

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Programação e Desenvolvimento de Software 2

Exemplos de TADs específicos

Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Introdução

- Quais exemplos de TADs já vimos?
 - Listas, Árvores binárias
- Será que existem quantos outros TADs?
 - Infinitos! Você pode definir os seus próprios!
- Sempre vou ter que implementá-los?
 - Não!
 - Bibliotecas to the rescue!

DCC UFMG

PDS 2 - Exemplos de TADs específicos 2

Introdução

- Geralmente, nenhum programa é escrito em linguagem de programação a partir do zero
- Bibliotecas
 - Autocontida, reutilizável e substituível
 - Importadas em múltiplas aplicações
- Você não precisa entendê-las completamente
 - Saber utilizar é o principal inicialmente!

DCC UFMG

PDS 2 - Exemplos de TADs específicos 3

Introdução

Reutilização

- Aplicado no dia-a-dia em vários contextos
- Maximizar o uso de software pré-existente
 - Tempo é dinheiro!
 - Por que reinventar a roda a todo momento?
- Tipos de reutilização
 - Objetos e Funções
 - Componentes
 - Sistemas

DCC UFMG

PDS 2 - Exemplos de TADs específicos 4

Bibliotecas vs. Frameworks

- Bibliotecas
 - Disponibiliza funcionalidades comuns
 - Conversão, comunicação, TADs, ...
- Frameworks
 - Pedacos de aplicações (incompletos)
 - Um sistema é implementado pela adição de componentes para preencher lacunas

DCC UFMG

PDS 2 - Exemplos de TADs específicos 5

Bibliotecas vs. Frameworks

DCC UFMG

PDS 2 - Exemplos de TADs específicos 6

Biblioteca padrão C++

- Classes, funções e variáveis escritas na linguagem para facilitar o desenvolvimento
- Componentes
 - Strings (expressões regulares)
 - Ponteiros inteligentes (unique_ptr, shared_ptr)
 - Entrada/Saída (streams)
 - Funcionalidades numéricas
 - Containers (STL)

Biblioteca padrão C++

- Funcionalidades através de headers

```
#include <string>
#include <iostream>
```

- Espaço de nomes (namespace)
 - std
- Headers da biblioteca padrão de C
 - <stdlib.h> → <cstdlib>

Biblioteca padrão C++

```
#include <iostream>

int main() {
    std::cout << "Hello world!" << std::endl;
    return 0;
}
```

```
#include <iostream>

using namespace std;

int main() {
    cout << "Hello world!" << endl;
    return 0;
}
```



Standard Template Library

- Parte da ISO C++ Standard Library
 - Economia de tempo e esforço
 - Componentes bem escritos e bem testados
- Soluções para problemas de programação que envolvem estruturas de dados
- Componentes
 - Containers
 - Iteradores
 - Algoritmos

Standard Template Library

```
struct NodeI {
    int data;
    NodeI* next;
};
```

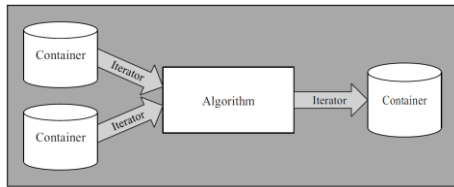
```
struct NodeA {
    Aluno data;
    NodeA* next;
};
```

```
template <typename T> class NodeG {
    T data;
    NodeG* next;
};
```

Standard Template Library

- Programação Genérica
 - A mesma definição de função atua da mesma forma sobre objetos de diferentes tipos
 - Polimorfismo universal – Paramétrico
 - Os tipos são passados como parâmetros
 - Código que pode ser reutilizado por classes em diferentes hierarquias de acordo com o tipo
- Linguagem
 - Templates (C++), Generics (Java)

Standard Template Library



DCC

PDS 2 - Exemplos de TADs específicos

13

Standard Template Library Containers

- Estruturas de dados que armazenam valores
 - Todos elementos são do mesmo tipo
 - Básicos ou definidos pelo programador (TADs)
- Praticamente todas os TADs são containers
 - Tipos específicos (estrutura rígida)
 - Tipos genéricos (podem ser reutilizados)

DCC

PDS 2 - Exemplos de TADs específicos

14

Standard Template Library Containers

- Acesso uniforme aos dados (contrato)
 - Independente de como está armazenado
 - Independente do tipo
- Recuperação dos dados
 - Índice (N-ésimo elemento)
 - Valor (Elemento com valor "João")
 - Propriedades (Elemento com "idade > 18")

DCC

PDS 2 - Exemplos de TADs específicos

15

Standard Template Library Containers

- Organização de dados
 - Acesso rápido, exibição, ...
- Operações (padronização)
 - Adicionar, remover, ordenar, buscar, ...
- Implementação
 - Correta, eficiente, documentação, ...

DCC

PDS 2 - Exemplos de TADs específicos

16

Standard Template Library Containers

Sequenciais	Associativos	Adaptadores
<ul style="list-style-type: none"> • Vector • Deque • List 	<ul style="list-style-type: none"> • Set • Map • Multiset • Multimap 	<ul style="list-style-type: none"> • Stack • Queue • Priority queue

DCC

PDS 2 - Exemplos de TADs específicos

17

Standard Template Library Containers

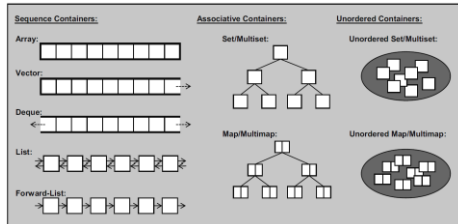
Container	Alocação de memória
Vector	Contígua
List	Node-wise
Set	Árvore binária
Map	Árvore binária

DCC

PDS 2 - Exemplos de TADs específicos

18

Standard Template Library Containers



DCC

PDS 2 - Exemplos de TADs específicos

19

Standard Template Library Containers - Sequenciais

- Elementos estão em uma ordem linear
 - Cada elemento é precedido por um elemento específico e seguido por outro
- Gerenciamento automático de memória que permite o tamanho variar dinamicamente
- Operações
 - `front()`, `back()`
 - `push_back()`, `pop_back()`
 - `size()`, `empty()`

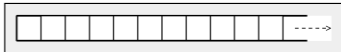
DCC

PDS 2 - Exemplos de TADs específicos

20

Standard Template Library Vector

- Modela um array genérico e dinamicamente redimensionável de elementos



- Vantagens
 - Acesso individual aos elementos (índice)
 - Adição/remoção de elementos no final*
 - Percorrer em uma ordem específica

<https://en.cppreference.com/w/cpp/container/vector>

DCC

PDS 2 - Exemplos de TADs específicos

21

Standard Template Library Vector

```
#include <iostream>
#include <vector>

int main() {

    std::vector<int> v = {7, 5, 16, 8};

    v.push_back(25);
    v.push_back(13);

    for(int n : v) {
        std::cout << n << std::endl;
    }

    return 0;
}
```

```
for (int i = 0; i < v.size(); i++)
    int n = v[i];
```

DCC

PDS 2 - Exemplos de TADs específicos

22

Standard Template Library Vector

- E se eu quisesse guardar "Pessoas"?

```
struct Pessoa {
    string nome;
    int idade;
};

#include <iostream>
#include <vector>
using namespace std;

int main() {
    Pessoa p1;
    p1.nome = "Ana";
    p1.idade = 23;

    Pessoa p2;
    p2.nome = "Pedro";
    p2.idade = 18;

    Pessoa p3;
    p3.nome = "Douglas";
    p3.idade = 30;

    vector<Pessoa> pessoas;
    pessoas.push_back(p1);
    pessoas.push_back(p2);
    pessoas.push_back(p3);
}
```

```
// Primeira forma de acesso
cout << pessoas[0].nome << endl;
cout << pessoas[1].nome << endl;

// Segunda forma de acesso
cout << pessoas.at(0).nome << endl;
return 0;
}
```

DCC

PDS 2 - Exemplos de TADs específicos

23

Standard Template Library List

- Lista duplamente encadeada



- Não provê acesso via índice
 - Iterador
- Adição/remoção é mais eficiente
 - Considerando-se que se está na posição
 - Não é necessário mover outros elementos

<https://en.cppreference.com/w/cpp/container/list>

DCC

PDS 2 - Exemplos de TADs específicos

24

Standard Template Library List

```
#include <iostream>
#include <list>

int main() {
    std::list<int> l = {7, 5, 16, 8};

    // Add an integer to the front of the list
    l.push_front(25);
    // Add an integer to the back of the list
    l.push_back(13);

    for (std::list<int>::iterator it=l.begin(); it != l.end(); ++it) {
        std::cout << *it << std::endl;
    }

    return 0;
}
```

DCC

PDS 2 - Exemplos de TADs específicos

25

Standard Template Library Containers - Associativos

- Elementos não possuem ordem específica
- Projetados para suportar o acesso direto aos elementos usando chaves pré-determinadas
- Chave
 - Usada internamente para guardar em ordem
- Operações
 - insert(), erase()
 - find(), count()

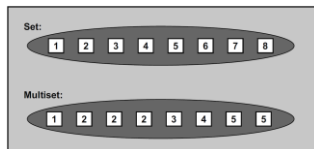
DCC

PDS 2 - Exemplos de TADs específicos

26

Standard Template Library Set

- Guarda uma coleção de elementos distintos
 - Dados armazenados (ordenados) em uma BST
 - Comparáveis de acordo com algum critério


<https://en.cppreference.com/w/cpp/container/set>

DCC

PDS 2 - Exemplos de TADs específicos

27

Standard Template Library Set – Exemplo 1

```
#include <iostream>
#include <set>

int main() {
    std::set<int> s;

    for(int i = 1; i <= 10; i++) {
        s.insert(i);
    }

    s.insert(7);

    for(int i = 0; i <= 10; i += 2) {
        s.erase(i);
    }

    std::cout << "s: " << s.size() << " " << std::endl;
    for (int e : s) {
        std::cout << e << std::endl;
    }

    return 0;
}
```

DCC

PDS 2 - Exemplos de TADs específicos

28

Standard Template Library Set – Exemplo 2

```
#include <iostream>
#include <set>

int main() {
    std::set<int> s = {1, 3, 5, 7, 9};

    if (s.find(6) != s.end()) {
        // Element in set
    } else {
        // Element not present in set
        s.insert(6);
    }

    if (s.insert(6).second) {
        // Element inserted in set
    } else {
        // Element NOT INSERTED in set
    }

    for (int e : s) {
        std::cout << e << std::endl;
    }

    return 0;
}
```

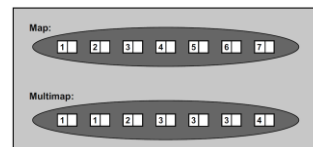
DCC

PDS 2 - Exemplos de TADs específicos

29

Standard Template Library Map

- Array associativo ou dicionário
 - Conjunto de pares <chave, valor>
 - As chaves devem ser distintas
 - Dados armazenados (ordenados) em uma BST


<https://en.cppreference.com/w/cpp/container/map>

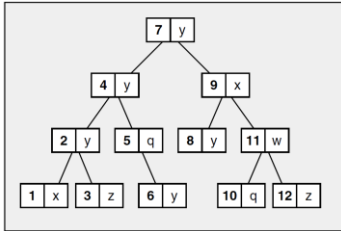
DCC

PDS 2 - Exemplos de TADs específicos

30

Standard Template Library

Map



Standard Template Library

Map

```
#include <iostream>
#include <string>
#include <map>

int main() {
    std::map<int, std::string> m;

    m.insert(std::pair<int, std::string>(2017123456, "Joao"));

    m[2016123456] = "Maria";
    m[2018123456] = "Carlos";
    m[2015123456] = "Jose";
    m[2014123456] = "Joana";

    std::map<int, std::string>::iterator it;
    for (it = m.begin(); it != m.end(); it++) {
        std::cout << it->first << " = " << it->second << std::endl;
    }

    return 0;
}
```

Considerações finais

```
struct Ponto {
    float x;
    float y;
};

void doSomethingOnList(list<Ponto> lista) {
    //Faz alguma coisa
}
```

Existe algum problema se
essa lista for muito grande?

Considerações finais

```
struct Ponto {
    float x;
    float y;
};

void doSomethingOnList(list<Ponto> const& lista) {
    //Faz alguma coisa
}
```

Considerações finais

