

结构、联合和枚举

函数延迟绑定：数据结构导致绑定。
记住：在编程过程后期再结构化数据。

本章介绍 3 种新的类型：结构、联合和枚举。结构是可能具有不同类型的值（成员）的集合。联合和结构很类似，不同之处在于联合的成员共享同一存储空间。这样的结果是，联合可以每次存储一个成员，但是无法同时存储全部成员。枚举是一种整数类型，它的值由程序员来命名。

在这 3 种类型中，结构是到目前为止最重要的一种，所以本章的大部分内容是关于结构的。16.1 节说明了如何声明结构变量，以及如何对其进行基本操作。随后，16.2 节解释了定义结构类型的方法，借助结构类型，我们就可以编写接受结构类型参数或返回结构的函数。16.3 节探讨如何实现数组和结构的嵌套。本章的最后两节分别讨论了联合（16.4 节）和枚举（16.5 节）。

16.1 结构变量

到目前为止介绍的唯一一种数据结构就是数组。数组有两个重要特性。首先，数组的所有元素具有相同的类型；其次，为了选择数组元素需要指明元素的位置（作为整数下标）。

结构所具有的特性与数组有很大不同。结构的元素（在 C 语言中的说法是结构的成员）可能具有不同的类型。而且，每个结构成员都有名字，因此为了选择特定的结构成员需要指明结构成员的名字而不是它的位置。

由于大多数编程语言提供类似的特性，因此结构可能听起来很熟悉。在其他一些语言中，经常把结构称为记录（record），把结构的成员称为字段（field）。

16.1.1 结构变量的声明

当需要存储相关数据项的集合时，结构是一种合乎逻辑的选择。例如，假设需要记录存储在仓库中的零件。每种零件需要存储的信息可能包括零件的编号（整数）、零件的名称（字符串）以及现有零件的数量（整数）。为了产生一个可以存储全部 3 种数据项的变量，可以使用类似下面这样的声明：

```
struct {
    int number;
    char name[NAME_LEN+1];
    int on_hand;
} part1, part2;
```

每个结构变量都有 3 个成员：number（零件的编号）、name（零件的名称）和 on_hand（现有数量）。注意，这里的声明格式和 C 语言中其他变量的声明格式一样。struct{...} 指明了类型，part1 和 part2 是具有这种类型的变量。