

Một số lớp STL

Nguyễn Văn Khiết

Một số lớp STL (Standard Template Library)

- vector
- list

vector

- Lớp template, quản lý mảng các phần tử (mảng động)
- include <vector> khi sử dụng
- Cung cấp các cơ chế để thêm, xóa, duyệt các phần tử trong vector
- Cung cấp cơ chế lấy phần tử tại một vị trí bất kỳ của vector

vector

- `front()`
- `back()`
- `at()`
- `push_back(g)`
- `pop_back()`
- `begin()`
- `end()`
- `empty()`
- `insert()`

vector

- `erase()`
- `reverse()`
- `size()`
- `swap()`
- `capacity()`
- ...

vector

Ví dụ:

```
vector <int> vector1;
```

```
for (int i = 0; i < 5; ++i)  
    vector1.push_back(i);
```

```
cout << "Vector is : ";  
vector <int> :: iterator it;  
for(it = vector1.begin(); it != vector1.end(); ++it)  
    cout << ' ' << *it;  
cout << '\n';
```

list

- Lớp template, quản lý danh sách các phần tử
- Include <list> khi sử dụng
- Cung cấp các cơ chế để thêm, xóa, duyệt các phần tử trong list
- Không cung cấp cơ chế lấy phần tử tại một vị trí bất kỳ trong list

list

- `front()`
- `back()`
- `push_front(g)`
- `push_back(g)`
- `pop_front()`
- `pop_back()`
- `begin()`
- `end()`
- `empty()`
- `insert()`

list

- `erase()`
- `assign()`
- `remove()`
- `reverse()`
- `size()`
- `sort()`
- ...

list

Ví dụ:

```
list <int> list1;
```

```
for (int i = 0; i < 5; ++i)
{
    list1.push_back(i);
    list1.push_front(i * 2);
}
cout << "List is : ";
list <int> :: iterator it;
for(it = list1.begin(); it != list1.end(); ++it)
    cout << " << *it;
cout << '\n';
```