

__const_asc_iterator.hpp

```
1  #ifndef __CONST_ASC_ITERATOR_HPP__
2  #define __CONST_ASC_ITERATOR_HPP__
3
4  #include "franklist.h"
5
6  using namespace vhuk;
7
8  template <typename T>
9  FrankList<T>::const_asc_iterator::const_asc_iterator(const base_iterator& rhv)
10
11      : base_iterator(rhv.ptr){}
12
13  template <typename T>
14  FrankList<T>::const_asc_iterator::const_asc_iterator(base_iterator&& rhv)
15
16      : base_iterator(rhv.ptr){}
17
18
19  template <typename T>
20  const typename FrankList<T>::const_asc_iterator& FrankList<T>
21  ::const_asc_iterator::operator=(const base_iterator& rhv)
22  {
23      return (*this = std::move(rhv));
24  }
25
26  template <typename T>
27  const typename FrankList<T>::const_asc_iterator& FrankList<T>
28  ::const_asc_iterator::operator=(base_iterator&& rhv)
29  {
30      if (this != &rhv)
31          this->ptr = rhv.ptr;
32      return (*this);
33  }
34
35  template <typename T>
36  typename FrankList<T>::const_reference FrankList<T>::const_asc_iterator::operator*
37  () const
38  {
39      return (this->ptr->val);
40  }
41
42  template <typename T>
43  typename FrankList<T>::const_pointer FrankList<T>::const_asc_iterator::operator->()
44  const
45  {
46      return &(this->ptr->val);
47  }
48
49  template <typename T>
50  const typename FrankList<T>::const_asc_iterator& FrankList<T>
51  ::const_asc_iterator::operator++()
52  {
53      this->ptr = this->ptr->asc;
54      return (*this);
55  }
56
57  template <typename T>
58  const typename FrankList<T>::const_asc_iterator FrankList<T>
```

```
54 | ::const_asc_iterator::operator++(value_type)
55 | {
56 |     FrankList::Node* tmp = this->ptr;
57 |     ++(*this);
58 |
59 |     return (base_iterator(tmp));
60 | }
61 |
62 | template <typename T>
63 | const typename FrankList<T>::const_asc_iterator& FrankList<T>
64 | ::const_asc_iterator::operator--()
65 | {
66 |     this->ptr = this->ptr->desc;
67 |     return (*this);
68 | }
69 |
70 | template <typename T>
71 | const typename FrankList<T>::const_asc_iterator FrankList<T>
72 | ::const_asc_iterator::operator--(value_type)
73 | {
74 |     FrankList::Node* tmp = this->ptr;
75 |
76 |     ++(*this);
77 |
78 |     return (base_iterator(tmp));
79 | }
80 |
81 | template <typename T>
82 | FrankList<T>::const_asc_iterator::const_asc_iterator(Node* ptr)
83 | : base_iterator(ptr){}
84 |
85 | #endif // __CONST_ASC_ITERATOR_HPP__
86 |
```