Chapter 14. Boost.Array

The library Boost.Array defines the class template `boost::array` in `boost/array.hpp`. `boost::array` is similar to `std::array`, which was added to the standard library with C++11. You can ignore `boost::array` if you work with a C++11 development environment.

With `boost::array`, an array can be created that exhibits the same properties as a C array. In addition, `boost::array` conforms to the requirements of C++ containers, which makes handling such an array as easy as handling any other container. In principle, one can think of `boost::array` as the container `std::vector`, except the number of elements in `boost::array` is constant.

Example 14.1. Various member functions of `boost::array`

```
#include <boost/array.hpp>
#include <string>
#include <algorithm>
#include <iostream>

int main()
{
  typedef boost::array<std::string, 3> array;
  array a;

  a[0] = "cat";
  a.at(1) = "shark";
  *a.rbegin() = "spider";

  std::sort(a.begin(), a.end());

  for (const std::string &s : a)
    std::cout << s << '\n';

  std::cout << a.size() << '\n';
  std::cout << a.max_size() << '\n';
}
```

As seen in Example 14.1, using `boost::array` is fairly simple and needs no additional explanation since the member functions called have the same meaning as their counterparts from `std::vector`.