**STL 기초 – 반복자**

반복자 : 순회 방법을 일반화 하기위해 사용하는 개념

① 컨테이너의 요소 하나를 가리키는 기본적인 역할을 한다.

② 가리키는 지점의 요소를 읽고 쓸 수 있다. 내용을 읽는 \* 연산자가 정의된다.

③ 증감에 의해 주변 요소로 이동할 수 있다. ++, -- 등의 연산자가 정의된다.

④ 반복자끼리 대입, 비교 가능해야 한다. 대입, 비교 연산자가 정의된다

* 포인터는 위 4가지 기능을 모두 가지므로 그 자체로 완벽한 반복자이며 따라서 모든 STL 알고리즘에 포인터를 사용할 수 있다
* 모든 컨테이너는 시작점과 끝다음점을 조사하는 begin, end 멤버 함수를 제공한다

iterarray 예제

#include <iostream>

using namespace std;

void main()

{

int ari[]={1,2,3,4,5};

int \*it;

for (it=&ari[0];it!=&ari[5];it++) {

printf("%d\n",\*it);

}

}

itervector

#include <iostream>

#include <vector>

using namespace std;

void main()

{

int ari[]={1,2,3,4,5};

vector<int> vi(&ari[0],&ari[5]);

vector<int>::iterator it;

for (it=vi.begin();it!=vi.end();it++) {

printf("%d\n",\*it);

}

}

iterlist

#include <iostream>

#include <vector>

using namespace std;

void main()

{

int ari[]={1,2,3,4,5};

list<int> li(&ari[0],&ari[5]);

list<int>::iterator it;

for (it=li.begin();it!=li.end();it++) {

printf("%d\n",\*it);

}

}

itergeneric 예제

#include <iostream>

#include <vector>

#include <list>

using namespace std;

template<typename IT>

void Print(IT s, IT e)

{

IT it;

for (it=s;it!=e;it++) {

printf("%d\n",\*it);

}

}

void main()

{

int ari[]={1,2,3,4,5};

vector<int> vi(&ari[0],&ari[5]);

list<int> li(&ari[0],&ari[5]);

Print(&ari[0],&ari[5]);

Print(vi.begin(),vi.end());

Print(li.begin(),li.end());

}