

第 18 章 探讨 C++ 新标准

本章首先复习前面介绍过的 C++11 功能，然后介绍如下主题：

- 移动语义和右值引用。
- Lambda 表达式。
- 包装器模板 function。
- 可变参数模板。

本章重点介绍 C++11 对 C++ 所做的改进。本书前面介绍过多项 C++11 功能，本章首先复习这些功能，并详细介绍其他一些功能。然后，指出一些超出了本书范围的 C++11 新增功能（考虑到 C++11 草案的篇幅比 C++98 长 98%，本书无法全面介绍）。最后，将简要地探讨 BOOST 库。

18.1 复习前面介绍过的 C++11 功能

本书前面介绍过很多 C++11 改进，但您现在可能忘了，本节简要地复习这些改进。

18.1.1 新类型

C++11 新增了类型 `long long` 和 `unsigned long long`，以支持 64 位（或更宽）的整型；新增了类型 `char16_t` 和 `char32_t`，以支持 16 位和 32 位的字符表示；还新增了“原始”字符串。第 3 章讨论了这些新增的类型。

18.1.2 统一的初始化

C++11 扩大了用大括号括起的列表（初始化列表）的适用范围，使其可用于所有内置类型和用户定义的类型（即类对象）。使用初始化列表时，可添加等号（=），也可不添加：

```
int x = {5};
double y {2.75};
short quar[5] {4,5,2,76,1};
```

另外，列表初始化语法也可用于 `new` 表达式中：

```
int * ar = new int [4] {2,4,6,7};           // C++11
```

创建对象时，也可使用大括号（而不是圆括号）括起的列表来调用构造函数：

```
class Stump
{
private:
    int roots;
    double weight;
public:
    Stump(int r, double w) : roots(r), weight(w) {}
};
```