__asc_iterator.hpp

```cpp
#ifndef __ASC_ITERATOR_HPP__
#define __ASC_ITERATOR_HPP__

#include "franklist.h"

using namespace vhuk;

template <typename T>
FrankList<T>::asc_iterator::asc_iterator(const base_iterator& rhv)

    : const_asc_iterator(rhv){}

template <typename T>
FrankList<T>::asc_iterator::asc_iterator(base_iterator&& rhv)

    : const_asc_iterator(rhv){}

template <typename T>
typename FrankList<T>::reference FrankList<T>::asc_iterator::operator*()
{
    return (const_cast<reference>((static_cast<const_asc_iterator*>(this)->operator*())));
}

template <typename T>
typename FrankList<T>::pointer FrankList<T>::asc_iterator::operator->()
{
    return (const_cast<pointer>((static_cast<const_asc_iterator*>(this)->operator->())));
}

template <typename T>
const typename FrankList<T>::asc_iterator& FrankList<T>::asc_iterator::operator=(const base_iterator& rhv)
{
    *(static_cast<const_asc_iterator*>(this)) = rhv;

    return (*this);
}

template <typename T>
const typename FrankList<T>::asc_iterator& FrankList<T>::asc_iterator::operator=(base_iterator&& rhv)
{
    *(static_cast<const_asc_iterator*>(this)) = rhv;

    return (*this);
}

template <typename T>
FrankList<T>::asc_iterator::asc_iterator(Node* ptr)

    : const_asc_iterator{ptr}{}


#endif // __ASC_ITERATOR_HPP__

```