

TAD: Tipo Abstrato de Dados (parte 1)

SCC0502 – Algoritmos e Estruturas de
Dados I

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
 - Uma variável **inteira** pode receber números inteiros e ser submetida a operações aritméticas, 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

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?

Tipo de dados

- Por exemplo, a **declaração de uma variável** especifica
 - Quantidade de bytes que deve ser reservada a ela
 - Variação entre linguagens
 - char - 1 byte; integer - 4 bytes; float - 4 bytes; double - 8 bytes; etc.
 - Como o dado representado por esses bytes deve ser interpretado
 - É inteiro ou real?

Tipo de dados

■ Perspectivas

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

Problema

- *Você foi contratado para implementar um sistema de cálculos matemáticos que trabalha com números racionais (frações)*

Problema

- Como definir um número racional?

Problema

- Como definir um número racional?
- Possivelmente como
 - 2 inteiros, um para o numerador e outro para o denominador
 - 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.

Variação de implementação

- Há **diferentes implementações** possíveis para o mesmo tipo de dado, podendo priorizar:
 - 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.

Substituição das implementações

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

Exemplo

programa cálculos matemáticos

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(“a media do aluno é”, media);

 ...

fim

Exemplo

programa cálculos matemáticos

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(“a media do aluno é”, media);

...

fim

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

Exemplo

programa cálculos matemáticos

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(“a media do aluno é”, media);

...

fim

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

Exemplo

- E se quisermos um programa de ensino de números racionais, para uma escola, por exemplo?
- **Dá para reutilizar prontamente a implementação anterior?**

Exemplo

- E se quisermos um programa de ensino de números racionais, para uma escola, por exemplo?
 - Dá para reutilizar prontamente a implementação anterior? Não
 - Qual o problema?

Exemplo

- E se decidirmos que a forma de implementar os números racionais não foi a melhor e quisermos *alterar os programas* que fizemos anteriormente?
 - É prático?

Exemplo

- E se decidirmos que a forma de implementar os números racionais não foi a melhor e quisermos *alterar os programas* que fizemos anteriormente?
 - É prático? Não
 - Qual o problema?





Pergunta principal

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

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*

Tipo abstrato de dados

Mundo Real	Coleção Bem Definida de Dados	Grupo de Operadores
 pessoa	<ul style="list-style-type: none"> a idade da pessoa 	<ul style="list-style-type: none"> nasce (idade recebe o valor zero) aniversário (idade aumenta em 1)
 fila de espera	<ul style="list-style-type: none"> nome de cada pessoa e sua posição na fila 	<ul style="list-style-type: none"> sai da fila (o primeiro) entra na fila (no fim)
 cadastro de funcionários	<ul style="list-style-type: none"> o nome, cargo e o salário de cada funcionário 	<ul style="list-style-type: none"> entra no cadastro sai do cadastro altera o cargo altera o salário
 Pilha de Cartas	<ul style="list-style-type: none"> informações que identificam a carta (nipe, valor), e sua posição na pilha de cartas 	<ul style="list-style-type: none"> põe uma carta na pilha (no topo da pilha) retira uma carta da pilha (a do topo)

TAD dicionário inglês-português

- Dados

- ?

- Operações

- ?

TAD dicionário inglês-português

- Dados

- Pares de palavras

- Operações

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

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

Tipo abstrato de dados

■ Metáfora da TV



- Como você aumenta o volume da TV?
 - Você aperta o botão do volume?
 - Ou desmonta a TV e mexe nos componentes internos do autofalante?

Exemplo original

programa cálculos matemáticos

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(“a media do aluno é”, media);

 ...

fim

Exemplo com TAD

programa principal

```
programa cálculos matemáticos  
  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 com TAD

programa principal

```
programa ensino números racionais
  usar TAD de números racionais ←
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
fim
```

TAD de números racionais

```
definir tipo racional vetor(2)

procedimento ler_numeros
início
...
fim

procedimento calcular_media
início
...
fim

...
```

Exemplo com TAD

programa principal

```
programa ensino números racionais
  usar TAD de números racionais ←
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
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?

Exemplo com TAD

programa principal

```
programa ensino números racionais
  usar TAD de números racionais ←
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
fim
```

TAD' de números racionais

definir tipo racional **registro de 2 inteiros**

```
procedimento ler_numeros
início
...
fim
```

```
procedimento calcular_media
início
...
fim
```

...

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

Programa 1

```
programa ensino números racionais
  usar TAD de números racionais
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
fim
```

Programa 2

```
programa cálculos matemáticos
  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

...
```

Programa 1

```
programa ensino números racionais
  usar TAD de números racionais
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
fim
```

Programa 2

```
programa cálculos matemáticos
  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

...
```

Como fazer isso em C?

Exemplo em C

- Converter o TAD em algoritmo anterior (números racionais) para C
 - Dados?
 - Structs
 - Operações?
 - Funções
- Modo simplificado. Veremos o modo mais detalhado na próxima aula!

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

■ Consequência: custo menor de desenvolvimento

Tipo abstrato de dados

- Em termos de implementação, sugere-se
 - Passagem de parâmetros
 - Um parâmetro pode especificar um objeto em particular, deixando a operação genérica
 - Não importa se o objeto veio do programa 1 ou 2

Tipo abstrato de dados

- Em termos de implementação, sugere-se
 - *Flag* para erro, sem emissão de mensagem no código principal
 - Independência do TAD

Programa 1

```
programa ensino números racionais
  usar TAD de números racionais
início
  declarar r racional
  imprimir("Agora vamos aprender...")
  ler_numeros(r,1)
  ...
fim
```

Programa 2

```
programa cálculos matemáticos
  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
  ...
  imprimir("Aprendizado falhou!")
  ...
fim
```

Qual o problema?

Exercício em duplas

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

Parte 1: especificação do TAD

- Dados: nome e endereço
- Operações de manipulação dos dados
 - esta-na-agenda?(nome)
 - busca-o-endereço(nome)
 - insere-na-agenda(nome,endereço)
 - risca-da-agenda(nome)
 - Etc.

Parte 2: implementação

- Para entregar impresso no início da próxima aula
 - Implementação em C, usando uma IDE, por enquanto (DEV C++)
 - Modo simplificado