. Si la lista es vacía devuelve el valor 0.
- Lógico Vacía: devuelve un valor lógico verdadero si la lista tiene longitud cero, falso en caso contrario.

Notas

- Algunos autores de libros refieren que las listas deben estar enumeradas siguiendo el modelo de numerado entero, donde se comienza por 0, muy usado en muchos sistemas y lenguajes de programación como C/C++. Otros refieren que se use el sistema tradicional que empieza por 1.
- En la mayoría de las implementaciones se utiliza el manejo de memoria dinámica para el almacenaje de las n posiciones y que en las mismas se haga referencia a las direcciones donde se encuentra la información guardada en la lista.

Bibliografía:

Fundamentals of Data Structures in C++. E. Horowitz, S. Sanhi, D. Mchta. Computer Science Press, 1995.

Data structures and algorithms. Aho .et al.