

__const_desc_iterator.hpp

```

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

```

```
53 | {
54 |     FrankList::Node* tmp = this→ptr;
55 |
56 |     ++(*this);
57 |
58 |     return (base_iterator(tmp));
59 | }
60 |
61 | template <typename T>
62 | const typename FrankList<T>::const_desc_iterator& FrankList<T>
63 | ::const_desc_iterator::operator--()
64 | {
65 |     this→ptr = this→ptr→asc;
66 |     return (*this);
67 | }
68 |
69 | template <typename T>
70 | const typename FrankList<T>::const_desc_iterator FrankList<T>
71 | ::const_desc_iterator::operator--(value_type)
72 | {
73 |     FrankList::Node* tmp = this→ptr;
74 |
75 |     ++(*this);
76 |
77 |     return (base_iterator(tmp));
78 | }
79 |
80 | template <typename T>
81 | FrankList<T>::const_desc_iterator::const_desc_iterator(Node* ptr)
82 | : base_iterator{ptr}{}
83 |
84 |
85 | #endif // __CONST_DESC_ITERATOR_HPP__
86 |
```