

\_\_const\_iterator.hpp

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

```
54     this->ptr = this->ptr->next;
55     return (*this);
56 }
57
58 template <typename T>
59 const typename FrankList<T>::const_iterator FrankList<T>
::const_iterator::operator++(value_type)
60 {
61     FrankList::Node* tmp = this->ptr;
62
63     ++(*this);
64
65     return (base_iterator(tmp));
66 }
67
68 template <typename T>
69 const typename FrankList<T>::const_iterator& FrankList<T>
::const_iterator::operator--()
70 {
71     this->ptr = this->ptr->prev;
72     return (*this);
73 }
74
75 template <typename T>
76 const typename FrankList<T>::const_iterator FrankList<T>::const_iterator::operator-
-(value_type)
77 {
78     FrankList::Node* tmp = this->ptr;
79
80     --(*this);
81
82     return (base_iterator(tmp));
83 }
84
85
86 #endif // __CONST_ITERATOR_HPP__
87
```