

TAD: Tipo Abstrato de Dados (parte 1)

SCC0202 – Algoritmos e Estruturas de
Dados I

Prof. Thiago A. S. Pardo

TADs e termos relacionados

- Termos relacionados, mas diferentes
 - Tipo de dados
 - Tipo abstrato de dados
 - Estrutura de dados

Tipo de dados

- Em linguagens de programação, o **tipo de uma variável** define o conjunto de valores que ela pode assumir e como ela pode ser manipulada
 - Por exemplo, uma variável **booleana** pode ser *true* ou *false*, sendo que operações de AND, OR e NOT podem ser aplicadas sobre ela
- **Novos tipos de dados** podem ser definidos em função dos existentes

3

Tipo de dados

- Por exemplo, a **declaração de uma variável** especifica
 - Quantidade de bytes que deve ser reservada a ela
 - Variação entre linguagens
 - Como é em C?
 - Como o dado representado por esses bytes deve ser interpretado
 - É inteiro ou real?

4

Tipo de dados

- **Perspectivas**

- **Computador:** formas de se interpretar o conteúdo da memória
- **Usuário:** o que pode ser feito em uma linguagem, sem se importar como isso é feito em baixo nível
 - Conceito
 - Também sentem isso?

5

Problema

- Como definir um **número racional**?

6

Problema

- Como definir um **número racional**?
- Possivelmente como
 - Um vetor de 2 elementos inteiros, cujo primeiro poderia ser o numerador e o segundo o denominador
 - Um registro de 2 campos inteiros: numerador e denominador
 - Etc.

7

Variação de implementação

- Há **diferentes implementações** possíveis para o mesmo tipo de dado para melhorar
 - Velocidade do código
 - Eficiência em termos de espaço
 - Clareza, etc.
- Todas definem o mesmo domínio e **não mudam o significado das operações**
 - Para racionais, podemos: criar, somar, multiplicar, ver se são iguais, imprimir, etc.

8

Substituição das implementações

- As mudanças nas implementações têm grande impacto nos programas dos usuários
 - Por exemplo
 - Re-implementação do código
 - Possíveis erros

9

Exemplo

```
programa sobre números racionais
início
  declarar i, vetor(10 linhas, 2 colunas) inteiros
  declarar media real
  media=0
  para i=1 até 10 faça
    ler vetor(i,1) e vetor(i,2)
    media=media + vetor(i,1)/vetor(i,2)
  media=media/10;
  imprimir(media);
  ...
fim
```

10

Exemplo

```
programa sobre números racionais
início
  declarar i, vetor(10 linhas, 2 colunas) inteiros
  declarar media real
  media=0
  para i=1 até 10 faça
    ler vetor(i,1) e vetor(i,2)
    media=media + vetor(i,1)/vetor(i,2)
  media=media/10;
  imprimir(media);
  ...
fim
```

Qual o impacto de se alterar a forma de representar os números racionais?

11

Pergunta principal

- Como podemos **modificar** as implementações dos tipos com o **menor impacto** possível para os programas que o usam?
- Podemos esconder (**encapsular**) de quem usa o tipo de dado **a forma como foi implementado**?




12

Tipo abstrato de dados

- Tipo de dados **divorciado da implementação**
 - Definido pelo par (v,o)
 - v: valores, dados a serem manipulados
 - o: operações sobre os valores/dados
- *Coleção bem definida de dados e um grupo de operadores que podem ser aplicados em tais dados*

13

Tipo abstrato de dados

mundo real	dados de interesse	ESTRUTURA de armazenamento	possíveis OPERAÇÕES
 pessoa	<ul style="list-style-type: none"> • a idade da pessoa 	<ul style="list-style-type: none"> • tipo inteiro 	<ul style="list-style-type: none"> • nasce ($i \leftarrow 0$) • aniversário ($i \leftarrow i + 1$)
 cadastro de funcionários	<ul style="list-style-type: none"> • o nome, cargo e o salário de cada funcionário 	<ul style="list-style-type: none"> • tipo lista ordenada 	<ul style="list-style-type: none"> • entra na lista • sai da lista • altera o cargo • altera o salário
 fila de espera	<ul style="list-style-type: none"> • nome de cada pessoa e sua posição na fila 	<ul style="list-style-type: none"> • tipo fila 	<ul style="list-style-type: none"> • sai da fila (o primeiro) • entra na fila (no fim)

14

TAD dicionário inglês-português

- Dados
 - ?

- Operações
 - ?

15

TAD dicionário inglês-português

- Dados
 - Pares de palavras

- Operações
 - Buscar tradução de uma palavra
 - Inserir novo par de palavras
 - Alteração de informação

16

Tipo abstrato de dados

- Os **dados armazenados** podem ser **manipulados** apenas pelos **operadores**
 - **Ocultamento** dos detalhes de representação e implementação, apenas funcionalidade é conhecida
 - **Encapsulam** dados e comportamento
 - Só se tem **acesso às operações** de manipulação dos dados, e não aos dados em si

17

Tipo abstrato de dados e estrutura de dados

- Uma vez que um TAD é definido, escolhe-se a **estrutura de dados** mais apropriada para representá-lo
 - Exemplos de estruturas de dados?

18

Exemplo

programa principal

```

programa sobre números racionais
  usar TAD de números racionais ←
início
  declarar i inteiro
  declarar r(10) racional
  para i=1 até 10 faça
    ler_numeros(r,i)
  calcular_media(r,10)
  ...
fim

```

TAD de números racionais

```

definir tipo racional vetor(2)

procedimento ler_numeros
início
...
fim

procedimento calcular_media
início
...
fim
...

```

Exemplo

programa principal

```

programa sobre números racionais
  usar TAD de números racionais ←
início
  declarar i inteiro
  declarar r(10) racional
  para i=1 até 10 faça
    ler_numeros(r,i)
  calcular_media(r,10)
  ...
fim

```

TAD de números racionais

```

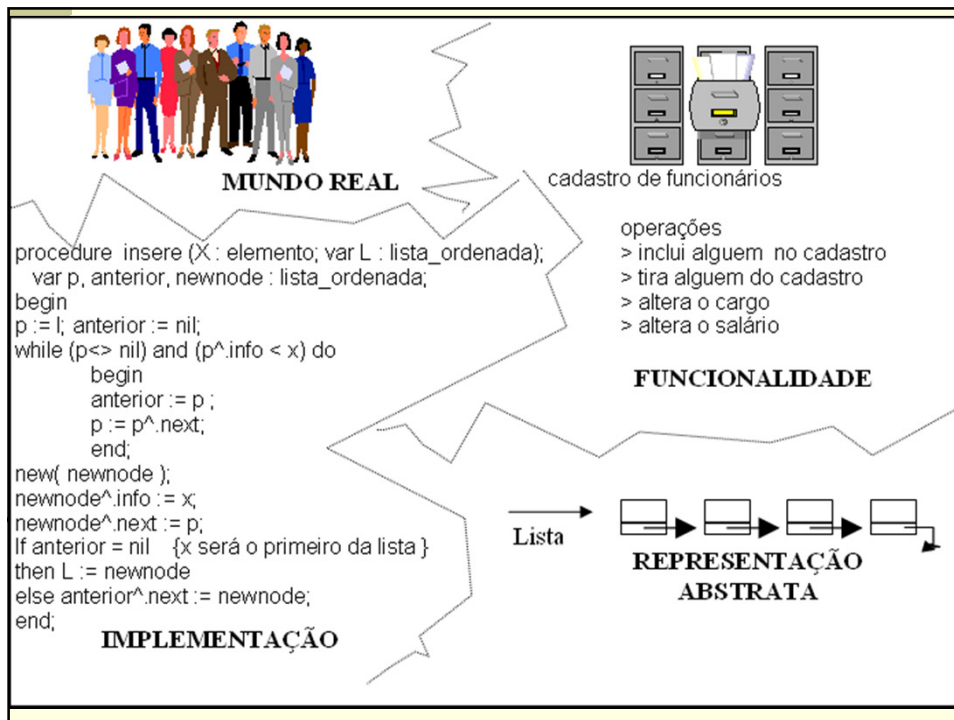
definir tipo racional vetor(2)

procedimento ler_numeros
início
...
fim

procedimento calcular_media
início
...
fim
...

```

Qual o impacto de se alterar a forma de representar os números racionais?



Tipo abstrato de dados

- **Vantagens**
 - Mais fácil programar
 - Não é necessário se preocupar com detalhes de implementação
 - Logicamente mais claro
 - Mais seguro programar
 - Apenas os operadores podem mexer nos dados
 - Maior independência, portabilidade e facilidade de manutenção do código
 - Maior potencial de reutilização de código
 - Abstração
- **Conseqüência:** custo menor de desenvolvimento

Tipo abstrato de dados

- Em termos de implementação, sugerem-se
 - **Passagem de parâmetros**
 - Um parâmetro pode especificar um objeto em particular, deixando a operação genérica
 - *Flag* para **erro**, sem emissão de mensagem no código principal
 - Independência do TAD

23

Tipo abstrato de dados

- Em **C/C++**
 - Dados/características/atributos
 - Operações/comportamentos/métodos
 - Modularidade
 - Herança
 - Objetos

24

Exercício

- Fazer um sistema de agenda de endereços
 - 1. Especificação do TAD
 - Dados/informação
 - Operações
 - 2. Implementação
 - Representação
 - Código

25

Operações: exemplo

- Dados: nome e endereço
- Operações de manipulação dos dados
 - está-na-lista?(nome)
 - põe-na-lista(nome,endereço)
 - tira-da-lista(nome)
 - pega-o-endereço(nome)

26

Operações: exemplo

- Dados: nome e endereço
- Operações de manipulação dos dados
 - está-na-lista?(nome)
 - põe-na-lista(nome,endereço)
 - tira-da-lista(nome)
 - pega-o-endereço(nome)
- Operações de interface (opcional)
 - que-operação-quer-fazer?(operação,nome,end)
 - aí-vai-a-resposta(string-resposta)

27

Interface: exemplo

- Repeat forever
- que-operação-quer-fazer?(op, nome, endereço)
- case op seja...
- 'inserir': se está-na-lista?(nome) = verdade
- então ai-vai-a-resposta('já está na lista!')
- senão põe-na-lista(nome, endereço)
- ai-vai-a-resposta('ok!')
- 'tirar': se está-na-lista?(nome) = verdade
- então tira-da-lista(nome)
- ai-vai-a-resposta('ok!')
- senão ai-vai-a-resposta('esse cara não está na lista')
- 'acha o endereço': se está-na-lista(nome) = verdade
- então pega-o-endereço (nome, endereço)
- ai-vai-a-resposta('endereço=', endereço)
- senão ai-vai-a-resposta('não achei o cara!')
- fim do case
- fim do repita forever

28