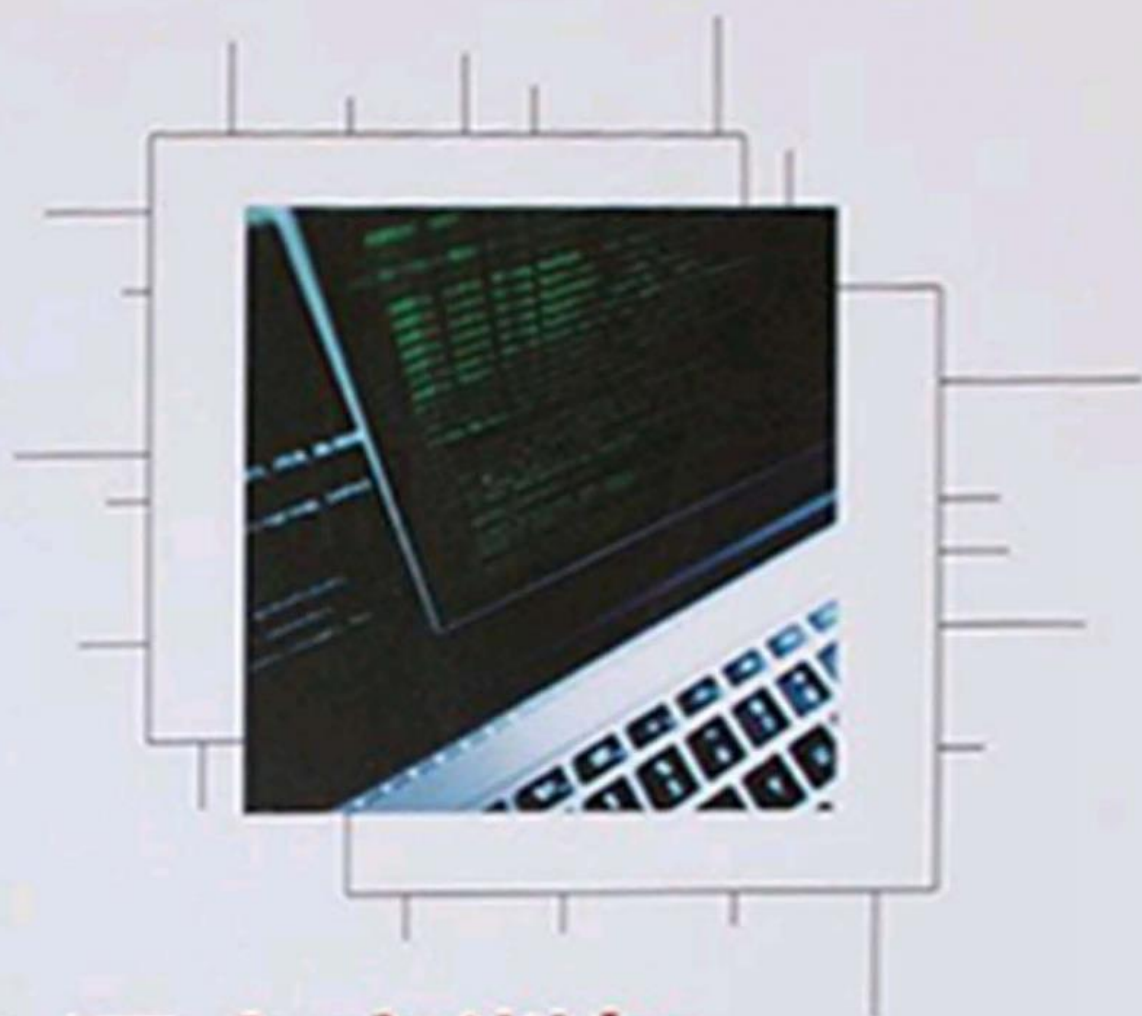




高等院校计算机类规划教材

全国高等院校计算机基础教育研究会立项项目成果



C语言实训与 考级教程

主 编 李志刚 刘 芳 邓玉洁
副主编 李海涌 贺娜娜



北京邮电大学出版社
www.buppress.com

内 容 简 介

为了帮助 C 语言初学者快速通过全国计算机等级(二级 C 语言)考试,切实提高其实战能力,本书根据作者多年讲授 C 语言程序设计课程、实训课程以及开发企业项目的经验编写,以培养读者计算思维能力为导向,第 1 部分针对新版考试大纲的要求,梳理和归纳 C 语言的基础知识和考点,并配有大量讲解详尽的例题和上机操作题,第 2 部分则针对复杂的综合应用问题,引入软件开发流程及企业编码规范,剖析应用中的难点和关键问题,指导读者完成一个六七百行代码的实训课大作业。

本书讲解由浅入深,由基础训练到综合训练循序渐进,可以作为大专院校学生 C 语言考级辅导教材,也可以作为 C 语言实训课程参考书,还可以作为 C 语言初学者的参考书。

图书在版编目(CIP)数据

C 语言实训与考级教程 / 李志刚, 刘芳, 邓玉洁主编. -- 北京: 北京邮电大学出版社, 2023.8

ISBN 978-7-5635-6978-6

I. ①C… II. ①李… ②刘… ③邓… III. ①C 语言—程序设计—水平考试—教材 IV. ①TP312.8

中国国家版本馆 CIP 数据核字(2023)第 143721 号

策划编辑: 马晓仟 责任编辑: 王小莹 责任校对: 张会良 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷:

开 本: 787 mm×1 092 mm 1/16

印 张: 18

字 数: 473 千字

版 次: 2023 年 8 月第 1 版

印 次: 2023 年 8 月第 1 次印刷

ISBN 978-7-5635-6978-6

定价: 47.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

当前信息技术日新月异,人工智能、5G 移动通信、区块链、物联网、云计算、大数据等新概念和新技术,给社会经济、人文科学、自然科学等领域带来了一系列革命性的突破。信息技术已经融入了人类社会生活的方方面面,深刻改变了人类思维、生产、生活、学习的方式,深刻展示了人类社会发展的前景。随着信息技术的全面发展,无处不在的计算思维能力成为人们认识 and 解决问题的能力的基本能力之一。

思维模式是人们看待世界和认识世界的方法与观点,也就是我们所说的世界观。在科学思维中,有以物理学为代表的实证思维和以数学为代表的逻辑思维,其看待世界的观点是不同的,并且形成了判断结论是否正确的不同标准。而计算思维是指运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为。

计算思维能力不仅是计算机专业学生应该具备的能力,而且也是所有大学生应该具备的能力。并非每一个大学生都要成为计算机科学家,但他们绝大多数都会从事科学研究、社会研究或者各种社会实践活动,掌握计算思维的基本方式对他们终身有益的。

C 语言是当今最为流行的程序设计语言之一,C 语言程序设计课程是提高人们计算思维水平的一门基础性课程,也是计算机相关专业学生的必修课。学生学习 C 语言程序设计的基础知识,并通过上机操作开展基础编程练习与综合编程练习,是掌握 C 语言程序设计基础知识和基本技能的关键步骤,同时也是培养和锻炼其计算思维能力的良好途径。

本书的各位编者全部来自教授 C 语言程序设计课的一线教师,都具有 5 年以上的教学经验和工程项目开发经验,在 C 语言教学的各个环节,从基础知识讲解、上机技能培训、实训课指导到 C 语言二级考级辅导都积累了丰富的经验,每年都为多个班级不同专业的学生教授 C 语言程序设计课及 C 语言实训课,帮助参加课程的学生在程序设计能力及计算思维能力上取得了很大的进步。编者编写本书是为了给大学生,尤其是准备 C 语言二级考级的大学生,提供一本合适的学习和练习教材。

本书 21 章分为两部分:C 语言考级教程和 C 语言实训任务。第 1 部分 C 语言考级教程(1~13 章)包含了 C 语言二级考试除公共知识外的所有知识点。第 1 章主要介绍计算思维的基本概念,学习编程语言时不仅要学习其语法规则,掌握编程技能,也要自觉运用

计算思维方法,在循序渐进的程序设计中培养计算思维能力,这种理念贯穿于后续各个章节。第2~12章是基础知识,包括C语言程序的组成结构、变量与常量、运算符与表达式、输入输出函数、选择结构、循环结构、数组、函数、指针、编译预处理、内存管理、结构体及文件等,知识层次清晰、讲解由浅入深,重点放在基础知识和考点的梳理、归纳和辨析上,并配有大量讲解详尽的例题。第13章是上机操作,总结和归纳了6大类35小类共93个算法题目,每个题目都有详细的解题思路和解题宝典。第2部分实训任务(14~21章)按照软件工程的开发流程,详细和完整地介绍一个实训任务的需求分析、概要设计、详细设计、调试、测试以及文档撰写各个环节的主要内容和关键问题,指导读者完成一个六七百行代码的实训课大作业。本书将实训任务的讲授重点放在软件设计上,采用结构化方法、自顶向下地将实训任务分解为若干个子模块,在迭代过程中逐步完善各模块功能,在代码编写上还引入软件企业采用的编码规范,以帮助学生提高所写软件程序的规范性。

建议C语言初学者在考级前按照本书中的开发流程认真完成实训任务,这将有助于深入理解C语言的基础知识,并大幅度提升编程能力。

全书贯彻计算思维能力训练,循序渐进地培养学生的计算思维能力,增强学生运用C语言基础知识和技能分析问题和解决问题的能力。本书第1章、第10章以及第14~21章由李志刚编写,第2~4章由贺娜娜编写,第5~6章由李海涵编写,第7章和第9章由邓玉洁编写,第8章、第11~12章由刘芳编写。本书第13章上机练习由陈义嘉、刘星宇、秦鑫洋、金吉安、王嘉恒、韩昱州和徐康博等在校本科生帮忙整理和汇总。全书由李志刚和刘芳统稿。本书提供实训任务CTraining代码资源,感兴趣的读者可以在北京邮电大学出版社官网(<http://www.buptpress.com>)上下载。

本书是全国高等院校计算机基础教育研究会立项项目成果,并得到北京邮电大学出版社的大力支持,在此一并表示衷心感谢!

由于作者水平有限,书中难免存在不足之处,恳请广大读者及同行给予批评和指正,谢谢!

编 者
2023 年 1 月

目 录

第 1 部分 C 语言考级教程

第 1 章 程序设计和计算思维	3
1.1 引言	3
1.1.1 第一个 C 语言程序例子	3
1.1.2 编程语言中的思维训练	4
1.1.3 非常受欢迎的编程语言	4
1.2 C 语言的特征	5
1.2.1 C 语言的编程语言特征	6
1.2.2 结构化程序设计	7
1.3 计算思维在 C 语言程序中的运用	9
1.3.1 C 语言编程学习中的计算思维	9
1.3.2 学习 C 语言的一些建议	10
第 2 章 C 语言概述	11
2.1 程序的组成结构	11
2.2 数据类型	12
2.3 常量、变量、标识符	13
2.3.1 常量和变量	13
2.3.2 标识符	14
第 3 章 运算符和表达式	17
3.1 C 语言运算符	17
3.1.1 运算符分类	17
3.1.2 运算符的结合性和优先级	18
3.2 算术运算符和算术表达式	18
3.2.1 基本的算术运算符	18
3.2.2 算术表达式的运算规则和要求	18
3.2.3 自增、自减运算符	19
3.2.4 算术运算中的类型转换	19
3.3 赋值运算符和赋值表达式	20

3.4 逗号运算符和逗号表达式	21
3.5 位运算符和位运算	22
第4章 输入输出函数	23
4.1 格式化输入输出函数	23
4.1.1 格式化输出函数	23
4.1.2 格式化输入函数	26
4.2 字符和字符串输入输出函数	28
4.2.1 字符输入、输出函数	28
4.2.2 字符串输入、输出函数	30
第5章 选择结构	31
5.1 关系运算符与关系表达式	31
5.1.1 关系运算符	31
5.1.2 关系表达式	31
5.2 逻辑运算符和逻辑表达式	32
5.2.1 逻辑运算符	32
5.2.2 逻辑表达式	33
5.2.3 逻辑运算的短路原则	34
5.3 条件运算	34
5.4 if 选择语句	35
5.4.1 if 语句的3种形式	35
5.4.2 使用if语句编程	37
5.5 多分支 switch 选择语句	39
5.5.1 switch 语句的形式	39
5.5.2 使用 switch 语句编程	40
5.6 选择结构的嵌套	41
第6章 循环结构	44
6.1 while 语句	44
6.2 do-while 语句	46
6.3 for 语句	47
6.4 循环嵌套	49
6.5 break 和 continue 语句	50
6.5.1 break 语句	50
6.5.2 continue 语句	51
第7章 数组	53
7.1 一维数组	53
7.1.1 一维数组的定义	53

7.1.2 一维数组的初始化	53
7.1.3 一维数组的引用	54
7.2 二维数组	58
7.2.1 二维数组的定义	58
7.2.2 二维数组的初始化	59
7.2.3 二维数组的引用	60
7.3 字符数组和字符串	63
7.3.1 字符数组	63
7.3.2 字符串结束标志	64
7.3.3 字符串的输出和输入	65
7.3.4 字符串处理函数	66
第8章 函数	70
8.1 函数定义和说明	70
8.2 函数的参数与返回值	71
8.2.1 函数的参数	71
8.2.2 函数的返回值	71
8.3 函数调用	72
8.4 递归调用	74
8.5 全局变量和局部变量	75
8.5.1 变量的作用域类别	75
8.5.2 变量的存储类别	76
第9章 指针	78
9.1 指针变量	78
9.1.1 指针变量的定义	78
9.1.2 指针变量的引用	79
9.2 指针与数组	81
9.3 指针与字符串	86
9.4 指向函数的指针	90
第10章 编译预处理和内存管理	93
10.1 编译预处理	93
10.1.1 宏定义命令	93
10.1.2 文件包含命令	95
10.1.3 条件编译命令	95
10.2 内存管理	96
10.2.1 变量的存储方式	96
10.2.2 动态内存分配	97

第 11 章 用户定义类型、结构体和共用体	99
11.1 用户定义类型 typedef	99
11.2 结构体的定义和使用	99
11.3 结构体指针	102
11.4 共用体	103
第 12 章 文件	105
12.1 文件的概念	105
12.2 文件的打开与关闭	106
12.2.1 文件的打开	106
12.2.2 文件的关闭	107
12.3 文件的读写	108
12.3.1 以字符形式读写文件	108
12.3.2 以字符串形式读写文件	109
12.3.3 以数据块形式读写文件	110
12.3.4 格式化读写文件	110
12.4 文件的定位	111
第 13 章 上机操作	113
13.1 有关数的运算	113
13.1.1 数据交换	113
13.1.2 数据整除	114
13.1.3 素数	116
13.1.4 阶乘	119
13.1.5 计算公式的值	120
13.1.6 数的分解与合并	124
13.2 一维数组类型	129
13.2.1 一维数组的平均值	129
13.2.2 一维数组的最大值和最小值	133
13.2.3 数组的排序	135
13.2.4 在数组中查找数据	136
13.2.5 数组元素的删除和移动	137
13.3 字符串类型	141
13.3.1 字母字符的判定与统计	141
13.3.2 字母大小写的转换	144
13.3.3 数字字符的判定与统计	145
13.3.4 数字字符转换为对应数值	147
13.3.5 字符串中字符的删除	149
13.3.6 字符串中字符的移动	155

13.3.7 字符串中查找子串·····	158
13.3.8 字符串的回文判断与逆置·····	161
13.3.9 字符串的复制·····	163
13.3.10 字符串数组·····	166
13.4 二维数组-矩阵·····	173
13.4.1 矩阵行列的最大值和最小值·····	173
13.4.2 矩阵周边元素下标的特点·····	174
13.4.3 二维数组的行列·····	176
13.4.4 矩阵对角线元素的特点·····	179
13.4.5 矩阵转置·····	183
13.5 结构体类型·····	184
13.5.1 结构体成员的访问·····	184
13.5.2 结构体数组的平均值·····	185
13.5.3 结构体数组的最大值和最小值·····	187
13.5.4 结构体数组元素的查找·····	189
13.5.5 结构体数组元素的排序·····	191
13.6 链表类型·····	195
13.6.1 链表的顺序访问·····	195
13.6.2 链表的插入·····	202
13.6.3 链表的删除·····	204
13.6.4 链表的排序·····	205

第 2 部分 C 语言实训任务

第 14 章 软件工程基础·····	209
14.1 软件·····	209
14.2 软件生命周期·····	209
14.3 结构化程序设计·····	210
14.4 面向对象的程序设计·····	212
14.5 软件测试·····	212
14.6 软件调试·····	213
第 15 章 编码规范·····	214
15.1 程序文件规范·····	214
15.2 模块和变量规范·····	216
第 16 章 实训任务内容和要求·····	218
16.1 实训任务安排和要求·····	218
16.2 实训任务题目需求·····	218

16.3 实训任务示例需求	220
第 17 章 实训任务需求分析	221
17.1 可行性分析	221
17.2 需求分析	221
第 18 章 实训任务概要设计	223
18.1 软件结构设计	223
18.2 数据设计	225
18.3 接口设计和过程设计	225
第 19 章 实训任务详细设计	228
19.1 主模块 main() 的详细设计	228
19.2 添加和显示记录模块的详细设计	233
19.3 删除和修改记录模块的详细设计	245
19.4 查询、插入和排序记录模块的详细设计	253
19.5 C 语言代码的调试	261
第 20 章 实训任务测试和文档	264
20.1 实训任务测试	264
20.2 文档	266
第 21 章 实训任务总结和改进方案	269
21.1 实训任务总结	269
21.2 进一步改进的方法	269
21.3 其他实现方案	273
参考文献	274
附录 1 全国计算机等级考试(二级 C 语言)程序设计考试大纲(2022 年版)	275
附录 2 全国计算机等级考试二级公共基础知识考试大纲(2022 年版)	277

第1部分

C语言考级教程 ▼

第1部分C语言考级教程分为13章，覆盖了全国计算机等级考试（二级C语言）除公共知识以外的所有知识点，其中第1章介绍程序设计和计算思维，第2章至第12章介绍C语言基础知识，包括了C语言概述、运算符和表达式、输入输出函数、选择结构、循环结构、数组、函数、指针、编译预处理和内存管理、用户定义类型、结构体、共用体以及文件等，第13章介绍上机操作。

第 1 章 程序设计和计算思维

目前 C 语言是全世界主要的计算机程序设计语言之一,C 语言程序设计已成为各高校计算机基础教学的核心课程。通过 C 语言的学习,学生不仅能够掌握计算机程序设计的基本概念、思想、方法以及编程技能,还能够培养和锻炼计算思维,为解决复杂软件工程问题打下良好基础。

本章要点

- ① 编程语言对思维训练的意义;
- ② 结构化程序设计的基本概念和观点;
- ③ C 语言程序设计中蕴含的计算思维方法。

1.1 引言

以一个 C 语言程序为例,阐述程序设计中蕴含的思维方法,引入计算思维概念,然后说明 C 语言在软件开发中的重要位置。

1.1.1 第一个 C 语言程序例子

在 C 语言程序设计教材中,通常第一个 C 语言程序例子是“Hello World!”程序,其典型的代码如下:

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    printf("Hello World!\n");
    return 0;
}
```

在 Windows 操作系统中,这是一个 C 语言控制台程序,其经过编译、链接之后会生成可执行文件,运行该可执行文件后将在控制台上输出字符串“Hello World!”。

上面简单几行代码在逻辑上蕴含了人们执行很多任务所遵循的基本思维方法。

① 借助于外部资源。为了高效地完成任务,人们往往借助于外部资源,充分利用外部资源可以做到事半功倍。上述程序中,引入 C 语言的库函数 `#include <stdio.h>` 就可以不必了解控制台的技术细节,而直接向控制台输出指定的字符串。

② 执行并完成任务。上述程序中,main 后面的“{ }”中代码的主要功能是使用“`printf("Hello World! \n")`”语句实现字符串在控制台的显示,这可以理解为做事的“内因”,而整个程序功能则是“外因通过内因”而实现的。

③ 反馈结果。在完成任务后,应该向“上级”反馈“任务已经完成”,在程序中,通过“return 0”语句通知 Windows 的控制台——本程序已经成功将字符串输出到控制台,并正常退出。

上面程序逻辑中涉及的外因、内因、反馈概念等,在计算思维中表现为“评估”“设计”和“通信”等基本方法。

1.1.2 编程语言中的思维训练

利用编程语言编写一段程序代码,不但表现为一种技术层面的开发过程,而且表现为一种计算思维活动过程。

学习计算机编程语言时,除了学习编程语言本身外,还应该学习什么呢? 微软公司创始人比尔·盖茨说:“你不必是编程或元素周期表方面的专家,但是具备这些专家思考方式的能力,则将会为你提供极大的帮助。你不一定会编写代码,但是你需要了解计算机工程师可以做什么以及不能做什么。”

那么,到底什么是计算思维? 通俗地说,计算思维就是计算机工程师面对问题时如何思考,以及找出问题的交互关系,并建立永久性解决方案的方法和步骤。例如,工程师在解决问题时有特定的思考流程,面对一个问题时首先将问题拆解成许多的小问题(拆解问题);其次找出问题彼此间的关联性或规律性(找到模式);再次将问题简化,忽略细节(建立抽象化);最后针对这个问题提供一个完整的解决方案(完成演算)。

计算思维在人类认识世界和改造世界过程中具有极其重要的地位,业界学者将其与实证思维、逻辑思维并列到同等地位,并归纳出人类认识世界和改造世界过程中的 3 种基本思维特征。

① 实证(实验)思维:以观察和总结自然规律为特征,以物理学科为代表。

② 逻辑(推理)思维:以推理和演绎为特征,以数学学科为代表。

③ 计算思维:运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为,是人类求解问题的一条途径。

计算思维的本质是抽象和自动化,它虽然具有计算机科学的许多特征,但是计算思维本身并不是计算机科学的专属。实际上,即使没有计算机,计算思维也会逐步发展。但正是计算机的出现,才给计算思维的研究和发展带来了根本性的变化,并且计算思维可通过对计算机科学基本知识和应用能力的学习而得以理解和掌握。通过编程学习计算思维是一条非常好的途径,但这要求我们至少要学习和掌握一门编程语言。

1.1.3 非常受欢迎的编程语言

C 语言是一种结构化程序设计语言,功能强大,具有高效性、快速性、灵活性和可移植性等优点,这使其自 1972 年诞生以来就一直是软件工程开发的主流编程语言,也是程序设计的初学者普遍学习的编程语言。

在最受欢迎编程语言排行榜——TIOBE 编程语言排行榜中,C 语言位居排行榜前列,图 1.1 是 TIOBE 于 2022 年 7 月公布的最受欢迎编程语言排行榜。

Jul 2022	Jul 2021	Change	Programming Language	Ratings	Change
1	3	^	 Python	13.44%	+2.48%
2	1	v	 C	13.13%	+1.50%
3	2	v	 Java	11.59%	+0.40%
4	4		 C++	10.00%	+1.98%
5	5		 C#	5.65%	+0.82%
6	6		 Visual Basic	4.97%	+0.47%
7	7		 JavaScript	1.78%	-0.93%
8	9	^	 Assembly language	1.65%	-0.76%
9	10	^	 SQL	1.64%	+0.11%
10	16	^	 Swift	1.27%	+0.20%

图 1.1 TIOBE^①于 2022 年 7 月公布的最受欢迎编程语言排行榜

虽然在过去几十年里新的编程语言如雨后春笋般产生,但是 C 语言几乎始终占据着最受欢迎编程语言排行榜前三的位置,图 1.2 所示为 1985—2020 年最受欢迎编程语言排行的变化情况,时至今日 C 语言仍然是用来搭建软件世界的基础建筑材料。历经数十年的研究和开发,C 语言的地位依旧稳固,很少有其他语言能够在性能、裸机兼容性或通用性等方面击败它。

Programming Language	2020	2015	2010	2005	2000	1995	1990	1985
C	1	2	2	1	1	2	1	1
Java	2	1	1	2	3	29	-	-
Python	3	6	6	7	23	9	-	-
C++	4	3	3	3	2	1	2	9
C#	5	4	5	6	10	-	-	-
JavaScript	6	8	10	10	7	-	-	-
PHP	7	7	4	4	19	-	-	-
SQL	8	-	-	-	-	-	-	-
R	9	14	46	-	-	-	-	-
Swift	10	16	-	-	-	-	-	-
Lisp	30	26	15	13	8	10	6	2
Fortran	31	23	24	14	15	18	3	5
Ada	33	27	21	16	17	4	9	3
Pascal	242	15	14	17	16	3	7	6

图 1.2 最受欢迎编程语言排行的变化情况

1.2 C 语言的特征

本节首先从软件开发角度归纳出一些 C 语言的编程语言特征,以加深初学者对它的理

^① TIOBE 编程社区指数是衡量编程语言流行度的指标。该指数每月更新一次,是基于谷歌、必应等 25 种搜索引擎上与编程相关的搜索查询得到的结果。TIOBE 编程社区指数旨在反映编程语言受欢迎程度的变化。

解,然后介绍结构化方法的特点以及算法的初步知识。

1.2.1 C 语言的编程语言特征

为更清晰地描述 C 语言特征,首先回顾 C 语言程序的编译和链接过程,然后,从强弱类型、动静态类型两个维度对编程语言进行分类。

1. C 语言的编译和链接过程

C 语言是一种编译型语言,从编写 C 语言程序开始到最终得到运行结果,一般需要经历编辑、编译、链接和运行 4 个主要操作阶段,这就是我们通常说的程序“开发”,如图 1.3 所示,而这 4 个操作阶段中将分别得到 3 种不同类型的文件。以 Windows 操作系统下 C 语言程序开发为例,编辑阶段之后得到的文件称为源文件(即源代码文件、源程序),扩展名为 .c,编译成功后得到的文件称为目标文件,扩展名为 .obj,进一步链接成功后得到的文件称为可执行文件,扩展名为 .exe。注意在 Windows 操作系统中文件名不区分大小写,这与 C 语言本身语法没有关系。

在 C 语言程序的基本操作过程中,每个操作阶段只要出现错误一般都需要回到最初源文件编辑阶段进行修改,直至得到希望的运行结果,其中修改错误环节一般称为程序的调试。

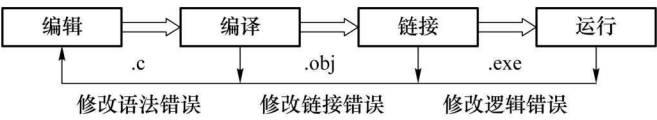


图 1.3 C 语言程序的基本操作过程

对 C 语言的初学者来说,使用最多的 C 语言的开发工具是微软公司的集成开发环境 (IDE),如 Microsoft Visual C++6.0、Microsoft Visual Studio 2010、Microsoft Visual Studio 2019 等版本,它们将上面 4 个操作阶段集成在一个开发环境中,这种集成开发环境为程序开发带来了方便,开发者在一个环境中就可以进行代码的编辑、编译、链接和运行。但这 4 个操作阶段实际上也可以单独完成,例如,使用编辑器 Notepad++或 Notepad2 进行源代码编写,使用编译和链接工具进行编译和链接操作,使用命令行在 Windows 控制台下运行程序等。

2. C 语言的主要特征

从图 1.1 可以看出,编程语言的类型非常多,除了 C 语言之外,还有 C++、C#、Java、Python、Perl、PHP 等,业界将编程语言从强弱类型、动静态类型两个维度划分为 4 种类型,即静态类型(static)和动态类型(dynamic)、强类型(strong)和弱类型(weak)。由于各个类型的严格规定需要引入更多知识,因此本书不给出严格规定,而将重点放在 C 语言特征的阐述上。图 1.4 所示是比较公认的编程语言类型划分方式。

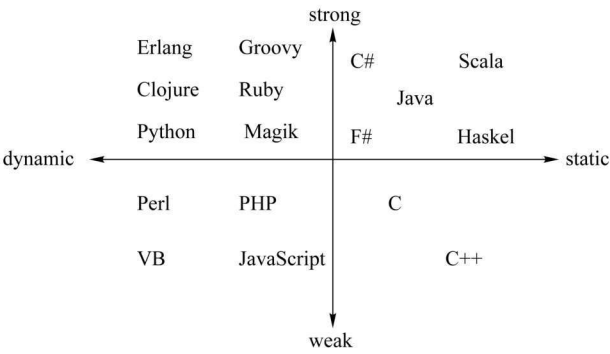


图 1.4 编程语言的类型

从图 1.4 中可以看出,C 语言属于静态类型,且处于强、弱类型之间,即它是静态类型语言,并同时具备强、弱类型的特点。

① C 语言是静态类型语言。C 语言源程序在编译时需要做数据类型检查,也就是要求源程序中声明所有变量的数据类型,如 int、float、double 等,并且数据类型一旦声明之后不可改变,否则会导致编译错误。相比之下,动态类型编程语言在源代码中不必给变量指定数据类型,而在程序运行期间才做数据类型的检查,如 Python 语言。

② C 语言兼具强、弱类型特点。一般而言,强类型语言是指需要强制类型定义的语言,即一个变量数据类型确定后,如果转换数据类型,则需要在程序中显示地使用强制类型转换规则来完成转换。例如,在 C 语言中,

```
float a = 3.5;
int b = (int)a;    //将 a 强制转换成 int 类型并赋值给整型变量 b
```

弱类型语言指变量类型可以做隐式的类型转换,又称为自动类型转换。例如在 C 语言中有多种数据类型混合计算的时候,若未主动转换数据类型,则系统会自动进行类型转换。自动类型转换规则是:将存储长度较短的数据类型转换成存储长度较长的数据类型,如图 1.5 所示。

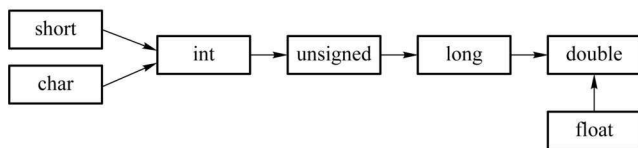


图 1.5 C 语言自动类型转换

根据以上对 C 语言特征的分析,在编写 C 语言源程序时应注意如下两个问题。

① 使用的变量类型必须声明,否则在编译时会报错。

② C 语言数据类型转换分为强制转换和自动转换,由于编译器不检测自动类型转换后的数据,所以需要编程者掌握类型自动转换规则,并判断转换后是否会产生计算错误。

1.2.2 结构化程序设计

C 语言是一门较难学习和使用的编程语言,但是学习 C 语言是学好其他编程语言的基础,也是培养和训练计算思维能力的有效途径。

C 语言是一种结构化程序设计语言,采用面向过程的程序设计(Process Oriented Programming, POP)方法。其以程序的可读性和清晰性为目标,采用结构化和模块化(函数)设计思想,易于维护和重用代码,易于通过自顶向下和逐步求精的方式解决各种复杂问题。即使当前流行的面向对象的程序设计中,当具体到一个功能模块(方法)时,也离不开结构化程序设计的思想。

1.3 种程序控制结构

结构化程序有 3 种基本结构,即顺序结构、选择结构和循环结构,这 3 种结构也是各种编程语言都具备的程序控制结构。

(1) 顺序结构(sequence structure)

顺序结构是最简单、最基本的程序结构。在这种结构中,程序的各块是按其书写顺序依次执行的。