

Estrutura de Dados

Tipos Abstratos de Dados

Prof^a. Anna Giselle Ribeiro

TIPOS ABSTRATOS DE DADOS

Objetivos

- Diferenciar conceitos:
 - Tipo de dados
 - Tipo abstrato de dados
 - Estrutura de dados

Cuidado para não confundir...

Tipo de dados,
Tipo abstrato de dados,
Estrutura de dados.

Tipo de Dados

- Tipo: Definição?
 - Tipo. *Def.*: Conceito que exprime a essência comum a um conjunto de coisas.
- Tipo de dados (em programação)
 - Conceito que exprime a **essência** de um conjunto de dados
 - **Valores** que os dados podem assumir
 - **Operações** que podem ser realizadas sobre estes dados

Tipo de Dados

- Ex.: Tipo de dado booleano:
 - Valores: True, False
 - Operações: AND, OR, NOT

Tipo de Dados

- Tipos primitivos
 - São definidos pela linguagem de programação
 - *Valores* assumidos pelos dados são mapeados diretamente para representações de máquina
 - *Operações* sobre os dados são mapeadas diretamente para operações de máquina

Tipo de Dados

- Tipos de dados definidos pelo programador
 - São definidos pelo programador com base nos tipos primitivos

```
Fracao {  
    int v[2];  
}
```

```
Fracao soma(Fracao f1, Fracao f2);
```


Tipo de Dados

- Tipos de dados definidos pelo programador
 - São definidos pelo programador com base nos tipos primitivos

```
Fracao {  
    int numerador;  
    int denominador;  
}
```

```
Fracao soma(Fracao f1, Fracao f2);
```

Perspectivas para Tipos de Dados

Computador

- Tipo de dado pode ser visto como um método para interpretar o conteúdo da memória do computador

Programador

- Tipo de dado pode ser visto como um método para definir o que o programador quer representar

Quando você usa *float*, você está preocupado com a forma de representação de um valor em ponto flutuante na memória, ou com o que deseja representar (ex.: a nota média de uma turma)?

O que é um Tipo Abstrato de Dado (TAD)?

Abstrato. *Def.:* O que se considera existente no domínio das ideias e sem base material.

Def.: Conceito que exprime
sem detalhes de implementação
os valores e as operações possíveis sobre um
conjunto de dados.

Tipo Abstrato de Dados (TADs)

Pensamos a respeito dos dados
em termos das
operações realizadas,
sem necessidade de saber
detalhes de implementação

Tipo Abstrato de Dados (TADs)

- TADs estão no centro de vários conceitos chave da Engenharia de Software:
 - Abstração
 - Modularidade
 - Encapsulamento

Abstração

Omitir ou esconder detalhes de baixo nível com uma ideia mais simples, de mais alto nível

Modularidade

Dividir um sistema em módulos, de modo que cada um seja projetado, implementado, testado, e reusado separadamente do resto do sistema.

Encapsulamento

Definir uma interface de acesso para um módulo de modo que ele seja responsável pelo seu próprio comportamento; assim, mudanças em outras partes do sistema não agridem sua integridade.

Tipo Abstrato de Dados (TADs)

```
Fracao {
```

?

```
}
```

```
Fracao Somar(Fracao f1, Fracao f2);  
Fracao Subtrair(Fracao f1, Fracao f2);  
Fracao Multiplicar(Fracao f1, Fracao f2);  
Fracao Dividir(Fracao f1, Fracao f2);
```

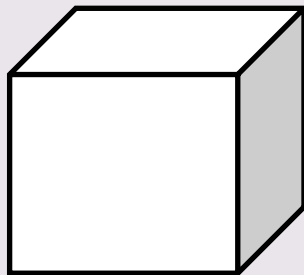
Tipo Abstrato de Dados (TADs)

- Usar TADs permitem um melhor “isolamento” entre partes do código

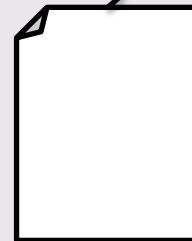
Tipo Abstrato de Dados (TADs)

Main.cpp

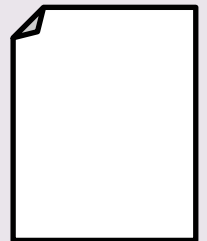
```
int main(void)
{
    Fracao f1 = CriarFracao(a, b);
    Fracao f2 = CriarFracao(c, d);
    Fracao soma = Somar(f1, f2);
}
```



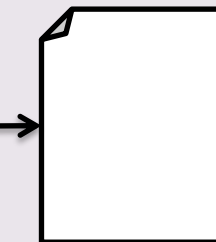
Executável 1



Fracao1.cpp



Fracao2.cpp



Fracao.h



Tipo Abstrato de Dados (TADs)

Main.cpp

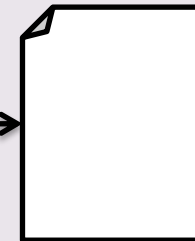
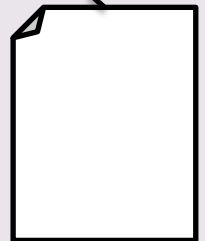
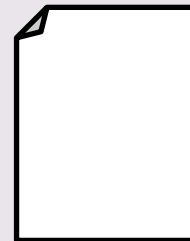
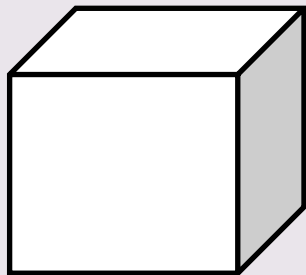
```
int main(void)
{
    Fracao f1 = CriarFracao(a, b);
    Fracao f2 = CriarFracao(c, d);
    Fracao soma = Somar(f1, f2);
}
```

Fracao.h

Fracao1.cpp

Fracao2.cpp

Executável 2



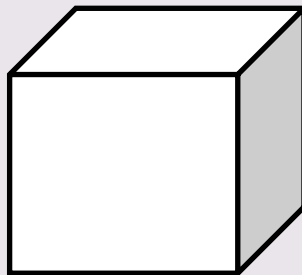
Tipo

Mudei a implementação de Fracao e não precisei mudar Main.cpp

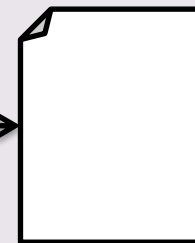
ADs)

Main.cpp

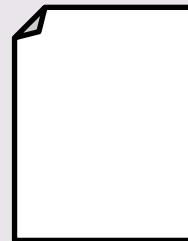
```
int main(void)
{
    Fracao f1 = CriarFracao(a, b);
    Fracao f2 = CriarFracao(c, d);
    Fracao soma = Somar(f1, f2);
}
```



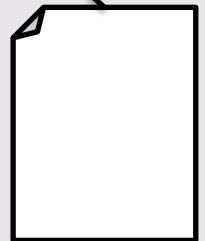
Executável 2



Fracao.h



Fracao1.cpp



Fracao2.cpp

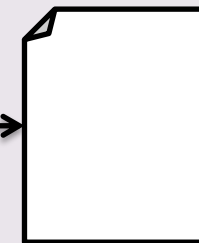


Tipo / (TADs)

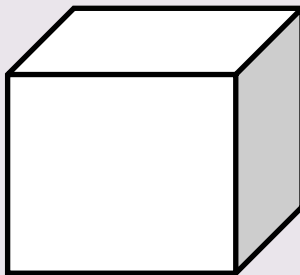
Se precisar mudar Fracao2.cpp, não preciso mudar outros arquivos

Main.cpp

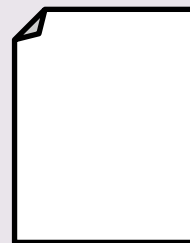
```
int main(void)
{
    Fracao f1 = CriarFracao(a, b);
    Fracao f2 = CriarFracao(c, d);
    Fracao soma = Somar(f1, f2);
}
```



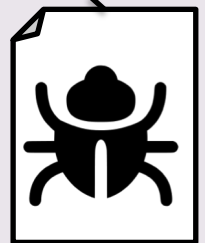
Fracao.h



Executável 2



Fracao1.cpp



Fracao2.cpp

Tipo Abstrato de Dados (TADs)

Programa

Usa o TAD

Interface

Especifica o TAD

Implementação

Implementa o TAD

Vantagens de usar TADs

- Reuso
- Manutenção

Tipo Abstrato de Dados

- Ex.: TAD Conjunto
 - Definição:
 - Coleção que não contém elementos duplicados
 - Operações:
 - Adicionar elemento
 - Remover elemento
 - Está vazio?
 - Contém determinado elemento?
 - ...

Exemplo – Interface

```
// Conjunto.h
```

```
Conjunto();
```

```
~Conjunto();
```

```
bool isEmpty();
```

```
void insert(Elemento);
```

```
void Remove(Elemento);
```

```
bool Contains(Elemento);
```

Exemplo – Uso

```
// Main.c

#include "Conjunto.h"

int main(){
    Conjunto c = new Conjunto();

    for(int i = 0; i < 10; i++){
        c.insert(i);
    }

    bool a = c.contains( 11 );

    return 0;
}
```

ESTRUTURA DE **DADOS**

O que é uma estrutura de dados?

Estrutura. *Def.:* Modo como as diferentes partes de um todo estão dispostas.

Def.: Representação concreta da estrutura de armazenamento, organização e manipulação de dados em um programa de computador.

Def.: Implementação de um TAD usando estruturas específicas.

Tipo abstrato de dados X Estrutura de dados

Propriedades abstratas

X

Detalhes concretos de implementação

Estrutura de dados

- Um TAD pode ser implementado por diferentes estruturas de dados
- TAD Conjunto
 - ED – Vetor
 - ED – Lista
 - ED – Árvore
 - ED – Tabela de dispersão

Qual a melhor implementação?

Seleção de estruturas de dados

- Quais as operações requeridas?
- Quais são as operações mais usadas?
- Qual a eficiência destas operações?

Revisão

1. Dada uma String, encontre o tamanho da maior substring sem repetir caracteres

Exemplo1 :

Input: "abcbcbcb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: "bbbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Revisão

1. Dado um array agrupe os anagramas

Example:

Input: ["eat", "tea", "tan", "ate", "nat", "bat"],

Output:

```
[ ["ate", "eat", "tea"],  
  ["nat", "tan"],  
  ["bat"] ]
```

Criar um board inicial de sudoku válido

Input:

```
[ ["5","3",".",".","7",".",".",".","."],  
  ["6",".",".","1","9","5",".",".","."],  
  [".","9","8",".",".",".",".","6","."],  
  ["8",".",".",".","6",".",".",".","3"],  
  ["4",".",".","8",".","3",".",".","1"],  
  ["7",".",".",".","2",".",".",".","6"],  
  [".","6",".",".",".",".","2","8","."],  
  [".",".",".","4","1","9",".",".","5"],  
  [".",".",".",".","8",".",".","7","9"] ]
```

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Output: true

Estrutura de Dados

Tipos Abstratos de Dados

Prof^a. Anna Giselle Ribeiro