Estrutura de Dados I

Luciana Lee

Tópicos da Aula

1. Tipo Abstrato de Dado
   - Conceito
   - Motivação
   - Implementação
   - TAD: Pilha

TAD: Tipo Abstrato de Dado

Abstração

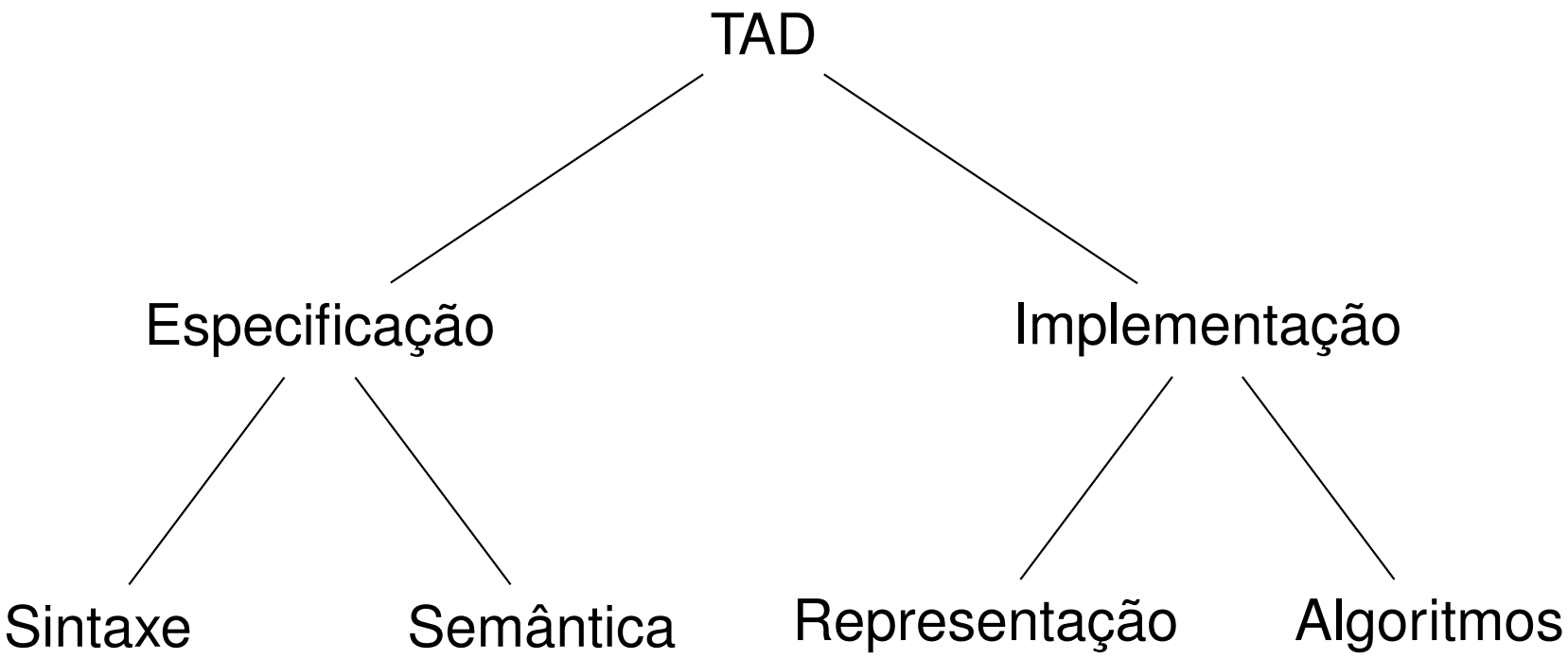
É a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais.

- Quando definimos um TAD (Tipo Abstrato deDados), nos concentramos nos aspectos essenciais do tipo de dado (operações) e nos abstraímos de como ele foi implementado.

Tipo Abstrato de Dado (TAD)

É uma especificação de um conjunto de dados e operações que podem ser executadas sobre esses dados.

TAD: Tipo Abstrato de Dado



- Separação de especificação e implementação: permite o uso do TAD sem conhecer nada sobre a sua implementação.

TAD: Tipo Abstrato de Dado

- A sintaxe de um TAD é conhecida como um conjunto de assinaturas de operações que especifica a sua interface.
- A especificação sintática de um TAD pode não ser suficiente para descrever o comportamento.
  - Especificar a sua semântica de maneira independente da implementação.

Motivação

- Encapsulam a representação dos dados e as operações que podem ser realizadas sobre eles
- Usuário do TAD $\times$ programador do TAD
  - Usuário só “enxerga” a interface, não a implementação
  - Não importa se a representação é com vetor, lista simplesmente encadeada, lista duplamente encadeada, árvore binária de busca, etc.
- Os usuários de um TAD só têm acesso às operações disponibilizadas sobre os dados

Motivação

- **Reúso:** TAD pode ser reaproveitado em vários programas ou módulos
- **Manutenção:**
  - Podemos modificar a implementação do TAD sem modificar o código que usa o TAD
  - E vice-versa, podemos modificar o código que usa o TAD sem modificar a implementação do TAD
- **Corretude:** o código pode ser testado em diferentes contextos.

Implementação

- Em linguagens orientadas a objeto (C++, JAVA) a implementação é feita usando classes e a especificação usando interfaces.
- Em linguagens estruturadas (C, pascal) a implementação é feita pela definição de tipos juntamente com a implementação de funções.

Práticas de Programação de TAD

- Para implementar um Tipo Abstrato de Dados em C, usa-se a definição de tipos juntamente com a implementação de funções que agem sobre aquele tipo.
- Como boa prática de programação, evita-se acessar o dado diretamente, fazendo o acesso somente através de das funções.

Práticas de Programação de TAD

- Uma boa técnica de programação é implementar TADs em arquivos separados do programa principal.
- Para isso geralmente separa-se a declaração e aimplementação do TAD em dois arquivos:
  - Arquivos cabeçalho (extensão .h): com as declarações das estruturas, variáveis globais e assinaturas das funções.
  - Arquivos com implementações (extensão .c): com as implementações das funções declaradas nos arquivos .h.

Compilação de um código no Linux

- O compilador lê o programa inteiro e o converte para código objeto.
- Código objeto é um código binário, também conhecido como código de máquina. Fica armazenado em um arquivo e pode ser executado pelo processador.
- O código compilado, após ser ligado, pode ser executado diretamente, sem necessitar de nova compilação e sem depender do compilador.

Tarefas do compilador

- Pré-processamento
- Compilação
- Ligação/edição

Pré-processamento

O pré-processamento é feito antes da compilação e consiste em:

- Retirada de comentários
- Inclusão de arquivos de cabeçalhos (headers)
- Inclusão de macros no meio do código.

Compilação

- As instruções são convertidas para código de máquina (código objeto – extensão .o)
- É feita a verificação da sintaxe do código, erros de sintaxe são detectados nessa fase.
- Durante a compilação de um programa, ocorrem os eventos de tempo de compilação. Por exemplo: avisar que uma variável não foi declarada.

Ligação/Edição

- Feita pelo **ligador** ou ainda *linker* ou *loader*.
- O ligador é responsável por ligar (linkar) as várias partes do código objeto:
  - códigos objeto gerados pelo programador
  - bibliotecas padrão da linguagem
  - rotinas (bibliotecas) do sistema
- Após a ligação/edição é gerado o código executável.
- Durante a ligação/edição são atribuídos os endereços das instruções de “jump” (desvio) e “call” (chamadas), ou seja, resolvem-se as referências externas.

Exemplo de TAD: Pilha

- É uma lista linear em que todas as inserções, retiradas e, geralmente, todos os acessos são feitos em apenas um extremo da lista.
- Os itens são colocados um sobre o outro. O item inserido mais recentemente está no topo e o inserido menos recentemente no fundo.
- O modelo intuitivo é o de um monte de pratos em uma prateleira, sendo conveniente retirar ou adicionar pratos na parte superior.

Propriedades e Aplicações de Pilhas

- **Propriedade:** o último item inserido é o primeiro item que pode ser retirado da lista. São chamadas listas **LIFO** (“Last-In, First-Out”).
- Existe uma ordem linear para pilhas, do “mais recente para o menos recente”.
- É ideal para processamento de estruturas aninhadas de profundidade imprevisível.
- Uma pilha contém uma seqüência de obrigações adiadas. A ordem de remoção garante que as estruturas mais internas serão processadas antes das mais externas.

Propriedades e Aplicações de Pilhas

- Aplicações em estruturas aninhadas:
  - Quando é necessário caminhar em um conjunto de dados e guardar uma lista de coisas a fazer posteriormente.
  - O controle de seqüências de chamadas de subprogramas.
  - A sintaxe de expressões aritméticas.
- As pilhas ocorrem em estruturas de natureza recursiva (como árvores). Elas são utilizadas para implementar a **recursividade**.

Operações em Pilhas

- Conjunto de operações:
  - Esvazia (Pilha): faz a pilha ficar vazia.
  - Vazia (Pilha): verifica se a pilha está vazia.
  - Empilha (Pilha, x): insere o item x no topo da pilha.
  - Desempilha (Pilha): retira o elemento que está no topo da pilha.
  - Tamanho (Pilha): retorna o número de elementos na pilha.

Implementação do TAD Pilha

- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas.
- As duas representações mais utilizadas são as implementações por meio de vetores e de estruturas encadeadas.
- Vamos implementar um TAD Pilha?

Exemplo de TAD: Fila

- É uma estrutura para armazenar um conjunto de elementos, que funciona da seguinte forma:
  - Novos elementos sempre entram no fim da fila;
  - O único elemento que se pode retirar da fila em um dado momento é seu primeiro elemento (ou seja, o mais antigo na fila).
- Aplicação:
  - Modelar situações em que é preciso armazenar um conjunto ordenado de elementos, no qual o primeiro elemento a entrar no conjunto será também o primeiro elemento a sair do conjunto, e assim por diante.
- São chamadas listas **FIFO** (“First-In, First-Out”).

TAD Fila

- Como podemos implementar uma fila?
  - Lista simplesmente encadeada?
  - Lista duplamente encadeada?
  - Lista circular dinâmica?
  - Lista circular estática?

Dúvidas?