

## **1. O que é um Tipo Abstrato de Dados (TAD) e qual a característica fundamental na sua utilização?**

Abstraída qualquer linguagem de programação, um tipo abstrato de dados (TAD) pode ser visto como um modelo matemático que encapsula um modelo de dados e um conjunto de procedimentos que atuam com exclusividade sobre os dados encapsulados. Em nível de abstração mais baixo, associado à implementação, esses procedimentos são implementados por subprogramas denominados operações, métodos ou serviços. Qualquer processamento a ser realizada sobre os dados encapsulados em um TAD só poderá ser executada por intermédio dos procedimentos definidos no modelo matemático do TAD, sendo esta restrição a característica operacional mais útil dessa estrutura. Nesses casos, um programa baseado em TAD deverá conter algoritmos e estruturas de dados que implementem, em termos da linguagem de programação adotada, os procedimentos e os modelos de dados dos TADs utilizados pelo programa. Assim, a implementação de cada TAD pode ocupar porções bem definidas do programa: uma para a definição das estruturas de dados e outra para a definição do conjunto de algoritmos. Nessas condições, quaisquer alterações realizadas na estrutura de um dado TAD não afetarão as partes do programa que utilizam esse TAD

## **2. Quais as vantagens de se programar com TAD's?**

**Abstração da Informação:** permite uma melhor compreensão dos algoritmos e maior facilidade de programação, e conseqüentemente aumenta a complexidade dos programas, tornando fundamental em qualquer projeto de software a modelagem prévia de seus dados.

**Maior independência e portabilidade de código e Manutenção:** alterações na implementação de um TAD não implicam em alterações em seu uso. Supondo que implementamos um TAD qualquer, e também temos uma aplicação usando este TAD. Se decidirmos mudar a implementação, desde que a aplicação esteja utilizando apenas os operadores, a aplicação continuará funcionando. O que mudou foi a implementação da TAD, mas sua funcionalidade (seus operadores) continuam os mesmos.

**Maior potencial de reutilização de código:** pode-se alterar a lógica de um programa sem a necessidade de reconstruir as estruturas de armazenamento. Uma vez definido, implementado e testado, o TAD pode ser acessado por diferentes programas