



TIC – Hub

Programa Residência em TIC 12

C A P A C I T A Ç ã O

CURSO

Estruturas de dados

Capítulo 2

Tipos abstratos de dados

Prof. Allberson Dantas

EXECUTORES



PARCEIROS



COORDENAÇÃO PPI



INICIATIVA



Iniciativa do Ministério da Ciência, Tecnologia e Inovações e Softex no âmbito do Programa MCTI FUTURO. É um Projeto apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, conforme disposto no Art. 7º da Portaria MCTI N° 5.275, de 5 de novembro de 2021. Este projeto é apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado no Projeto de Residência em TIC 12. Leia o termo de consentimento para tratamento de dados pessoais.

Introdução

Caro(a) aluno(a)!

Nesta aula, você conhecerá um tema importantíssimo para quem está começando na programação e que o acompanhará em sua carreira: os **tipos abstratos de dados** ou TADs.

Conceito Básico

Os TADs são uma maneira de organizar e estruturar dados de maneira eficiente e intuitiva, facilitando o desenvolvimento e a manutenção de programas de computador. São como modelos que definem a estrutura, as propriedades e as operações que podem ser realizadas sobre os dados.

Para entender melhor, imagine um armário com diferentes compartimentos, cada um destinado a armazenar um tipo específico de item, como roupas, livros ou utensílios de cozinha.

Da mesma forma, os TADs definem "compartimentos virtuais" para armazenar e manipular dados de diferentes tipos, como números, textos ou imagens.

Um TAD é definido por duas principais características:

- **Interface:** especifica quais operações podem ser realizadas sobre os dados, independentemente de como esses dados são armazenados ou implementados internamente. Isso inclui operações básicas como inserir, remover, acessar e modificar dados.
- **Implementação:** refere-se à forma como os dados e as operações são efetivamente organizados e executados no nível de código. A implementação pode variar, mas a interface deve permanecer consistente para o usuário do TAD.

Operações Básicas em TADs

No contexto dos TADs, as operações básicas manipulam e gerenciam os dados armazenados. Cada estrutura de dados possui um conjunto específico de operações que definem como os elementos podem ser inseridos, removidos, localizados, acessados e atualizados.

Compreender essas operações com detalhes e aplicá-las em cenários práticos é crucial para que você utilize os TADs eficientemente.

1. Inserção

A operação de inserção adiciona novos elementos à estrutura de dados. O modo como essa operação é realizada varia de acordo com o tipo de TAD.

Por exemplo, imagine uma lista de compras em que você precisa adicionar novos itens. A inserção pode ser feita no final da lista ou em uma posição específica, como logo após o leite, se você deseja que os itens fiquem agrupados por categoria.



Adaptado de: <https://www.tuacasa.com.br/lista-de-compras/>

Ou, ainda, imagine uma pilha de pratos em um restaurante. A inserção de um novo prato se dá no topo da pilha, empilhando-o sobre os pratos anteriores. Isso garante que o primeiro prato colocado na pilha seja o primeiro a ser retirado, seguindo a ordem LIFO presente na estrutura de dados Pilha.



Fonte: <https://pt.dreamstime.com/foto-de-stock-pilha-de-pratos-image77402491>

2. Remoção

A operação de remoção elimina elementos da estrutura de dados. Similarmente à inserção, o modo de remoção varia de acordo com o tipo de TAD.

Pense no seguinte: uma lista de compras em que você precisa remover um item. A remoção pode ser feita por meio da posição do item na lista ou pelo nome do item em si.

Um exemplo mais clássico seria a fila de banco, na qual a “remoção” de um cliente se dá no início da fila, liberando-o para ser atendido e permitindo que o próximo cliente da fila seja atendido. Isso garante que o primeiro cliente a entrar na fila seja o primeiro a ser removido, seguindo a ordem FIFO presente na estrutura de dados Fila.



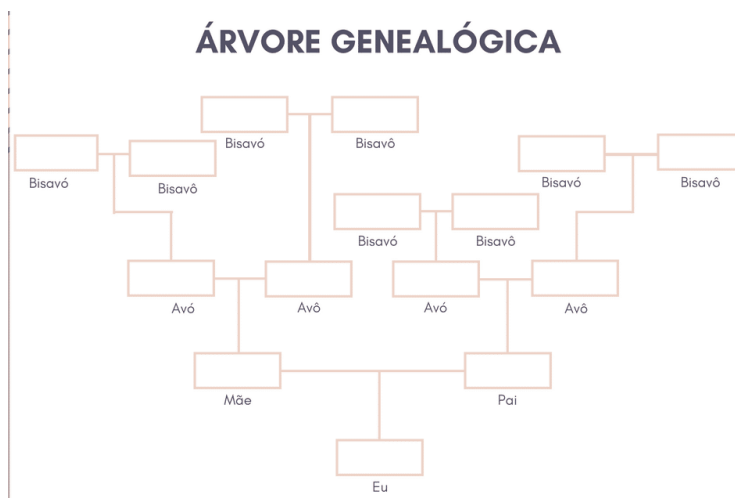
Fonte: <https://www.delepesoasuspesos.com/credito-deudas/165-como-evitar-las-filas-en-los-bancos>

3. Busca

A operação de busca localiza um elemento dentro da estrutura de dados. A eficiência da busca depende da escolha da estrutura de dados e do algoritmo de busca utilizado.

Voltemos ao exemplo da pilha de pratos em um restaurante. A busca por um prato específico na pilha não é eficiente, pois a única forma de acessar um prato é removendo os pratos que estão em cima dele.

Você também pode pensar em uma árvore genealógica ordenada alfabeticamente por sobrenome. A busca binária por um membro da família específico divide a árvore pela metade a cada etapa, seguindo os sobrenomes, até encontrar o membro desejado.



Fonte: <https://tirarcidadaniaitaliana.org/blog/arvore-genealogica-como-montar/>

4. Acesso

A operação de acesso permite obter o valor de um elemento específico na estrutura de dados. O modo de acesso varia de acordo com o tipo de TAD.

Por exemplo, em uma lista de compras, você vai verificar o preço de um item.

O acesso é feito por meio da posição do item na lista, recuperando o valor associado ao item (seu preço).

5. Atualização

A operação de atualização modifica o valor de um elemento existente na estrutura de dados. O modo de atualização varia de acordo com o tipo de TAD, envolvendo a simples alteração do valor ou a substituição completa do elemento.

Imagine mais uma vez uma árvore genealógica. Agora, você precisa atualizar a data de nascimento de um membro da família. A atualização envolve localizar o membro na árvore, remover o nó existente (se a data de nascimento altera a posição na árvore) e inserir um novo nó com a data de nascimento atualizada, mantendo a ordem hierárquica da árvore.

Benefícios dos TADs

- **Modularidade:** permitem dividir o código em módulos independentes e reutilizáveis, facilitando a organização e a manutenção do software;
- **Manutenabilidade:** simplificam a atualização e modificação do código, pois as alterações podem ser feitas em um módulo sem afetar o restante do sistema;
- **Encapsulamento:** protegem os dados internos da estrutura, evitando acessos indevidos e garantindo a integridade das informações;
- **Abstração:** permitem que os programadores se concentrem apenas na funcionalidade fornecida pela estrutura de dados, sem se preocupar com os detalhes de como ela é implementada.

Algumas aplicações dos TADs

Os TADs possuem uma ampla gama de aplicações na resolução de problemas computacionais:

- **Sistemas de gerenciamento de banco de dados:** armazenam e organizam grandes volumes de dados de forma eficiente;
- **Compiladores:** traduzem código de uma linguagem de programação para outra, utilizando TADs para representar a estrutura do programa;
- **Sistemas operacionais:** gerenciam recursos do sistema, como memória e processos, utilizando TADs para organizar e controlar os dados.

Indo um pouco a mais no nosso dia a dia, podemos ver esses conceitos abstratos na vida real. Já pensou em como a função "desfazer" em um editor de texto funciona? Isso mesmo, é como uma pilha! Cada ação que você realiza é empilhada, e você pode desfazer uma por vez, como se estivesse retirando pratos de uma pilha.

E aí, ficou mais claro o que são os **Tipos Abstratos de Dados**? Eles são como ferramentas essenciais para profissionais da computação (principalmente para devs), ajudando-os a organizar e manipular dados de forma eficiente nos nossos programas.

Espero contar com toda sua dedicação para se aprofundar em TADs!

Vamos lá? :)