

国家级精品课程系列教材
中国大学 MOOC 配套教材
教育部大学计算机课程改革项目成果



C 语言程序设计

第 3 版

董卫军 索琦 邢为民 编著
耿国华 主审

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



C 语言程序设计 3 版

本书特色

- **视角创新，主线清晰**
 结合一线教师教学经验与心得，以培养计算思维、强化计算思想、提升信息素质为主线。
- **问题导向，强化应用**
 突出知识点与实践应用的关联性，使读者掌握程序设计思想，提升创新思维能力。
- **内容翔实，结构科学**
 采用“核心语法 + 典型应用 + 知识扩展 + 疑难辨析”的内容组织方式，使读者理解核心语法、掌握典型应用，实现知识扩展为提升、疑难辨析以解惑的目标。
- **MOOC 在线，资源丰富**
 本书包含配套 MOOC，读者可在“中国大学 MOOC”平台学习，本书其他相关资源可登录华信教育资源网（www.hxedu.com.cn）下载。



策划编辑：戴晨辰
 责任编辑：靳平
 封面设计：欧美尼 李根星



定价：55.00 元

国家级精品课程系列教材
中国大学MOOC配套教材
教育部大学计算机课程改革项目成果

C 语言程序设计

(第 3 版)

董卫军 索琦 邢为民 编著
耿国华 主审

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是国家级精品课程“大学计算机”系列中“C 语言程序设计”课程的主教材，也是该课程中国大学 MOOC 配套教材。本书以快速掌握程序设计思想为主线，采用“核心语法为先导、实践应用为目的、知识扩展为提升、疑难辨析以解惑”的内容组织方式，突出知识点与技术点的关联性。全书内容主要包括基本数据类型、简单程序设计、循环结构程序设计、数组、指针、模块化程序设计、复杂数据类型、动态存储空间管理、文件的处理。

本书体系完整、结构严谨、注重应用、强调实践，在编写时兼顾了全国计算机等级考试的要求。为方便教学，本书还配有电子课件，任课教师可登录华信教育资源网（www.hxedu.com.cn）免费注册并下载。

本书既可作为高等学校“C 语言程序设计”课程的教材，也可作为全国计算机等级考试二级 C 语言程序设计的培训或自学教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

C 语言程序设计/董卫军，索琦，邢为民编著. —3 版. —北京：电子工业出版社，2019.10
ISBN 978-7-121-37598-9

I. ①C… II. ①董… ②索… ③邢… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 219769 号

策划编辑：戴晨辰

责任编辑：靳 平

印 刷：北京虎彩文化传播有限公司

装 订：北京虎彩文化传播有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：18.25 字数：470.4 千字

版 次：2011 年 6 月第 1 版

2019 年 10 月第 3 版

印 次：2019 年 10 月第 1 次印刷

定 价：55.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88258888，88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：dcc@phei.com.cn。

前 言

程序设计作为实现计算思维的核心课程之一，在大学生的知识体系中占有重要位置，其内容组织应该体现创造性思维的素质教育培养过程。MOOC（慕课）作为一种新型教学模式实现了教学过程的两个转变：一是由以教师为中心向以学习者为中心的转变；二是由被动学习向主动学习的转变。与传统课堂授课不同的是，MOOC 必须重新梳理和组织知识点，实现包含从授课到习题、讨论、答疑、测验，直到最终学习评价等环节的完整教学过程。

本书是国家级精品课程“大学计算机”系列中“C 语言程序设计”课程的主教材，也是该课程中国大学 MOOC 配套教材，还是“教育部大学计算机课程改革项目”成果之一。

在学习 C 语言程序设计时，要边学习、边思考、边消化、边实践。虽然有大量的网上资源，纸质教材仍是“慢思考”阅读的首选方式。本书在遵循教育和学习规律的基础上，按照应用需求梳理和组织 C 语言的知识点，内容编排由易到难、循序渐进，克服传统的“C 语言程序设计”教材以语法介绍为主、缺乏技能训练的不足。

本书以快速掌握程序设计为主线，采用“核心语法+典型应用+知识扩展+疑难辨析”的内容组织方式，达到“核心语法为先导、实践应用为目的、知识扩展为提升、疑难辨析以解惑”的目标，突出知识点与实践应用的关联性，使学习者在有限的时间内充分了解计算思维的基本过程，真正掌握程序设计思想，并能最终将这个过程融入创新思维活动之中。

全书共 9 章，对 C 语言程序设计的思想和方法从**基础技能**、**应用提升**两个层面逐层讲解。

- **基础技能层面**：从培养程序设计基本概念和基本逻辑思维能力入手，主要包括基本数据类型、简单程序设计、循环结构程序设计、数组、指针 5 个方面的知识，重点讲解程序设计的基本思想和 C 语言的基本数据类型，程序控制的基本构架，大量同类数据的存储与处理，以及如何通过指针解决问题。使学习者了解程序设计的基本思想，初步具备使用 C 语言解决问题的能力。
- **应用提升层面**：从培养分析问题和解决问题的能力入手，主要包括模块化程序设计、复杂数据类型、动态存储空间管理、文件的处理等 4 方面的内容，重点培养程序设计能力。使学习者初步掌握分析和解决问题的方法，真正掌握程序设计技能。

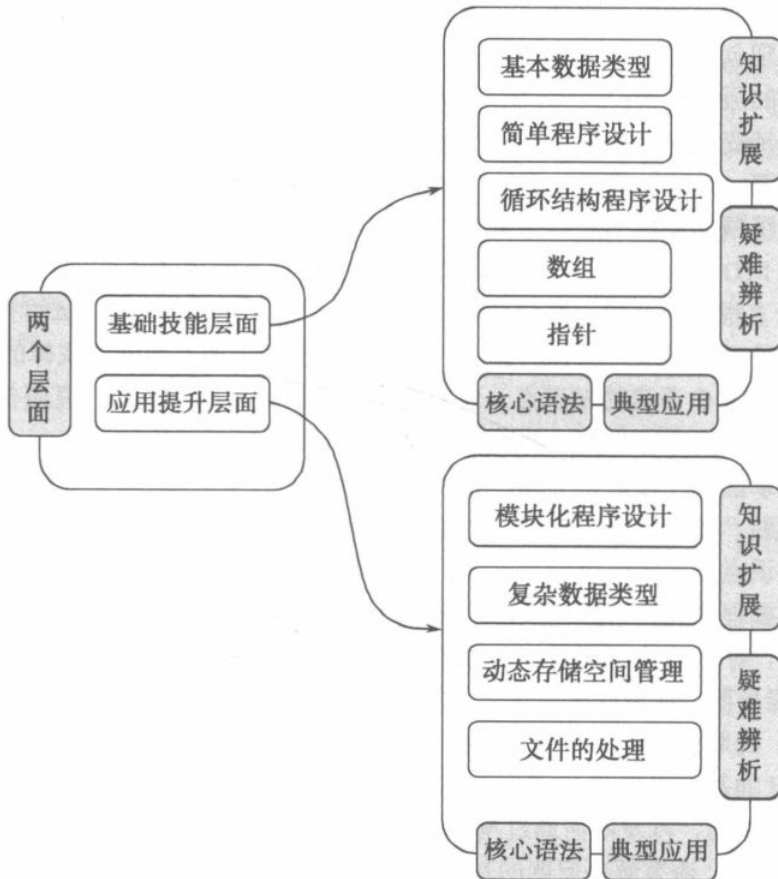
本书体系完整、结构严谨、注重实用、强调实践，在编写时兼顾了全国计算机等级考试的要求。为方便教学，本书还配有电子课件，任课教师可登录华信教育资源网（www.hxedu.com.cn）免费注册并下载。相应课程在中国大学 MOOC 定期开课（C 语言程序设计 董卫军）。另外，本书 C 语言程序中的变量符号为正体，为了保持一致，文中提及 C 语言程序中的变量符号也相应为正体。

本书由从事计算机教育多年的一线教师编写，董卫军担任主编，西北大学耿国华教授担任主审。由于作者水平有限，书中难免有不妥之处，恳请读者指正。

董卫军

于西安·西北大学

· 知识结构框图 ·



第 1 章 基本数据类型	1	2.2.1 if 语句	49
1.1 程序与工程化程序设计	1	2.2.2 switch 语句	51
1.1.1 程序	1	2.2.3 选择结构的嵌套	52
1.1.2 工程化程序设计	1	2.2.4 选择结构程序设计举例	54
1.2 程序设计语言	7	2.3 应用举例	55
1.2.1 程序设计语言的概念	7	2.4 疑难辨析	60
1.2.2 C 语言简介	7	习题 2	61
1.3 数据类型与常量	11	第 3 章 循环结构程序设计	66
1.3.1 数据类型	11	3.1 循环结构语句	66
1.3.2 常量	12	3.1.1 while 语句	66
1.4 变量	16	3.1.2 for 语句	68
1.4.1 标识符	16	3.1.3 循环结构程序设计	71
1.4.2 变量的定义	16	3.2 多重循环	73
1.5 基本运算	17	3.2.1 多重循环的引入	73
1.5.1 运算符与表达式	17	3.2.2 多重循环结构程序设计	73
1.5.2 赋值运算	18	3.3 知识扩展	77
1.5.3 算术运算	20	3.3.1 do...while 语句	77
1.5.4 关系运算	24	3.3.2 break 和 continue 语句	78
1.5.5 逻辑运算	25	3.4 应用举例	79
1.5.6 条件运算	26	3.5 疑难辨析	80
1.6 知识扩展	28	习题 3	81
1.6.1 数据的输入与输出	28	第 4 章 数组	87
1.6.2 逗号运算	31	4.1 一维数组的使用	87
1.6.3 数据类型长度运算	32	4.1.1 一维数组概述	88
1.6.4 算术自反赋值运算	33	4.1.2 一维数组应用举例	90
1.6.5 位运算	33	4.2 二维数组的使用	96
1.7 疑难辨析	37	4.2.1 二维数组概述	96
习题 1	40	4.2.2 二维数组应用举例	98
第 2 章 简单程序设计	47	4.3 知识扩展	100
2.1 顺序结构程序设计	47	4.3.1 字符串与字符数组	100
2.1.1 语句	47	4.3.2 三维数组的使用	104
2.1.2 顺序结构程序设计举例	48	4.4 应用举例	107
2.2 选择结构程序设计	49	4.5 疑难辨析	109

习题 4	111	7.1.4 结构体类型的指针	201
第 5 章 指针	118	7.2 共用体	203
5.1 指针与指针的运算	118	7.2.1 共用体类型及共用体变量的 定义	204
5.1.1 指针变量与指针常量	118	7.2.2 共用体变量的引用	205
5.1.2 指针的运算	120	7.3 知识扩展	207
5.2 指针与数组	124	7.3.1 枚举类型	207
5.2.1 指针与一维数组	124	7.3.2 用 typedef 定义类型	209
5.2.2 指针与二维数组	126	7.4 应用举例	210
5.3 知识扩展	130	7.5 疑难辨析	212
5.3.1 指针数组	130	习题 7	213
5.3.2 指向指针的指针	132	第 8 章 动态存储空间管理	219
5.4 应用举例	133	8.1 存储空间的分配与管理	219
5.5 疑难辨析	135	8.1.1 存储空间的组成	219
习题 5	140	8.1.2 静态和动态存储方式	220
第 6 章 模块化程序设计	146	8.1.3 静态和动态存储分配	220
6.1 模块化程序设计概述	146	8.2 动态存储空间的申请与释放	221
6.1.1 模块化程序设计的基本 思想	146	8.2.1 动态存储空间的申请	222
6.1.2 函数简介	147	8.2.2 动态存储空间的释放	223
6.2 函数的使用	148	8.3 链表的使用	224
6.2.1 自定义函数的定义	148	8.3.1 链表的概念	224
6.2.2 自定义函数的说明	151	8.3.2 链表的基本操作	226
6.2.3 函数调用	151	8.4 应用举例	230
6.2.4 函数使用举例	152	8.5 疑难辨析	242
6.3 知识扩展	156	习题 8	244
6.3.1 变量的存储类别、生存期与 作用域	156	第 9 章 文件的处理	248
6.3.2 函数的递归调用	163	9.1 文件的基本概念	248
6.3.3 理解函数指针	164	9.1.1 C 语言支持的文件格式	248
6.3.4 编译预处理	165	9.1.2 文件操作的基本思路	250
6.4 应用举例	171	9.2 文件的基本操作	250
6.5 疑难辨析	178	9.2.1 文件指针	250
习题 6	180	9.2.2 文件的打开与关闭	251
第 7 章 复杂数据类型	195	9.2.3 字节级文件的读/写	253
7.1 结构体	195	9.2.4 字符串文件的读/写	255
7.1.1 结构体类型与变量的定义	195	9.2.5 文件结束判断函数	256
7.1.2 结构体变量的使用	197	9.3 知识扩展	260
7.1.3 结构体数组	200	9.3.1 文件的格式化读/写	260

9.3.3 文件位置指针的移动·····	264	附录 C 标准 C 语言头文件·····	280
9.4 应用举例·····	266	附录 D C 语言系统关键字·····	281
9.5 疑难辨析·····	269	附录 E ASCII 编码表·····	282
习题 9·····	271	参考文献·····	283
附录 A Visual C++集成环境使用指南·····	275		
附录 B 常用运算符及其优先级和结合性·····	279		

第1章 基本数据类型

程序设计是软件构造活动中的重要组成部分，它以某种程序设计语言为工具，给出解决特定问题的求解步骤。数据类型、常量、变量及运算是编写程序的基础。数据类型定义了一个对象（如常量和变量）应具有何种数值及对其可进行什么样的运算。常量和变量是程序中最基本的数据处理对象。运算用运算符来描述，决定运算对象应该进行何种运算。

1.1 程序与工程化程序设计

1.1.1 程序

从自然语言角度来讲，程序是解决问题方法步骤的描述；从计算机角度来讲，程序是用计算机语言描述问题的求解过程。

程序的执行过程是问题的解决过程，因此，要想解决问题，首先要写出解决问题的正确步骤。

例如，求一元二次方程 $ax^2+bx+c=0$ （设 $a \neq 0$ ）实数根的步骤如下：

第一步 获得系数 a 、 b 、 c ；

第二步 计算 $d=b^2-4ac$ ；

第三步 若 $d > 0$

计算： $x_1 = (-b + \sqrt{d}) / (2a)$, $x_2 = (-b - \sqrt{d}) / (2a)$

输出：两个实根 x_1 和 x_2 ，转第六步；

第四步 若 $d < 0$

输出：没有实根，转第六步；

第五步 计算： $x_1 = x_2 = (-b) / (2a)$

输出：两个相同的实数根 x_1 ，转第六步；

第六步 结束。

程序设计的过程就是分析、解决问题的方法和步骤，并将其记录下来的过程。在描述问题求解步骤时，就要使用程序设计语言，C语言就是一种很好的程序设计语言。

1.1.2 工程化程序设计

随着计算机技术的发展，在计算机软件开发和维护过程中遇到了一系列严重问题，从而导致软件开发和维护日益复杂，这种现象称为软件危机。1968年，北大西洋公约组织的计算机科学家在联邦德国召开国际会议，讨论软件危机问题，正式提出并使用了“软件工程”这个名词，从此诞生了“软件工程”学科。

软件工程是指导计算机软件开发和维护的工程学科，通过采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间检验而证明正确的管理技术和当前能够得到的最好的软件开发方法结合起来。

软件工程的核心思想是把软件看成一个工程产品来处理,把需求分析、可行性研究、工程审核、质量监督等工程化概念引入软件生产中。软件工程包括 3 个要素:方法、工具和过程。方法是完成软件工程项目的手段;工具支持软件的开发、管理及文档的生成;过程支持软件开发中各个环节的控制和管理。通过这 3 个要素来达到工程项目的 3 个基本目标:进度、经费和质量。

同任何其他事物一样,一个软件产品也要经历孕育、产生、成长、成熟、衰亡等过程,这个过程一般称为软件生存期,即从软件产生直到消亡的时间间隔。可以把整个软件生存期划分为若干个生命周期,每经过一个生命周期,软件产品就经历一次升级。而一个生命周期又可分为若干个阶段,通常软件生命周期包括可行性分析、需求分析、系统设计(概要设计和详细设计)、编码、测试、维护等阶段,如图 1.1 所示。



图 1.1 软件生命周期的若干个阶段

1. 可行性分析

可行性分析决定“做还是不做”,是整个项目的第一步,其最根本的任务是对以后的行动方针提出建议。这一步涉及成本、人力资源、环境分析、预计回报等诸多因素。可行性分析不能以偏概全,必须为决策提供有价值的证据。

一般地,软件领域的可行性分析主要考虑 4 个要素:经济、技术、社会环境和人。

(1) 经济可行性分析主要包括成本分析和收益分析。成本一般包括开发成本、维护成本。收益包括短期收益和长远收益。

(2) 技术可行性分析就是分析参与开发人员的技术水平与软、硬件资源能否满足开发要求。技术可行性分析可以简单地表述为:是否能做?能否做得好?是否做得快?

(3) 社会环境可行性分析至少包括两种因素:市场与政策。市场又分为未成熟市场、成熟市场和将要消亡市场。

(4) 软件开发是复杂的过程,需要各种人员的相互配合,最好是人尽其才。

经过可行性分析,如果问题无法解决,分析员应该建议停止开发,以避免时间、资源、人力和资金的浪费;如果问题值得解决,分析员应该推荐一个较好的解决方案,并且为工程制订一个初步的计划。

2. 需求分析

需求分析处于软件开发过程的初期,它对于整个软件开发过程及软件产品的质量至关重要。在该阶段,开发人员要准确理解用户的要求,进行细致的调查分析,将用户非形式的需求转化为完整的需求定义,再由需求定义转化到相应的形式功能规约(需求规格说明)。随着软件系统复杂性的提高及规模的扩大,需求分析在软件开发中的地位愈加突出,需求分析也就愈加困难。

需求分析阶段要完成以下几个主要任务。

1) 问题识别

在功能需求上明确所要开发的软件必须具备的功能；在性能需求上明确所要开发的软件的技术性能指标；在环境需求上明确所要开发软件运行时所需要的软、硬件的要求；在用户界面需求上明确人机交互方式、输入/输出数据格式。

2) 分析与综合，建立软件的逻辑模型

分析人员对获取的需求进行一致性的分析检查，然后在分析、综合中逐步细化软件功能，划分成各个子功能，以图文结合的形式建立起软件的逻辑模型。

3) 编写文档

编写需求规格说明书，把双方共同的理解与分析结果用规范的方式描述出来，作为今后各项工作的基础。编写初步用户使用手册，着重反映所要开发软件的用户功能界面和用户使用的具体要求，用户手册能强制分析人员从用户的角度考虑软件。编写确认测试计划，作为今后确认和验收的依据。修改、完善软件开发计划，在需求分析阶段对所开发的系统要有进一步的了解，能更准确地估计开发成本、进度及资源要求，所以对原计划中不太合适的部分进行必要的修正。

需求分析在系统开发过程中的作用如图1.2所示。

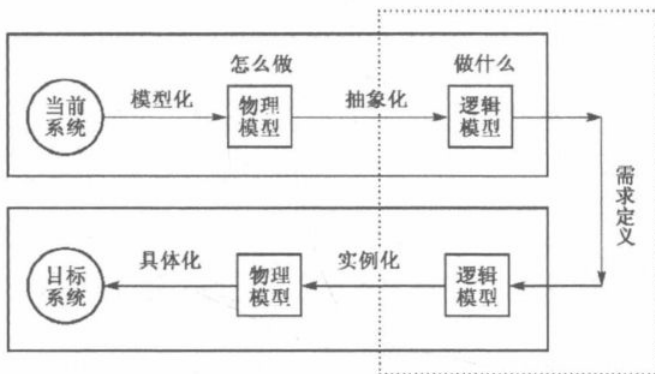


图 1.2 需求分析在系统开发过程中的作用

可以看出，需求分析的核心就是借助当前系统的逻辑模型导出目标系统的逻辑模型，解决目标系统“做什么”的问题。

需求分析有多种方法，常见的有面向数据流的结构化分析（SA）方法和面向对象的分析（OOA）方法。下面以 SA 方法为例来说明需求分析。

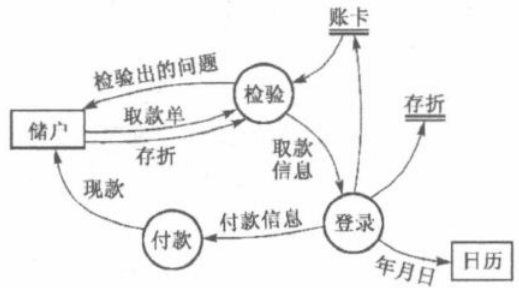
具体来说，SA 方法就是用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，直到找到满足功能要求的可实现的软件为止。

SA 方法使用的工具包括数据流图、数据词典、判定表与判定树。

数据流图表达了数据处理过程中对数据加工的情况。数据流图中的主要图形元素如图 1.3 所示。描述银行取款过程的数据流图如图 1.4 所示。

数据流图（Data Flow Diagram, DFD）是 SA 方法中用于表示系统逻辑模型的一种工具，它以图形的方式描绘数据在系统中的流动和处理过程，由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。数据词典（Data Dictionary, DD）用来定义数据流图中的各个元素的具体含义，它以一种准确的、无二义性的说明方式为系统的分析、设计及维护提供了

有关元素一致的定义和详细的描述。它和数据流图共同构成了系统的逻辑模型,是需求规格说明书的主要组成部分。在数据词典中,必须对数据流图中每个被命名的图形元素加以定义,其内容有名字、编号、描述、定义等。



判定树是用来表达加工逻辑的一种工具,它比判定表更直观。

3. 概要设计

在软件需求分析阶段,已经搞清楚了软件“做什么”的问题,并把这些需求通过需求规格说明书进行详细描述,这也是目标系统的逻辑模型。系统分析员审查软件计划、软件需求分析提供的文档,提出候选的最佳推荐方案供专家审定,审定后进行概要设计。

进入概要设计阶段,要把软件的逻辑模型变换为物理模型,即着手实现软件的需求,并将设计的结果反映在设计规格说明书中,所以概要设计是一个把软件需求转换为软件表示的过程,这种表示只是描述了软件总的体系结构,所以概要设计又称结构设计。

概要设计的基本过程如下。

- (1) 首先研究、分析和审查数据流图。
- (2) 然后从软件的需求规格说明书中厘清数据流加工的过程,对于发现的问题应及时解决。
- (3) 接下来根据数据流图决定数据处理问题的类型。数据处理问题可分为两种:变换型和事务型,针对数据处理问题的两种不同类型分别进行分析处理。
- (4) 最后由数据流图推导出系统的初始结构图,利用一些启发式原则来改进系统的初始结构图,直到得到符合要求的初始结构图为止。

4. 详细设计

详细设计又称过程设计或软件算法设计,该阶段不进行程序编码,是编码的先导,并为以后编码做准备。在这一阶段,对主要模块内部实现细节用到的算法要进行精确的表达,包括以下 6 项主要任务。

- (1) 为每个模块进行详细的算法设计。用图形、表格、语言等工具将每个模块处理过程的详细算法描述出来。
- (2) 对模块内的数据结构进行设计。对需求分析、概要设计确定的概念性的数据类型进行确切的定义。
- (3) 对数据结构进行物理设计。确定数据库的物理结构,物理结构主要是指数据库的存储记录格式、存储记录安排和存储方法,这些都依赖于具体使用的数据库系统。

- (4) 其他设计。
- (5) 编写详细的设计规格说明书。
- (6) 评审。对处理过程的算法和数据库的物理结构都要进行评审。

5. 编码

编码阶段的主要任务是使用选定的程序设计语言，把模块的过程性描述转换为用程序设计语言书写的源程序（源代码）。应根据项目的应用领域选择适当的编程语言和编程的软/硬件环境，同时应具有良好的程序设计风格。

6. 测试

程序编写完成后要对其进行测试，这是保证软件质量的关键步骤，是对软件规格说明、设计和编码的最后复审，其工作量约占总工作量的 40% 以上。测试的目的是尽可能发现并改正被测试程序中的错误，提高软件可靠性。测试的基本流程如图 1.5 所示。

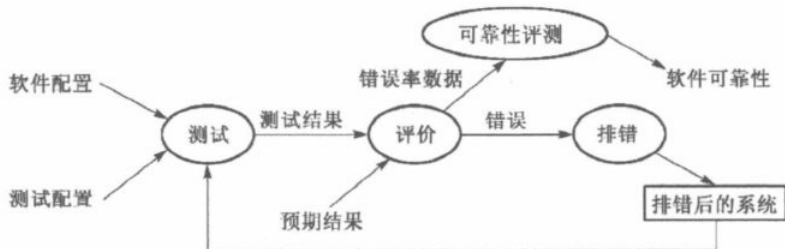


图 1.5 测试的基本流程

一般来说，测试主要分为两个层次：第一个层次是开发过程中的软件测试；第二个层次是第三方测试，测试与软件开发各阶段的关系如图 1.6 所示。



图 1.6 测试与软件开发各阶段的关系

1) 开发过程中的软件测试

开发过程中的软件测试是由软件开发方进行的测试，包括单元测试、集成测试、系统测试 3 个主要环节，其目的主要在于发现软件的缺陷并及时修改。

(1) 单元测试。单元测试针对编码过程中可能存在的各种错误，如用户输入验证过程中的边界值错误等。

(2) 集成测试。集成测试主要针对详细设计中可能存在的问题，尤其是检查各单元与其他程序之间的接口上可能存在的错误。

(3) 系统测试。系统测试主要针对概要设计，检查系统作为一个整体是否被有效地运行，如在软件设置中是否达到了预期的高性能。系统测试是保证软件质量的最后阶段。

2) 第三方测试

经集成测试和系统测试后,已经按照设计要求把所有模块组装成一个完整的软件系统,接口错误也已经基本排除,接着就要进行第三方测试。第三方测试有别于开发人员或用户进行的测试,其目的是保证测试工作的客观性,主要包括确认测试和验收测试。

(1) 确认测试。

确认测试又称有效性测试,是第三方测试机构根据软件开发商提供的用户手册,对软件进行的质量保证测试。确认测试的任务是验证软件的功能和性能及其他特性是否与用户的要求一致、是否符合国家相关标准法规、系统运行是否安全可靠等。

(2) 验收测试。

验收测试是软件开发结束后,用户对软件产品投入实际应用前进行最后一次质量检验活动。它不只是检验软件某个方面的质量,而是要进行全面的质量检验,并且要决定软件是否合格,因此验收测试是一项严格的正式测试活动,要根据事先制订的计划,进行软件配置评审、功能测试、性能测试等多方面检测。

3) 常见的测试方法

从测试是否针对系统的内部结构和具体实现算法的角度来看,测试可分为黑盒测试和白盒测试。

(1) 黑盒测试。

黑盒测试又称功能测试或数据驱动测试,它是在已知软件应具有功能的基础上,通过测试来检测每个功能是否都能正常使用。在测试时,把程序看成一个不能打开的黑盒子,在完全不考虑程序内部结构和内部特性的情况下,测试者在程序接口进行测试,它只检查程序功能是否符合需求规格说明书的规定,程序是否能接收输入数据产生正确的输出信息并保持外部信息(如数据库或文件)的完整性。

黑盒测试是穷举输入测试,不仅要测试所有合法的输入,还要对不合法的可能输入进行测试。黑盒测试主要有等价类划分、边值分析、因果图、错误推测等方法,主要用于软件确认测试。

(2) 白盒测试。

白盒测试又称结构测试或逻辑驱动测试,是指知道软件内部工作过程,通过测试来检测软件内部动作是否按照需求规格说明书的规定正常进行,按照程序内部的结构测试程序,检验程序中的每条通路是否都能按预定要求正确工作。白盒测试是穷举路径测试。在使用这一方案时,测试者必须检查程序的内部结构,从检查程序的逻辑着手,得出测试数据。

白盒测试的主要方法有逻辑覆盖法、基本路径测试法等,主要用于软件验证。

7. 维护

维护是指在软件交付使用后,为了改正错误或满足用户新需求而修改软件的过程。要求进行维护的原因多种多样,归结起来有以下3类。

(1) 改错需求。改正在特定使用条件下暴露出来的一些潜在程序错误或设计缺陷。

(2) 软件移植需求。如果在软件使用过程中数据环境发生变化(如一个事务处理代码发生改变)或处理环境发生变化(如安装了新的硬件或操作系统),则要修改软件以适应这种变化。

(3) 增加功能需求。用户和数据处理人员在使用软件时常会提出增加新功能及改善总体性能的要求, 为满足这些要求, 就要修改软件。

对于这3类问题主要进行的维护工作就是纠错性维护、适应性维护和完善性维护。

实践经验表明, 各种维护占整个维护工作的比例大致为: 纠错性维护占20%左右, 完善性维护占50%左右, 适应性维护占25%左右, 其他维护活动占5%左右。根据这些统计可以看出, 维护工作不仅是改错, 大部分维护工作是围绕软件的完善展开的。

1.2 程序设计语言

1.2.1 程序设计语言的概念

程序设计语言是用于编写计算机程序的一组记号和一组规则。程序设计语言包含3个方面, 即语法、语义和语用。语法表示程序的结构或形式, 即表示构成程序的各个记号之间的组合规则, 但不涉及这些记号的特定含义, 也不涉及使用者。语义表示程序的含义, 即表示按照各种方法所表示的各个记号的特定含义, 也不涉及使用者。语用表示程序与使用的关系。

程序设计语言的基本成分:

- 数据成分——用以描述程序所涉及的数据;
- 运算成分——用以描述程序中所包含的运算;
- 控制成分——用以描述程序中所包含的控制;
- 传输成分——用以表达程序中数据的传输。

1.2.2 C语言简介

C语言是一种比较流行的计算机程序设计语言。它既具有高级语言的特点, 又具有汇编语言的特点。它既可以作为系统程序设计语言, 编写系统软件, 也可以作为应用程序设计语言, 编写应用软件。它的应用范围广泛, 不仅应用在软件开发上, 在其他计算机应用领域也都用到了C语言, 如单片机及嵌入式系统开发。

1. C语言的特点

同其他程序设计语言相比, C语言具有以下主要特点。

(1) 语言简洁、紧凑, 使用方便、灵活。

C语言一共只有32个关键字、9种控制语句, 压缩了一切不必要的成分, 程序书写形式自由, 语句简洁。

(2) 运算符丰富, 适用的范围也很广泛。

C语言共有34种运算符, 它把括号、赋值符号、强制类型转换符号等都作为运算符处理, 从而使C语言的运算符类型丰富, 表达式类型多样, 灵活使用各种运算符可以实现用其他高级语言难以实现的运算和操作。

(3) 数据结构丰富, 具有现代化语言的各种数据结构。

C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用

体类型等。这些丰富的数据类型能用来实现各种复杂的数据结构（如链、表、树、栈等）的运算。

(4) 具有结构化的控制语句。

如 if...else 语句、switch 语句、while 语句、do...while 语句、for 语句等，这些语句可以实现程序中所有的控制结构。另外，函数是 C 语言程序的基本单位，将函数作为程序模块的基本单元，可以实现程序的模块化。

(5) 可直接对硬件操作。

C 语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能。这个特点使得 C 语言既具有高级语言的功能，又具有许多低级语言的特点，可以用来编写系统程序。

(6) 目标代码质量好，程序执行效率高。

C 语言生成的目标代码一般只比汇编语言生成的目标代码的效率低 10%~20%。

(7) 程序的可移植性好。

与汇编语言相比，用 C 语言编写的程序基本上不用被修改就能被用于各种型号的计算机和操作系统中，具备很好的移植性。

由于 C 语言灵活性大、功能强，可以编写出任何类型的程序，所以 C 语言一直是主流的程序设计语言。

2. C 语言程序的结构

下面通过几个简单的 C 语言程序，认识与体会 C 语言程序的结构。

【例 1.1】 在屏幕上输出一行字符串。

程序如下：

```
#include <stdio.h>           /*引用标准输入和输出函数*/
#include <stdlib.h>          /*引用标准 lib 库函数*/
void main()                 /*主函数，无返回值*/
{
    printf("This is a C program.\n"); /*语句*/
    system("pause");         /*暂停程序的执行，按任意键可以继续执行*/
}
```

此程序的功能是在屏幕上输出下面的一行信息：

```
This is a C program.
```

其中，main()是主函数，每个 C 语言程序都必须有一个 main()，表示整个 C 语言程序的入口。函数体用花括号“{ }”括起来。本例中主函数内有一个 printf 函数调用语句，它是 C 语言的输出函数，其功能是将双引号中的字符串原样输出。“\n”是换行控制符，即在“This is a C program.”输出之后回车换行。每个语句后要有一个分号(;)，表示语句结束。

注意：

① 调用标准输出 printf 函数，要在程序的前面用文件包含命令“#include <stdio.h>”。

② system 函数可使在桌面上直接运行编译后的可执行程序暂停执行，这样就可以在运行的窗口中看到运行结果，按任意键可以继续执行。但调用该函数，要在程序前面用文件包含命令“#include <stdlib.h>”。