

Re: Best way to "redefine" a std class

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-12/1007.html

From: adbarnet (adbarnet_at_barnet.com)

Date: 12/07/04

Date: Tue, 7 Dec 2004 20:24:59 -0000

..as for the iterators – just typedef them:

```
template <typename T> myList {
public:
    typedef std::list<T> MyListType;
    MyListType::const_iterator const_iterator;

    const_iterator begin() const { return theList.begin(); }
    const_iterator end() const { return theList.end(); }

private:
    MyListType theList;
    MyListType::iterator iterator;
    iterator begin(){ return theList.begin(); }
    iterator end() { return theList.end(); }
};
```

"Nafai" <nafai3000@yahoo.es> wrote in message
news:Mogtd.4193329\$A6.12929524@telenews.teleline.es...
> *Hello I want to define a class myList, which is the same that std::list*
> *except from:*
> *– insert (I want to redefine it)*
> *– elements can only be consulted or deleted but not modified.*
> *– I want*
>
> *Which is the best way to do that?:*
>
> *1.*
> *template <typename T> myList : public list<T> {*
> *public:*
> *insert(...) { ... }*
> *...*
> *// what about iterators?*
> *};*
>
> *OR*
>

```
> 2.  
>  
> template <typename T> myList {  
> private:  
> list<T> theList;  
> public:  
> // all the methos of list addapted to myList  
> // i.e.:  
> int size() {return theList.size(); }  
> ...  
> // what about iterators?  
> };  
>  
>  
>  
>  
>
```