

# Algoritmos, Estruturas de Dados e Tipos Abstratos de Dados (TADs)

SCC0202 - Algoritmos e Estruturas de Dados I

Prof. Fernando V. Paulovich

*\*Baseado no material do Prof. Gustavo Batista*

<http://www.icmc.usp.br/~paulovic>

[paulovic@icmc.usp.br](mailto:paulovic@icmc.usp.br)

Instituto de Ciências Matemáticas e de Computação (ICMC)  
Universidade de São Paulo (USP)

25 de setembro de 2017



# Sumário

- 1 Algoritmos e Tipos de Dados
- 2 Tipos Abstratos de Dados (TADs)
- 3 Implementação de um TAD

# Sumário

- 1 Algoritmos e Tipos de Dados
- 2 Tipos Abstratos de Dados (TADs)
- 3 Implementação de um TAD

# Algoritmos, Estruturas de Dados e Programas

## Algoritmo

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

# Algoritmos, Estruturas de Dados e Programas

## Algoritmo

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

## Estrutura de Dados

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

# Algoritmos, Estruturas de Dados e Programas

## Algoritmo

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

## Estrutura 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**

# Sumário

- 1 Algoritmos e Tipos de Dados
- 2 Tipos Abstratos de Dados (TADs)**
- 3 Implementação de um TAD



# Tipos Abstratos de Dados (TADs)

## Definição

Pode ser visto como um modelo matemático, acompanhado das operações definidas sobre o modelo (Ziviani, 2003)

# 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)

# 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

# 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)

## Modelo Matemático

Um TAD pode ser visto como uma tupla  $(v, o)$ , onde

- $v$  é o conjunto de valores
- $o$  é o conjunto de operações aplicadas sobre esses valores

Exemplo, tipo REAL

- $v = \mathbb{R}$
- $o = \{+, -, *, /, =, <, >, <=, >=\}$

# Implementação de um TAD

## Implementação

Uma vez definido um TAD e especificadas as operações associadas, ele pode ser implementado em uma linguagem de programação

# Implementação de um TAD

## 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 de dados abstrato, 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

```
1  /* definição de valor */
2  Inteiro numerador;
3  Inteiro denominador;
4
5  /* definição de comportamentos */
6  Racional criar(Inteiro var1, Inteiro var2)
7  Pré-condição :
8     var2 != 0
9  Pós-condição :
10     numerador = var1
11     denominador = var2
12
13 Racional adição(Racional var1, Racional var2)
14 Pré-condição :
15     nenhuma
16 Pós-condição :
17     numerador = (var1.numerador * var2.denominador) + (var2.numerador * var1.denominador)
18     denominador = var1.denominador * var2.denominador
19
20 Racional multiplicação(Racional var1, Racional var2)
21 Pré-condição :
22     nenhuma
23 Pós-condição :
24     numerador = var1.numerador * var2.numerador
25     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)

# Implementando o TAD

- Implementar significa mapear a estrutura de dados e as operações em uma linguagem de programação (que o computador entenda)
- Nesse curso, a empregada será C

# Implementando o TAD

```
1  #ifndef RACIONAL_H
2  #define RACIONAL_H
3
4  struct racional {
5      int num;
6      int den;
7  };
8
9  typedef struct racional RACIONAL;
10
11 RACIONAL *criar(int num, int den);
12 void apagar(RACIONAL **rac);
13
14 RACIONAL *adicao(RACIONAL *v1, RACIONAL *v2);
15 RACIONAL *multiplicacao(RACIONAL *v1, RACIONAL *v2);
16
17 void imprimir(RACIONAL *rac);
18
19 #endif
```

# Implementando o TAD

```
1  #include "racional.h"
2
3
4  RACIONAL* criar(int num, int den) {
5  ...
6  }
7
8  void apagar(RACIONAL **rac) {
9  ...
10 }
11
12 void imprimir(RACIONAL *rac) {
13 ...
14 }
15
16 RACIONAL *adicao(RACIONAL *v1, RACIONAL *v2) {
17 ...
18 }
19
20 RACIONAL *multiplicacao(RACIONAL *v1, RACIONAL *v2) {
21 ...
22 }
```

# Implementando o TAD

```
1  #include "racional.h"
2
3  int main() {
4      RACIONAL *r1 = criar(1, 2);
5      RACIONAL *r2 = criar(1, 2);
6      imprimir(r1);
7      imprimir(r2);
8
9      RACIONAL *r3 = adicao(r1, r2);
10     imprimir(r3);
11
12     RACIONAL *r4 = multiplicacao(r1, r2);
13     imprimir(r4);
14
15     apagar(&r1);
16     apagar(&r2);
17     apagar(&r3);
18     apagar(&r4);
19
20     return 0;
21 }
```

# Exercício

## Definição

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

## Operações

- criação de um número complexo a partir de dois reais
- adição ( $z1 + z2 = a + c + (b + d)i$ )
- multiplicação ( $z1.z2 = (ac - bd) + (ad + cb)i$ )

## Exercício

Definir o TAD e implementar em linguagem C

# Sumário

- 1 Algoritmos e Tipos de Dados
- 2 Tipos Abstratos de Dados (TADs)
- 3 Implementação de um TAD



# 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
  - Avaliação de polinômios

# Implementação de um TAD

- Pense na implementação de um TAD que represente um polinômio e suporte as seguintes operações
  - Soma de polinômios
  - Avaliação de polinômios
  
- Como você definiria a estrutura de dados e os algoritmos empregados?

# Implementação de um TAD

- Pense na implementação de um TAD que represente um polinômio e suporte as seguintes operações
  - Soma de polinômios
  - Avaliação de polinômios
- Como você definiria a estrutura de dados e os algoritmos empregados?
  - Sugestão: um vetor poderia ser usado para guardar os coeficientes de um polinômio (isso funciona?)