

Programação Orientada a Objetos

Aula 04 – Biblioteca Padrão C++

Hugo Marcondes

Departamento Acadêmico de Eletrônica
DAELN

hugo.marcondes@ifsc.edu.br



Notes

STL – Standard Template Library

■ STL

- Conjunto de templates de classes com implementações de estrutura de dados e funcionalidades comum em programação

- Mas antes de entender os “componentes” da STL em si, vamos entender o que é um template de classe !

Notes



C++ Templates

- Templates C++ é uma poderosa construção da linguagem que permite a definição de uma classe genérica (um template)
 - Principalmente utilizado na generalização de tipos
- Quando a classe é utilizada (através da declaração/criação de um objeto) é necessário especificar quais são os tipos de dados que irão ser “preenchidos” na classe.
- Desta forma, classes genéricas são “especializadas” pelo compilador C++ no momento em que o código é gerado.



Notes

Template Class

- Uma classe genérica é definida através da declaração “template< typename T >”, antes de sua declaração, conforme abaixo:

```
1 template<typename T>
2 class MyClass {
3     T _attribute;
4
5     public:
6         void set_attribute(T parameter);
7         T get_attribute();
8 };
```



Notes

- Conjunto de componentes para tratar “problemas” comuns de programação
- “Containers”
 - Componentes utilizados para armazenar um conjunto de objetos (mesma classe)
 - Sequenciais
 - vector
 - deque
 - list
 - Associativos
 - set
 - map



- “Iterators”
 - Fornecem uma forma comum para acessar os objetos dentro dos “containers”
 - Cada container define o seu iterator
- “Algorithm”
 - Funções genéricas definidas na STL para interagir com os objetos de um container
 - Utilizam os iterators para acessar os containers de forma unificada
 - Ex. busca(find), ordenação(sort), etc



- Suporta o acesso randômico a seus componentes
- Tempo de inserção e remoção constantes no final do vetor
- Tempo linear de inserção/remoção no início e no meio do vetor
- Número de elementos pode variar e o gerenciamento da memória é automático

<http://www.cplusplus.com/reference/vector/vector/>



Vector – Exemplo Hello World

```
1 #include <vector>
2 #include <iostream>
3 using namespace std;
4
5 char* szHW = "Hello World";
6
7 int main(int argc, char* argv[])
8 {
9     vector<char> vec;
10    vector<char>::iterator vi;
11    char* cptr = szHW;
12
13    while (*cptr != '\0') {
14        vec.push_back(*cptr);
15        cptr++;
16    }
17    for (vi=vec.begin(); vi!=vec.end(); vi++) {
18        cout << *vi;
19    }
20    cout << endl;
21    return 0;
22 }
```



Containers Sequenciais

- deque: Double ended queue

<http://www.cplusplus.com/reference/deque/deque/>

- list: Lista (Lista duplamente encadeada)

<http://www.cplusplus.com/reference/list/list/>



Notes

Containers Associativos

- Set
 - Coleção de dados ordenados em uma estrutura de árvore binária balanceada
 - Facilita os algoritmos de busca
- Map
 - Coleção de dados associados através do uso de um par de chave e valor
 - Set na realidade é um tipo de Map, onde chave == valor



Notes

Exemplo de uso do Set

```
1 #include <string>
2 #include <set>
3 #include <iostream>
4 using namespace std;
5
6 int main(int argc, char* argv[])
7 {
8     set<string> strset;
9     set<string>::iterator si;
10    strset.insert("laranja");
11    strset.insert("maça");
12    strset.insert("limão");
13    strset.insert("banana");
14    strset.insert("uva");
15    strset.insert("uva");
16    // This one overwrites the previous occurrence
17    for (si=strset.begin(); si!=strset.end(); si++) {
18        cout << *si << " ";
19    }
20    cout << endl;
21    return 0;
22 }
```



Notes

Exemplo de uso do Map

```
1 #include <string.h>
2 #include <iostream>
3 #include <map>
4 #include <utility>
5
6 using namespace std;
7 int main()
8 {
9     map<int, string> Estudante;
10    map<int, string>::iterator ii;
11    Estudante[5234] = "Joao da Silva";
12    Estudante[3374] = "Carlos Pereira";
13    Estudante[1923] = "Bruna da Silva";
14    Estudante[7582] = "Jonas Macedo";
15    Estudante[5328] = "Vitor Machado";
16
17    cout << "Estudante[3374]=" << Estudante[3374] << endl << endl;
18    cout << "Map size: " << Estudante.size() << endl;
19
20    for(ii=Estudante.begin(); ii!=Estudante.end(); ++ii) {
21        cout << (*ii).first << ": " << (*ii).second << endl;
22    }
23 }
```



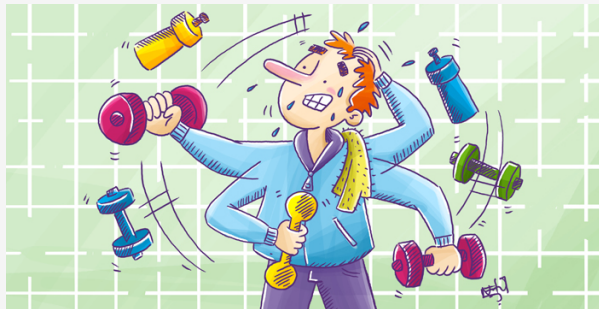
Notes

- <http://www.cplusplus.com/reference/stl/>
- <http://www.codeproject.com/Articles/6513/Practical-Guide-to-STL>
- <http://cs.brown.edu/~jak/proglang/cpp/stltut/tut.html>
- <http://www.yolinux.com/TUTORIALS/LinuxTutorialC++STL.html>

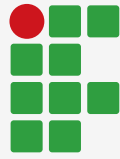


Vamos Praticar!

- Acessando a documentação do site <http://www.cplusplus.com/>, implemente um programa com a seguinte rotina:
 - Solicita para o usuário um conjunto de números inteiros (quantidade arbitrária)
 - Imprime este conjunto na ordem em que o usuário digitou
 - Apresenta qual é o menor número digitado
 - Apresenta qual é o maior número digitado
 - Imprime os números digitados em ordem crescente



That's all folks!



**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
Florianópolis



Notes

Notes
