

```
1
2
3 #include "CharacterFrequencyIterator.h"
4 #include <algorithm>
5
6
7 void CharacterFrequencyIterator::mapIndices() noexcept {
8     for (int i = 0; i < 256; i++) {
9         fMappedIndices[i] = (*fCollection)[i].character();
10    }
11    size_t i = 1;
12
13    while (i < 256)
14    {
15        size_t j = i;
16
17        while (j > 0 && std::less<CharacterMap>{}((*fCollection)[fMappedIndices[j - 1]], (*fCollection)[fMappedIndices[j]]))
18        {
19            std::swap(fMappedIndices[j - 1], fMappedIndices[j]);
20            j--;
21        }
22
23        i++;
24    }
25 }
26
27
28 CharacterFrequencyIterator::CharacterFrequencyIterator(const CharacterCounter*
    aCollection) noexcept :
29     fCollection(aCollection),
30     fIndex(0)
31 {
32     mapIndices();
33 }
34
35
36 const CharacterMap& CharacterFrequencyIterator::operator*() const noexcept {
37     return (*fCollection)[fMappedIndices[fIndex]];
38 }
39
40 CharacterFrequencyIterator& CharacterFrequencyIterator::operator++() noexcept {
41     fIndex++;
42     CharacterFrequencyIterator result = *this;
43     if ((*result).frequency() == 0) fIndex = 256;
44     return result;
45 }
46
47 CharacterFrequencyIterator CharacterFrequencyIterator::operator++(int) noexcept {
```

```
{
48     CharacterFrequencyIterator old = *this;
49     ++(*this);
50     return old;
51 }
52
53 bool CharacterFrequencyIterator::operator==(const CharacterFrequencyIterator&  ➤
    aOther) const noexcept{
54     return fIndex == aOther.fIndex && fCollection == aOther.fCollection;
55 }
56
57 bool CharacterFrequencyIterator::operator!=(const CharacterFrequencyIterator&  ➤
    aOther) const noexcept {
58     return !(*this == aOther);
59 }
60
61 CharacterFrequencyIterator CharacterFrequencyIterator::begin() const noexcept {
62     CharacterFrequencyIterator result = *this;
63     result.fIndex = 0;
64     return result;
65 }
66
67 CharacterFrequencyIterator CharacterFrequencyIterator::end() const noexcept {
68     CharacterFrequencyIterator result = *this;
69     result.fIndex = 256;
70     return result;
71 }
72
```