

Renteria Magaña Rayni Damian. 17212178

Tarea 2. Tipo de Datos Abstractos.

Tipo de dato abstracto de datos (TAD) 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. Al tipos definidos por el sistema, un tipo de dato abstracto corresponde a un puede ser de tamaño indefinido) de valores legales de datos y un número de operaciones primitivas que se pueden realizar sobre esos valores. Los usuarios pueden variables con valores que están en el rango de valores legales y pueden operar sobre esos valores utilizando las operaciones definidas. Por ejemplo, en el caso de la pila ya citada, se puede definir dicha pila como un tipo abstracto de datos y las operaciones realizadas sobre instancias de la pila.

nes sobre la pila como las únicas operaciones legales que están permitidas para ser realizadas. Los módulos se utilizan frecuentemente como una técnica de implementación para tipos abstractos de datos, y el tipo abstracto de datos es un concepto más teórico. Para construir un tipo abstracto de datos se debe poder:

1. Exponer una definición del tipo.
4. Hacer disponible un conjunto de operaciones que se pueden utilizar para manipular instancias de ese tipo.
- otegar los datos asociados con el tipo de modo que sólo se puede actuar sobre ellas con las rutinas proporcionadas.
4. Hacer instancia múltiples del tipo. do información y no cumplen básica

Ventajas de Los TAD

1. Permite una mejor conceptualización y modelado del mundo real. Mejora la representación y la comprensibilidad. Clarifica los objetos basados en estructuras y comportamientos comunes.
- Mejora la robustez del sistema. Si hay características subyacentes en los lenguajes permiten la especificación del tipo de cada variable, los tipos abstractos de datos permiten la comprobación de tipos para evitar errores de tipo en tiempo de ejecución.
3. Mejora el rendimiento (prestaciones). Para sistemas tipados, el conocimiento de los objetos permite la optimización de tiempo de compilación.
4. Separa la implementación de la especificación. Permite la modificación y mejora de la implementación sin afectar al interfaz público del tipo abstracto de dato.
3. Permite la extensibilidad del sistema. Los componentes de software reutilizables son más fáciles de crear y mantener.
6. Recoge mejor la semántica del tipo. Los tipos abstractos de datos agrupan o localizan las operaciones y la representación de atributos.

- Punteros
- Listas enlazadas
- Listas doblemente enlazadas
- Pilas
- Colas y Colas de Prioridades
- Árboles
- Arboles binarios

Referencias

Aguilar, L. J., & Martínez, I. Z. (2002). *Estructura de datos: Algoritmos, abstracción y objetos*.

McGraw-Hill/interamericana.