**STL迭代器**

**一. 迭代器类型**

**Input迭代器** istream

**Output迭代器** ostream

**Forward迭代器**

**Bidirectional迭代器** list, set, multiset, map,multimap

**Random access迭代器** vector, deque, string, array

注：随机迭代器，提供”迭代器算术运算”，比如：

Iter[n] 索引位置为n的元素

Iter+=n向前跳n个元素

Iter-=n 向后跳n个元素

Iter1 < iter2 判断iter1是否在iter2之前

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. for (**int** i=0;i<coll.size(); ++i) {
2. cout << coll.begin() [i]<< ' '; // iter[n]
3. }
4. for (pos =coll.begin(); pos < coll.end()-1; pos += 2) { //注意，只有随机迭代器才能用 operator <
5. cout << \*pos << ' ';
6. }

for (int i=0;i<coll.size(); ++i) {

cout << coll.begin() [i]<< ' '; // iter[n]

}

for (pos =coll.begin(); pos < coll.end()-1; pos += 2) { //注意，只有随机迭代器才能用 operator <

cout << \*pos << ' ';

}

**二. 迭代器相关辅助函数**

**1. advance()， 可将迭代器的位置增加或减少，增加的幅度由参数决定。**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <iterator>
2. void advance (InputIterator& pos, Dist n)

#include <iterator>

void advance (InputIterator& pos, Dist n)

**注**：n可以是负值，advance()不检查迭代器是否超过序列end()

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <iostream>
2. #include <list>
3. #include <algorithm>
4. using namespace std;
5. **int** main()
6. {
7. list<**int**> coll;
8. //insert elements from 1 to 9
9. for (**int** i=1; i<=9; ++i) {
10. coll.push\_back(i);
11. }
12. list<**int**>::iterator pos = coll.begin();
13. //print actual element
14. cout << \*pos << endl; // 1
15. //step three elements forward
16. advance (pos, 3);
17. //print actual element
18. cout << \*pos << endl; // 4
19. //step three elements backward
20. advance (pos, -1);
21. //print actual element
22. cout << \*pos << endl; // 3
23. }

#include <iostream>

#include <list>

#include <algorithm>

using namespace std;

int main()

{

list<int> coll;

//insert elements from 1 to 9

for (int i=1; i<=9; ++i) {

coll.push\_back(i);

}

list<int>::iterator pos = coll.begin();

//print actual element

cout << \*pos << endl; // 1

//step three elements forward

advance (pos, 3);

//print actual element

cout << \*pos << endl; // 4

//step three elements backward

advance (pos, -1);

//print actual element

cout << \*pos << endl; // 3

}

**2. distance()，用来处理两个迭代器之间的距离。**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <iterator>
2. Dist distance (InputIterator pos1, InputIterator pos2)

#include <iterator>

Dist distance (InputIterator pos1, InputIterator pos2)

**注：**两个迭代器必须指向同一个容器，如果不是随机迭代器，则pos2的位置必须与pos1相同或在其后。

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <iostream>
2. #include <list>
3. #include <algorithm>
4. using namespace std;
5. **int** main()
6. {
7. list<**int**> coll;
8. //insert elements from -3 to 9
9. for (**int** i=-3; i<=9; ++i) {
10. coll.push\_back(i);
11. }
12. //search element with value 5
13. list<**int**>::iterator pos;
14. pos = find (coll.begin(), coll.end(), //range
15. 5); //value
16. if (pos != coll.end()) {
17. //process and print difference from the beginning
18. cout << "difference between beginning and 5: "
19. << distance(coll.begin(),pos) << endl; //8, 第一个位置的distance是0
20. }
21. else {
22. cout << "5 not found" << endl;
23. }
24. }

#include <iostream>

#include <list>

#include <algorithm>

using namespace std;

int main()

{

list<int> coll;

//insert elements from -3 to 9

for (int i=-3; i<=9; ++i) {

coll.push\_back(i);

}

//search element with value 5

list<int>::iterator pos;

pos = find (coll.begin(), coll.end(), //range

5); //value

if (pos != coll.end()) {

//process and print difference from the beginning

cout << "difference between beginning and 5: "

<< distance(coll.begin(),pos) << endl; //8, 第一个位置的distance是0

}

else {

cout << "5 not found" << endl;

}

}

**3. iter\_swap()， 可交换两个迭代器所指的内容。**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <algorithm>
2. void iter\_swap (ForwardIterator1 pos1, ForwardIterator2 pos2)

#include <algorithm>

void iter\_swap (ForwardIterator1 pos1, ForwardIterator2 pos2)

注：交换迭代器pos1和pos2所指的值，迭代器类型不必相同，但其中所指的值必须可以相互赋值。

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7320674)

1. #include <iostream>
2. #include <list>
3. #include <algorithm>
4. #include "print.hpp"
5. using namespace std;
6. **int** main()
7. {
8. list<**int**> coll;
9. //insert elements from 1 to 9
10. for (**int** i=1; i<=9; ++i) {
11. coll.push\_back(i);
12. }
13. PRINT\_ELEMENTS(coll); // 1 2 3 4 5 6 7 8 9
14. //swap first and second value
15. iter\_swap (coll.begin(), ++coll.begin());
16. PRINT\_ELEMENTS(coll); // 2 1 3 4 5 6 7 8 9
17. //swap first and last value
18. iter\_swap (coll.begin(), --coll.end());
19. PRINT\_ELEMENTS(coll); // 9 1 3 4 5 6 7 8 2
20. }

#include <iostream>

#include <list>

#include <algorithm>

#include "print.hpp"

using namespace std;

int main()

{

list<int> coll;

//insert elements from 1 to 9

for (int i=1; i<=9; ++i) {

coll.push\_back(i);

}

PRINT\_ELEMENTS(coll); // 1 2 3 4 5 6 7 8 9

//swap first and second value

iter\_swap (coll.begin(), ++coll.begin());

PRINT\_ELEMENTS(coll); // 2 1 3 4 5 6 7 8 9

//swap first and last value

iter\_swap (coll.begin(), --coll.end());

PRINT\_ELEMENTS(coll); // 9 1 3 4 5 6 7 8 2

}

**三. 迭代器配接器**

1. Reverse(逆向)迭代器

2. Insert(插入)迭代器

3. Stream(流)迭代器