



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa



Projeto Olímpico de Programação

[C++] STL

STL

- Standard Template Library;
- Biblioteca padrão que disponibiliza a implementação das **Estruturas de Dados** mais utilizadas;
- Opa! **Isso é muito bom**, o programador ganha muito tempo. Agora ele não precisa mais “se preocupar” em implementar essas estruturas;
- **Mas ... o que são Estruturas de Dados?!**

STL

Antes ...

Precisamos entender alguns conceitos !

STL

Conceitos Básicos

Tipo de Dado

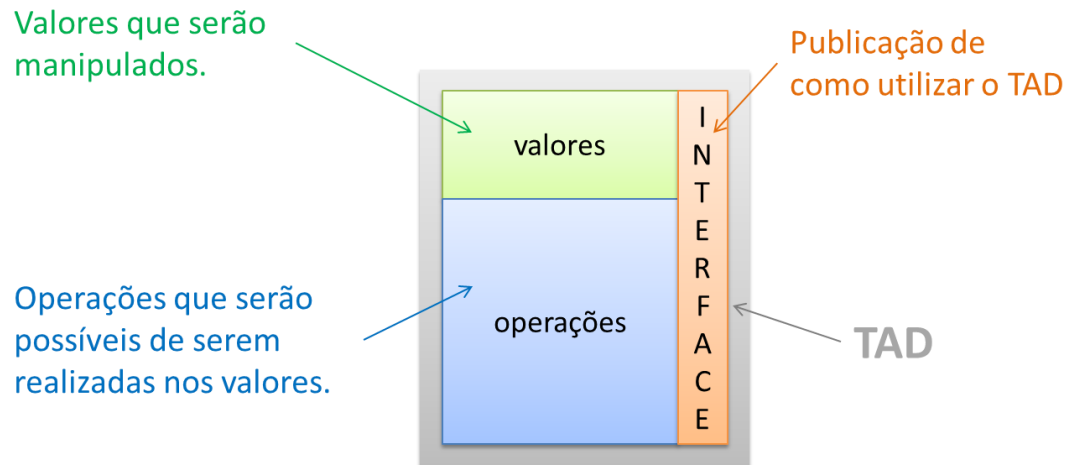
- “É um conceito abstrato, definido por um conjunto de propriedades lógicas;
- Os tipos de dados podem ser diferenciados pelo conjunto de **valores** que podem assumir, pelo conjunto de **operações** que podem ser efetuadas sobre esses dados, pela quantidade de **bytes** ocupada por eles e pela maneira como os **bits** serão interpretados e representados na memória”.

STL

Conceitos Básicos

Tipo Abstrato de Dado (TAD)

- Define uma coleção de **valores** e um conjunto de **operações** que podem ser realizadas sobre esses valores.

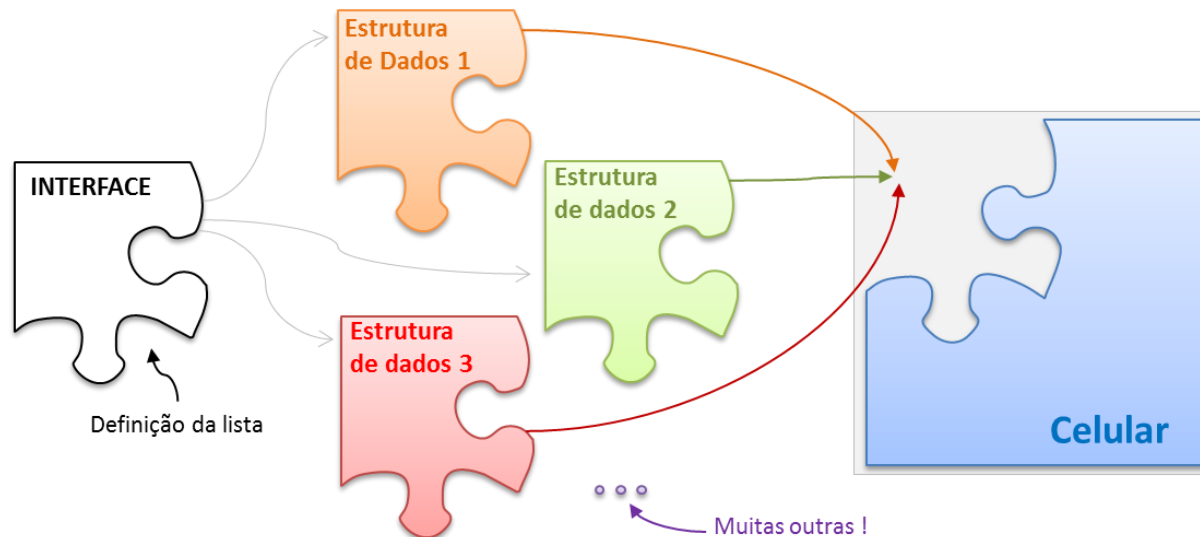


STL

Conceitos Básicos

Estrutura de Dados

- É uma forma particular de implementar um TAD.



STL

Conceitos Básicos

Estrutura de Dados

- Classicamente, pode ser implementada de duas formas:
 - Estática (uso de vetores);
 - Dinâmica (encadeamento de nó).

STL

Conceitos Básicos

Estrutura de Dados

- Exemplos de Estruturas de Dados:
 - Lista;
 - Pilha;
 - Fila;
 - Deque;
 - Mapa;
 - Árvore;
 - Grafo ...

STL

- Ok! Entendido o que é uma “Estrutura de Dados”!

A STL disponibiliza:

- Containers (repositório de dados);
- Iterators (Percurso no repositório);
- Algorithms (Manipulação no repositório).

Containers

<http://en.cppreference.com/w/cpp/container>
<http://www.cplusplus.com/reference/stl>

STL

Containers

Container	Descrição	Lib	Declaração
string	Array dinâmico de caracteres.	string	string s;
vector	Array dinâmico de objetos.	vector	vector<int> v;
list	Lista encadeada.	list	list<int> l;
stack*	Estrutura de dados “Pilha” (LIFO).	stack	stack<int> s;
queue*	Estrutura de dados “Fila” (FIFO).	queue	queue<int> q;
deque	Array dinâmico, com inserções e remoções apenas no início e/ou fim.	deque	deque<int> d;
set	Lista encadeada de elementos ordenados. Não permite repetição.	set	set<int> s;
map	Mapeamento de elementos “<chave, valor>”. O valor chave não pode repetir.	map	map<int,int> m;

Iterators

<http://en.cppreference.com/w/cpp/iterator>

STL

Iterators

Iterator	Descrição	Exemplo
back	Último elemento da coleção	<code>cout << v.back()</code>
begin	Iterator para o início da coleção	<code>cout << *v.begin()</code>
end*	Iterator para o final da coleção	<code>cout << *--v.end()</code>
front*	Primeiro elemento da coleção	<code>cout << v.front()</code>
advance**	Avança um iterator para determinada distância	<code>advance(it, 2);</code>
distance**	Distância entre dois iterators	<code>cout << distance(v.begin(), v.end());</code>
size	Quantidade de elementos da coleção	<code>cout << v.size()</code>
empty	Verifica se a coleção está vazia	<code>cout << v.empty()</code>

Algorithms

<http://en.cppreference.com/w/cpp/algorithm>

STL

Algorithms

Muitos ...

Os algoritmos e iteradores são aplicados aos containers!