Subject: PROPOSAL
Posted by kohait00 on Thu, 30 Sep 2010 14:16:41 GMT
View Forum Message <> Reply to Message

Here comes a proposal, meybe it is worth thinking about..

Iter<T>
ConstIter<T>

are interfaces to iterate through the containers (without beeing able to modify them)

currently supported containers:

ITER_CONT(Vector<T>)
ITER_CONT(Array<T>)
ITER_CONT(BiVector<T>)
ITER_CONT(BiArray<T>)
ITER_CONT(Index<T>)
ITER_CONT(ArrayIndex<T>)
ITER_CONT(Segtor<T>)

ITER_PTR(Ptr<T>)
ITER_PTR(One<T>)
ITER_PTR(T*)

Any
Value
Link<T>

the classes themselves are not changed at all.

```
Vector<int> vi;
vi.SetCount(10);
One<Iter<int> > ii = IterCreator::GetIter(vi);
while(ii->Next())
 ii->Get() = Random();

//or use macro
FOREACH(int, i, vi)
 i = Random();
```

there are also const variants, and respective macros

attached is a Test environment. it uses the Gen package from bazaar..for the Copyable interface.

a 'drawback' is the fact that the iter interfaces are all created on the heap to be able to clone them..here one can understand that i.e. C# enumerators rely on garbage collect features

nevertheless, i think it could be a helping package, when implementing abstraction layers..

please comment...

File Attachments

1) Iter.zip, downloaded 213 times