AULA 02

Tipos Abstratos de Dados (TADs)

Prof. Tiago A. Almeida
talmeida@ufscar.br

Definições

✔ Algoritmo

- Pode ser visto como uma sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema (Ziviani, 2003)

✔ Estruturas de Dados

- Organização de dados e operações (algoritmos) que podem ser aplicadas sobre esses como forma de apoio a solução de problemas (complexos)

✔ Programas

- Formulações concretas de algoritmos abstratos, baseados em representações e estruturas específicas de dados (Wirth, 1976) - algoritmos que podem ser executados em computadores

Tipos de Dados

Definição

- Caracteriza o conjunto de valores a que uma constante pertence, ou que podem ser assumidos por uma variável ou expressão, ou que podem ser gerados por uma função (Wirth, 1976)
  - Tipos simples: **`int`**, **`float`**, **`double`**, etc
  - Tipos estruturados: **`structs`**

Tipos Abstratos de Dados (TADs)

✔ Definição

- Pode ser visto como um modelo matemático, acompanhado das operações definidas sobre o modelo (Ziviani, 2003)
  - É usado para encapsular tipos de dados (pensar em termos das operações suportadas e não como são implementadas)
  - Não há necessidade de saber a representação interna de um tipo de dado
  - Não se preocupa com a eficiência de tempo e espaço, estas são questões de implementação

Tipos Abstratos de Dados (TADs)

✔ Modelos Matemático

- Um TAD pode ser visto como uma tupla (ν, o), onde
  - ν é o conjunto de valores
  - o é o conjunto de operações aplicadas sobre esses valores

- Exemplo, tipo REAL
  - $\nu = \mathbb{R}$
  - $o = \{+, -, *, /, =, <, >, <=, >=\}$

Tipos Abstratos de Dados (TADs)

✔ Implementação

- Uma vez definido um TAD e especificadas as operações associadas, ele pode ser implementado em uma linguagem de programação
  - É possível chegar a diversas implementações para um mesmo tipo abstrato de dados, cada uma delas apresentando vantagens e desvantagens em relação às outras

Exemplo de um TAD

TAD: Racional

- Conceito matemático de um número racional
  - Pode ser expresso como o quociente de dois inteiros
  - As operações definidas são:
    - criação de um número racional a partir de dois inteiros
    - adição
    - multiplicação

Exemplo de um TAD

```
/* definição de valor */
Inteiro numerador;
Inteiro denominador;

/* definição de comportamentos */
Racional criar(Inteiro var1, Inteiro var2)
Pré-condição :
   var2 != 0
Pós-condição :
   numerador = var1
   denominador = var2

Racional adição(Racional var1, Racional var2)
Pré-condição :
   nenhuma
Pós-condição :
   numerador = (var1.numerador * var2.denominador) + (var2.numerador *
var1.denominador)
   denominador = var1.denominador * var2.denominador

Racional multiplicação(Racional var1, Racional var2)
Pré-condição :
   nenhuma
Pós-condição :
    numerador = var1.numerador * var2.numerador
    denominador = var1.denominador * var2.denominador
```

Implementando o TAD

- Implementar significa mapear a estrutura de dados e as operações em uma linguagem de programação (que o computador entenda)
- No nosso caso: linguagem C

Implementando o TAD

```c
//estrutura de dados
typedef struct {
   int num;
   int den;
} Racional;


//operações (ou algoritmos)
void criar(Racional *rac, int num, int den) {
   ...
}

void adicao(Racional *v1, Racional *v2, Racional *res) {
   ...
}

void multiplicacao(Racional *v1, Racional *v2, Racional *res) {
   ...
}
```

Ver `Racional.c`

Exercício

Definição

- Escrever o TAD para números complexos. Sabe-se que um número complexo possui a forma: $z = a + bi$, onde a é a parte real e b é a parte imaginária, ambas representadas por valores reais. Sejam dois números complexos $z1 = a + bi$ e $z2 = c + di$:

Operações

- Criar um número complexo a partir de dois reais (`double`)
- Somar: $z1 + z2 = (a + c) + (b + d)i$
- Subtrair: $z1 - z2 = (a - c) + (b - d)i$
- Multiplicar: $z1.z2 = (ac - bd) + (ad + bc)i$
- Conjugado: $z = a - bi$
- Imprimir

Ver `Complexo.c`

Exercício

Complexo.h

```
//estrutura de dados
typedef struct {
      double real;
      double imag;
   } Complexo;

void criarComplexo(Complexo *c, double real, double imag);
void somarComplexo(Complexo *c1, Complexo *c2, Complexo *res);
void subtrairComplexo(Complexo *c1, Complexo *c2, Complexo *res);
void multiplicarComplexo(Complexo *c1, Complexo *c2, Complexo *res);
void conjugadoComplexo(Complexo *c, Complexo *res);
void imprimirComplexo(Complexo *c);
```

Teste

- $z1 = 5 + 8i$
- $z2 = 1 + 2i$

Implementação de um TAD

- Na implementação de um TAD, a escolha da estrutura de dados empregada tem papel importante
  - Uma escolha mal feita pode resultar em implementações ineficientes ou mesmo não-factíveis

Implementação de um TAD

- Pense na implementação de um TAD que represente um polinômio e suporte as seguintes operações
  - Soma de polinômios
  - Produto de polinômios
  - Avaliação de polinômios

- Como você definiria a estrutura de dados?

Implementação de um TAD

- Pense na implementação de um TAD que represente um polinômio e suporte as seguintes operações
  - Soma de polinômios
  - Produto de polinômios
  - Avaliação de polinômios

- Como você definiria a estrutura de dados?
  - Sugestão: um vetor poderia ser usado para guardar os coeficientes de um polinômio (isso funciona?)

AULA 02

Exercícios

Prof. Tiago A. Almeida
talmeida@ufscar.br