前言

本书运用演绎（deductive）方法于编程，将程序与使其工作的抽象数学理论相联系。将这些理论和依此理论所写的算法以及描述它们性质的定理和引理的规范（specification）一起介绍。算法在一门真实编程语言中的实现是本书的中心。写给人类的规范，应该甚至必须在适当非正式性（appropriate informality）上严格精准。而写给计算机的代码，必须在一般性上完全精准。

与科学和工程的其它领域一样，编程的适当基础是演绎方法。它运用特定数学行为，使复杂系统更易于分解为组成部分（components）。反之，它也是设计高效、可靠、安全和经济型（economical）软件的前提条件。

本书写给那些想要更深入理解编程的人，不论是全职的软件开发人员，还是以编程作为职业活动的重要环节的科学家和工程师们。

本书应该从头到尾阅读。读者只须阅读代码、证明引明、完成练习，便可理解书中的内容。除此之外，我还建议了若干项目，其中某些是开放式的（opne-ended）。本书虽简明扼要，但细心的最终会看到它的各部分之间联系和我们选材的理由，读者的最终目标是挖掘出（discovering）书的系统化的理论。

我们假定读书具有运用初等代数的能力。假定读者熟悉大学离散数学中逻辑学和集合理论的理本词汇。附录A总述了我们使用的符号。必要时，我们提供了一些抽象代数中的概念来指定（specify）算法。假定读者熟悉编程、理解计算机体系结构和基础算法和数据结构。

我们之所以选择C++，是因为它将强大的抽象机制和对底层机器的忠实表达相结合。我们使用了语言的一个小的子集，将需求写成结构化的解释，希望不熟悉C++的读者也能阅读本书。附录B指出了本书中使用的语言子集。在数学符号和C++混淆的地方，排版和上下文决定了是数学符号还是C++。本书中的许多概念和程序与STL（C++标准模板库）相对应。本书始自STL的设计决定。本书也忽略了某些讨论：一个真实库，例如STL，必须讨论：命名空间、可见性、内联指示符等等。

第1章描述值（values）、对象（objects）、类型（types）、过程（procedures）和概念（concepts）。第2-5章描述了代数结构上的算法，例如半群（semigroups）和全序集（totally ordered sets）。第6-11章描述了内存抽象上的算法。第12章描述了包括其它对象的对象。后记陈述了我对本书内容的一个整体映像。