

刚开始不会觉得 `decltype` 有多大价值。但在模板的上下文中，`decltype` 会变得十分强大，详见第 12 和第 26 章。

1.1.34 标准库

C++ 具有标准库，其中包含许多有用的类，在代码中可方便地使用这些类。使用标准库中类的好处是不需要重新创建它们，也不需要浪费时间去实现系统已经自动实现的内容。另一好处是标准库中的类已经过成千上万用户的严格测试和验证。标准库中的类也经过了性能优化，因此使用这些类在大多数情况下比使用自己的类效率更高。

标准库中可用的功能非常多。第 16~24 章将详细讲述标准库。当开始使用 C++ 时，最好立刻了解标准库可以做什么。如果你是一位 C 程序员，这一点尤其重要。作为 C 程序员，使用 C++ 时可能会以 C 的方式解决问题，然而使用 C++ 的标准库类可以更方便、安全地解决问题。

这就是本章前面介绍标准库中的一些类的原因，例如 `std::string`、`array`、`vector`、`pair` 和 `optional`。本书一直在示例中使用这些代码，以确保你能够习惯使用标准库类。第 16~24 章将介绍更多的类。

1.2 第一个大型的 C++ 程序

下面的程序建立一个雇员数据库，在前面讨论结构体时曾将其用作示例。在此，将使用本章前面讲述的许多特性来完成一个功能完整的 C++ 程序。这个实际的示例使用了类、异常、流、`vector`、命名空间、引用及其他语言特性。

1.2.1 雇员记录系统

管理公司雇员记录的程序应该灵活并具有有效的功能。这个程序包含的功能有：

- 添加和解雇雇员
- 雇员晋升和降级
- 查看所有雇员，包括过去及现在的雇员
- 查看所有当前雇员
- 查看所有以前雇员

程序的代码分为三部分：`Employee` 类封装了单个雇员的信息，`Database` 类管理公司的所有雇员，单独的 `UserInterface` 提供程序的交互界面。

1.2.2 `Employee` 类

`Employee` 类维护某个雇员的全部信息，该类的方法提供了查询以及修改信息的途径。`Employee` 还知道如何在控制台显示自身。此外存在调整雇员薪水和雇佣状态的方法。

1. `Employee.cppm`

`Employee.cppm` 模块接口文件定义了 `Employee` 类，此文件的各部分分别在随后进行描述。文件的前几行如下：

```
export module employee;
import std;
namespace Records {
```