

04/08/20
TAREA 1
Luis Diego Lazcano

1.1 Tipos de datos abstractos (TDA)

Los tipos de datos abstractos (TDA) encapsulan datos y funciones que trabajan con estos datos. Los datos no son visibles para el usuario en un tipo de dato abstracto y el acceso a los datos es exclusivamente bajo el llamado a funciones, también llamadas métodos. Así, el tipo de dato abstracto son representados por clases, las cuales representan pequeñas deficiencias: el dato que representa el estado de un objeto de este tipo de dato abstracto es visible (Algunas veces no accesible) en la parte private de la clase declarada para cada programa, la clase es reconocida mediante la `using include`. Ejemplos de tipos de datos abstractos son: `Stack`, `Queue`, etc. Los TDA por lo general manejan memoria dinámica, esto es, la asignación dinámica de memoria es una característica que le permite al usuario crear tipos de datos y estructuras de cualquier tamaño de acuerdo a las necesidades que se tengan en el programa.

1.2 Modularidad

Modularidad en Ciencias de la Computación es la característica por la cual un programa de computador está compuesto de porciones que se conocen como módulos. El diseño estructurado es la técnica de diseño de algoritmos en que se basa la programación modular paradigma de programación que persigue desarrollar programas modulares.

La modularidad se basa en la descomposición de un problema en una serie de sub problemas; dividiéndolo en módulos que resultan de segmentar el problema en funciones lógicas que son perfectamente diferenciados. Esta división exige la presencia de un módulo denominada módulo de base o principal a objeto de que controle y se relacione con los demás.

Es una técnica de programación que todavía se utiliza tanto para la construcción de algoritmos computacionales básicos así como apoya el desarrollo de sistemas de gestión (en el diseño de diagramas modulares).