

全国计算机技术与软件专业技术资格(水平)考试同步辅导丛书

程序员考试同步辅导 (程序设计篇)

重点·难点突破

全国计算机技术与软件专业技术资格水平考试命题研究组 编审
何光明 丛书主编 杨明 杨萍 本书主编

重点难点考点剖析透彻
例题习题考题一书网罗
试题分析贯穿考点讲解
同步辅导结合实战训练
权威专家明示命题规律

清华大学出版社

全国计算机技术与软件专业技术资格(水平)考试同步辅导丛书

程序员考试同步辅导(程序设计篇)

——重点·难点突破

全国计算机技术与软件专业技术资格(水平)考试命题研究组 编审

何光明 丛书主编

杨 明 杨 萍 本书主编

清华大学出版社

北 京

内 容 简 介

本书按照人事部、信息产业部最新颁布的全国计算机技术与软件专业技术资格(水平)考试大纲和指定教材编写。全书共分为5章,内容包括程序语言知识、C语言、数据结构、常用算法设计方法和 Visual Basic 程序设计。主要从考试大纲要求、考点辅导、典型例题分析和专项习题训练几个方面加以系统的阐释。另外,附录中附有最新考试大纲和6套程序员考试下午样卷及参考答案,以及最新2004年上半年程序员考试下午试题及参考答案。

本书具有考点分析透彻、例题典型、习题丰富等特点,非常适合考生使用,也可作为高等院校或培训班的教材。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933
本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

程序员考试同步辅导(程序设计篇)——重点·难点突破/全国计算机技术与软件专业技术资格(水平)考试命题研究组编;何光明丛书主编;杨明,杨萍本书主编.—北京:清华大学出版社,2004.10

ISBN 7-302-09597-3

(全国计算机技术与软件专业技术资格(水平)考试同步辅导丛书)

I.程… II.①全…②何…③杨…④杨… III.程序设计-水平考试-自学参考资料 IV.TP311.1

中国版本图书馆CIP数据核字(2004)第096950号

出版者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社总机:010-62770175 客户服务:010-62776969

组稿编辑:章忆文

文稿编辑:刘颖

封面设计:陈刘源

印装者:北京昌平环球印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:28.75 字数:678千字

版 次:2004年10月第1版 2004年10月第1次印刷

书 号:ISBN 7-302-09597-3/TP·6658

印 数:1~5000

定 价:38.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

丛书序

计算机技术与软件专业技术资格(水平)考试(简称计算机与软件考试)是原中国计算机专业技术资格和水平考试(简称软件考试)的完善与发展。这是由国家人事部和信息产业部领导下的国家级考试,其目的是科学、公正地对全国计算机与软件专业技术人员进行职业资格、专业技术资格认定和专业技术水平测试。

原软件考试在全国范围内已经实施了十多年,到2003年底,累计参加考试的人数约有一百万人。该考试由于其权威性和严肃性,得到了社会及用人单位的广泛认同,并为推动我国信息产业特别是软件产业的发展和提高各类IT人才的素质做出了积极的贡献。

为了更好地贯彻落实国办47号文《振兴软件产业行动纲要》的精神,规范计算机技术与软件专业人才评价工作,促进计算机技术与软件专业人才队伍建设,人事部、信息产业部在总结计算机软件专业资格和水平考试实施情况的基础上,重新修订了计算机软件专业资格和水平考试有关规定(国人部发[2003]39号),颁布了新的考试大纲。新的考试政策进行了重大的改革:考试名称调整为计算机技术与软件专业技术资格(水平)考试;考试对象从狭义的计算机软件拓展到计算机技术与软件专业的各个主要领域(5个专业类别、3个级别层次和20个职业岗位资格);资格考试和水平考试合并,采用水平考试的形式(与国际接轨,报考不限学历与资历条件),执行资格考试的政策(各用人单位可以从考试合格者中择优聘任专业技术职务),在全国率先执行了以考代评的政策。

配合全国计算机技术与软件专业技术资格(水平)考试新大纲的推出,我们组织一批具有较高理论水平和丰富实践经验的专家教授编写了《全国计算机技术与软件专业技术资格(水平)考试同步辅导丛书》。该丛书按照最新考试大纲和最新教程的要求,全面介绍相关知识与技术,帮助考生学习和备考。

首推书目

本套丛书首推10本,书目如下:

- (1) 程序员考试同步辅导(计算机软硬件基础知识篇)
- (2) 程序员考试同步辅导(程序设计篇)
- (3) 软件设计师考试同步辅导(计算机与软件工程知识篇)
- (4) 软件设计师考试同步辅导(软件设计篇)
- (5) 网络管理员考试同步辅导(计算机与网络基础知识篇)

- (6) 网络管理员考试同步辅导(网络系统管理与维护篇)
- (7) 网络工程师考试同步辅导(计算机与网络知识篇)
- (8) 网络工程师考试同步辅导(网络系统设计与管理篇)
- (9) 数据库系统工程师考试同步辅导(信息系统知识篇)
- (10) 数据库系统工程师考试同步辅导(数据库系统设计与管理篇)

丛书特色

(1) 编写的目的是引导考生花较少的时间顺利通过全国计算机技术与软件专业技术资格(水平)考试,编写主线以“考点讲解、试题分析”贯穿,以“辅导与训练并重,习题与分析结合”的原则进行。

(2) 考虑到全国计算机技术与软件专业技术资格(水平)考试内容多、范围广,丛书将同一门考试内容按学科合理分类,划分为2本分册,这样便于把考点讲透,以区别以前市面上用一本书通包所有内容,但却均是泛泛而谈的不足。

(3) 作为复习迎考用书,本丛书内容上不求完整性、系统性,而是将“针对性”放在重中之重,即将考点、重点、难点不惜笔墨地分析透彻,把与考试无关的内容大刀阔斧地略去。

(4) 丛书章名、节名与信息产业部最新指定教程同步,每一小节包括4个板块:考点辅导、典型例题分析、同步练习、同步练习答案。各板块内容安排为:

- 考点辅导:突出常考知识与核心知识,对考点、重点、难点内容进行解释与讲述,让考生掌握问题的本质。
- 典型例题分析:精选出常考题型与考试真题进行解析,增强考生解题能力。
- 同步练习:设计一部分习题,供考生即学即练之用。
- 同步练习答案:给出习题的答案及分析,便于考生复习与检查。

(5) 例题选取精心,分析到位。书中的例题一部分选自近年全国计算机技术与软件专业技术资格(水平)考试的真题,一部分是根据最新考试要求精心设计而成,具有典型性和针对性。所有例题均给出了详尽的分析,便于考生掌握完整的解题思路,以达举一反三、触类旁通之功效。

(6) 实战练习丰富,附有答案。本书针对考试过关,着重实战,每个章节均配有练习题,这些练习题是对逐段所学内容的巩固与提高,最后还提供了几套样卷,便于读者检测自己的总体水平。所有练习题、模拟题均配有答案(及分析),便于自测使用。

关于作者

丛书聘请具有较高理论水平和丰富实践经验的专家编写。他们长期从事全国计算机技术与软件专业技术资格(水平)考试的教学、培训、阅卷和命题研究工作,积累了丰富的经验,对命题规律和出题方向有较深的体会与独到的见解。

读者定位

本套丛书特别适合参加全国计算机技术与软件专业技术资格(水平)考试的读者群阅读,同时可供高等院校或培训班作教材使用。

互动交流

读者的进步,我们的心愿。如果发现书中有任何疑惑之处,或有建议或意见,请与我们交流。联系信箱: gmkeji@163.com。

特别致谢

在此,对丛书所选用的参考文献的著作者,及丛书所引用习题、试题的命题老师表示真诚的感谢。感谢为本丛书出版提供帮助的各界人士。

全国计算机技术与软件专业技术资格(水平)考试命题研究组

前 言

全国计算机技术与软件专业技术资格(水平)考试实施已经历了十多年,其权威性得到社会各界的广泛认可。为了适应当前信息技术的飞速发展,为了更好地服务于考生,本书以 2004 年新版程序员考试大纲为依据,严格按照全国计算机技术与软件专业技术资格(水平)考试指定用书——《程序员教程》的结构编排,兼顾计算机技术发展和知识更新,细化各章节的基础知识要点,配以真题与典型例题并加以详细剖析编写而成。

2004 版新大纲为考生提供了多种语言选择,除了保留传统的结构程序设计语言 C 外,还增加了 Visual Basic 可视化程序设计语言。应试考生既可以选择传统的 C 语言进行相应题目进行答题,也可以选择 Visual Basic 可视化程序设计语言进行答题。特别指出的是,Visual Basic 可视化程序设计语言进入程序员考试刚刚开始,相应的题目难度不是太大,对于考生来说,选择 Visual Basic 可视化程序设计的题目可能是不错的机遇。本书章名、节名与信息产业部最新指定教程同步,每一小节分 4 个板块:考点辅导、典型例题分析、同步练习、同步练习答案。其中,考点辅导部分主要以专题的方式,重点介绍程序员下午考试所需的各个方面的知识,典型例题分析是本书的重点,书中的例题一部分是历年程序员考试真题、初级程序员中的典型试题(包括所有 Visual Basic 试题)、部分高级程序员上午试题,一部分是根据最新考试大纲精心设计而成,具有典型性和代表性。所有例题均给出了详尽的分析。每章均配有数量一定的习题及答案,对读者所学的知识 and 能力起到巩固、拓宽和提高的作用。

全书共 5 章,内容主要包括:第 1 章程序语言知识,第 2 章 C 语言,第 3 章数据结构,第 4 章常用算法设计方法,第 5 章 Visual Basic 程序设计,附录部分提供了 6 套程序员考试下午样卷及参考答案,供考生自测。其中,算法的重点内容为第 3 章,第 2 和 5 章是实现算法的手段,当然第 5 章有些独立,因为其可视化的控件掌握是其中的部分考试内容。

本书由杨明、杨萍主编。另外,参与本书资料整理工作的还有何光明、董伯英、吴子勤、张勇、庞铮、郁军、吴寅、陈宏、樊芳芳、徐军、何群、许勇、孙玉香、杨茂龙、毛红梅、史春联、周松、刘菁、刘瀚、吴婷、谢波等。

在本书编写的过程中,参考了许多相关的书籍和资料,编者在此对这些参考文献的作者表示感谢。

由于时间仓促和水平有限,书中难免存在错漏和不妥之处,敬请读者批评指正。

编 者
2004 年 9 月

目 录

第 1 章 程序语言知识 1	
1.1 程序设计语言基础知识..... 1	
1.1.1 考点辅导..... 1	
1.1.2 典型例题分析..... 5	
1.1.3 同步练习..... 8	
1.1.4 同步练习答案..... 8	
1.2 结构化程序设计基础知识..... 9	
1.2.1 考点辅导..... 9	
1.2.2 典型例题分析..... 10	
1.2.3 同步练习..... 13	
1.2.4 同步练习答案..... 13	
1.3 面向对象程序设计基础知识..... 14	
1.3.1 考点辅导..... 14	
1.3.2 典型例题分析..... 17	
1.3.3 同步练习..... 19	
1.3.4 同步练习答案..... 19	
1.4 语言处理程序基础知识..... 20	
1.4.1 考点辅导..... 20	
1.4.2 典型例题分析..... 25	
1.4.3 同步练习..... 32	
1.4.4 同步练习答案..... 33	
1.5 本章小结..... 33	
1.6 达标训练题及参考答案..... 33	
1.6.1 达标训练题..... 33	
1.6.2 参考答案..... 35	
第 2 章 C 语言 37	
2.1 C 语言简介..... 37	
2.1.1 考点辅导..... 37	
2.1.2 典型例题分析..... 39	
2.1.3 同步练习..... 39	
2.1.4 同步练习答案..... 40	
2.2 C 语言的数据类型、运算符和表达式..... 40	
2.2.1 考点辅导..... 40	
2.2.2 典型例题分析..... 42	
2.2.3 同步练习..... 42	
2.2.4 同步练习答案..... 43	
2.3 C 语言的基本语句..... 44	
2.3.1 考点辅导..... 44	
2.3.2 典型例题分析..... 44	
2.3.3 同步练习..... 46	
2.3.4 同步练习答案..... 47	
2.4 标准输入输出函数..... 48	
2.4.1 考点辅导..... 48	
2.4.2 典型例题分析..... 50	
2.4.3 同步练习..... 50	
2.4.4 同步练习答案..... 51	
2.5 数组和函数..... 51	
2.5.1 考点辅导..... 51	
2.5.2 典型例题分析..... 56	
2.5.3 同步练习..... 59	
2.5.4 同步练习答案..... 64	
2.6 指针..... 66	
2.6.1 考点辅导..... 66	
2.6.2 典型例题分析..... 70	
2.6.3 同步练习..... 73	
2.6.4 同步练习答案..... 73	
2.7 结构和联合..... 74	
2.7.1 考点辅导..... 74	
2.7.2 典型例题分析..... 76	
2.7.3 同步练习..... 78	
2.7.4 同步练习答案..... 80	
2.8 预定义..... 80	

2.8.1	考点辅导	80	3.4.4	同步练习答案	155
2.8.2	典型例题分析	81	3.5	数组	156
2.8.3	同步练习	81	3.5.1	考点辅导	156
2.8.4	同步练习答案	82	3.5.2	典型例题分析	160
2.9	位运算	82	3.5.3	同步练习	176
2.9.1	考点辅导	82	3.5.4	同步练习答案	177
2.9.2	典型例题分析	82	3.6	串	178
2.9.3	同步练习	84	3.6.1	考点辅导	178
2.9.4	同步练习答案	84	3.6.2	典型例题分析	180
2.10	文件	84	3.6.3	同步练习	184
2.10.1	考点辅导	84	3.6.4	同步练习答案	187
2.10.2	典型例题分析	88	3.7	树和二叉树	189
2.10.3	同步练习	96	3.7.1	考点辅导	189
2.10.4	同步练习答案	99	3.7.2	典型例题分析	202
2.11	本章小结	100	3.7.3	同步练习	213
2.12	达标训练题及参考答案	100	3.7.4	同步练习答案	216
2.12.1	达标训练题	100	3.8	排序	218
2.12.2	参考答案	105	3.8.1	考点辅导	218
第3章	数据结构	108	3.8.2	典型例题分析	227
3.1	数据结构简介	108	3.8.3	同步练习	237
3.1.1	考点辅导	108	3.8.4	同步练习答案	240
3.1.2	典型例题分析	109	3.9	查找	242
3.1.3	同步练习	109	3.9.1	考点辅导	242
3.1.4	同步练习答案	109	3.9.2	典型例题分析	246
3.2	线性表	110	3.9.3	同步练习	248
3.2.1	考点辅导	110	3.9.4	同步练习答案	249
3.2.2	典型例题分析	117	3.10	图	250
3.2.3	同步练习	135	3.10.1	考点辅导	250
3.2.4	同步练习答案	136	3.10.2	典型例题分析	254
3.3	栈	136	3.10.3	同步练习	261
3.3.1	考点辅导	136	3.10.4	同步练习答案	263
3.3.2	典型例题分析	142	3.11	本章小结	263
3.3.3	同步练习	146	3.12	达标训练题及参考答案	264
3.3.4	同步练习答案	146	3.12.1	达标训练题	264
3.4	队列	147	3.12.2	参考答案	269
3.4.1	考点辅导	147	第4章	常用算法设计方法	272
3.4.2	典型例题分析	152	4.1	迭代法	272
3.4.3	同步练习	153	4.1.1	考点辅导	272

4.1.2 典型例题分析	272	4.9 本章小结	307
4.1.3 同步练习	273	4.10 达标训练题及参考答案	308
4.1.4 同步练习答案	274	4.10.1 达标训练题	308
4.2 穷举搜索法	274	4.10.2 参考答案	309
4.2.1 考点辅导	274	第 5 章 Visual Basic 程序设计	311
4.2.2 典型例题分析	274	5.1 概述	311
4.2.3 同步练习	278	5.1.1 考点辅导	311
4.2.4 同步练习答案	279	5.1.2 典型例题分析	314
4.3 递归法	279	5.1.3 同步练习	315
4.3.1 考点辅导	279	5.1.4 同步练习答案	315
4.3.2 典型例题分析	280	5.2 用户界面设计	316
4.3.3 同步练习	284	5.2.1 考点辅导	316
4.3.4 同步练习答案	285	5.2.2 典型例题分析	325
4.4 递推法	286	5.2.3 同步练习	330
4.4.1 考点辅导	286	5.2.4 同步练习答案	332
4.4.2 典型例题分析	286	5.3 Visual Basic 语言基础	332
4.4.3 同步练习	287	5.3.1 考点辅导	332
4.4.4 同步练习答案	288	5.3.2 典型例题分析	345
4.5 回溯法	289	5.3.3 同步练习	357
4.5.1 考点辅导	289	5.3.4 同步练习答案	366
4.5.2 典型例题分析	289	5.4 应用程序中的文件处理	369
4.5.3 同步练习	295	5.4.1 考点辅导	369
4.5.4 同步练习答案	296	5.4.2 典型例题分析	373
4.6 贪婪法	297	5.4.3 同步练习	376
4.6.1 考点辅导	297	5.4.4 同步练习答案	376
4.6.2 典型例题分析	297	5.5 访问数据库	377
4.6.3 同步练习	298	5.5.1 考点辅导	377
4.6.4 同步练习答案	300	5.5.2 典型例题分析	378
4.7 分治法	300	5.5.3 同步练习	379
4.7.1 考点辅导	300	5.5.4 同步练习答案	379
4.7.2 典型例题分析	300	5.6 本章小结	379
4.7.3 同步练习	303	5.7 达标训练题及参考答案	379
4.7.4 同步练习答案	304	5.7.1 达标训练题	379
4.8 动态规划法	305	5.7.2 参考答案	385
4.8.1 考点辅导	305	附录 A 程序员级考试大纲(2004	
4.8.2 典型例题分析	305	年新版)	388
4.8.3 同步练习	307		
4.8.4 同步练习答案	307		

附录 B 程序员考试下午样卷 一及参考答案.....	394	附录 F 程序员考试下午样卷 五及参考答案.....	419
附录 C 程序员考试下午样卷 二及参考答案.....	401	附录 G 程序员考试下午样卷 六及参考答案.....	425
附录 D 程序员考试下午样卷 三及参考答案.....	408	附录 H 2004 年上半年程序员考试 下午试题及参考答案.....	432
附录 E 程序员考试下午样卷 四及参考答案.....	414	参考文献.....	443

第 1 章 程序语言知识

大纲要求:

- 程序设计语言的基本成分(数据、运算、控制和传输)。
- 汇编、编译、解释系统的基础知识。
- 过程(函数)调用。

1.1 程序设计语言基础知识

1.1.1 考点辅导

1. 语言的简介

语言可以分为两大类:自然语言和人工语言。其中,人类所使用的语言一般被称为自然语言,自然语言是人类在自身发展过程中,为了表达思想、交流经验和互通信息而自然形成的语言。但自然语言一般不被计算机识别,必须通过某种途径生成计算机所能够操作的代码。于是,出现了人工语言,人工语言是人们为了达到某种目的而专门设计的语言。

计算机程序设计语言是用来编写程序的语言,是软件系统的重要组成部分,与程序设计语言相对应的各种语言处理程序则为该语言提供支持和辅助作用。程序设计语言一般分为机器语言、汇编语言和高级语言三大类。

2. 机器语言的简介

计算机基本的物理构成是复杂的电子线路。这些线路可以在不同的电信号作用下,通过各种装置产生不同的动作。计算机的核心部分(CPU、存储器等)只能够识别高、低电平两种状态,即二进制的 0 和 1。由 0 和 1 的不同组合所形成的可以为计算机直接识别的二进制指令代码的集合就称为机器语言,而在指令代码中一般包括两部分:操作码和地址码。操作码指出操作的性质,即要求机器做何种操作;地址码则指出被操作的对象所在的位置。机器语言是一种能够被计算机的硬件直接识别和执行的语言。计算机在接受到这些二进制代码以后,由译码器产生各种控制信号送往有关的单元,从而产生相应的操作。

机器语言是最基本的、出现最早的计算机编程语言,是唯一可以为计算机直接执行的语言。用机器语言编写的程序小,执行效率高,占用内存空间小,运行速度快,可以直接控制计算机的硬件。但是不同类型计算机的硬件结构可能有所不同,其各自具有不同的指令系统,因此用机器语言编程对程序设计者的水平要求很高,他们必须对所使用的计算机的硬件工作原理及线路连接关系十分清楚,而且一般在某种机器上编写的程序不能在另一种机器上运行,因此人们称机器语言为“面向机器”的程序设计语言。用机器语言编写的程序可读性差,可移植性差,容易出错,出错后很难查找,因此,人们往往不直接使用机

器语言进行编程,而是借助于其他更为简单、编程效率更高的计算机语言,通过间接的方法来产生可以为计算机直接接受的代码。

3. 汇编语言

为了克服机器语言的缺点,便于一般人员进行编程,在 20 世纪 50 年代初出现了汇编语言。其特点是引入了一些助记符来表示操作,用一些特定的符号来表示计算机中的某些单元,例如用“ADD”表示加运算,用“AL”,“AH”表示某一寄存器等。这些助记符比较接近英文,其他助记符也比较简单,便于记忆、理解,从而简化了编程过程。由于汇编语言比机器语言更直观,所以使用汇编语言编写的程序也比用机器语言编写的程序更容易读和修改。

使用助记符和有关符号编写的程序被称为汇编语言程序。由于计算机只能识别二进制代码,不能够识别这些符号,因此还必须通过某种方法将汇编语言程序“翻译”成相应的二进制代码。由这些二进制代码组成的程序称为目标程序,“翻译”过程称为汇编。汇编的方法有人工汇编和机器汇编两种,人工汇编是由人根据机器的指令系统通过查表进行转换,机器汇编则是借助于专门的汇编程序由计算机来完成转换。

汇编语言同机器语言一样,都是面向某一种 CPU 的语言,使用汇编语言编写出的应用程序对于计算机的 CPU 及 CPU 的外围硬件设备具有很大的依赖性。这就要求编程人员具有相关的计算机硬件知识,因为汇编语言仍然属于一种“面向机器”的程序设计语言,编程过程离不开机器的硬件结构,助记符、符号及汇编过程都依赖机器的特定指令系统。所以,汇编语言程序的通用性仍然较差,应用范围受到一定的限制。但是汇编语言程序仍然具有运行速度快、能够直接控制硬件等特点,在实时性要求较高的计算机自动控制系统中或者对计算机硬件设备要求特殊管理、编写系统软件的情况下,仍然得到了广泛的应用。

4. 高级语言的特点

机器语言和汇编语言都是“面向机器”的程序设计语言,人们习惯上称它们为“低级语言”。由于它们难学、难记、程序难调试和可移植性差,所以在非计算机专业人员中不易推广。随着计算机的迅速普及和人们对解决日益增加的实际问题的需要,出现了各种形式的高级语言。高级语言又称为算法语言,是一种“面向问题”的程序设计语言。

高级语言是普及型的计算机程序设计语言,其各种命令的形式接近于自然语言和数学算式的格式表示。它们有着各自的特点,有着各自严格的语法规则,便于记忆、书写、阅读和修改。使用高级语言编写的程序的每一条命令,从字面上就能看出其含义。高级语言基本上摆脱了机器类型的影响,程序设计者在进行程序设计时可以不考虑机器的硬件结构,只需要掌握应用问题的解决方法和有关的算法,按照语言的语法规则书写命令,就可以编出程序。

使用某种高级语言编写出来的程序被称为该语言的源程序。计算机不能直接识别用高级语言编写的程序指令,必须将高级语言程序“翻译”成计算机可以直接识别的机器语言程序。然而,用人工进行这样的“翻译”实际上是不可能的。因此,人们在创造高级语言的同时还要编写出用计算机自身将高级语言程序“翻译”成机器语言程序的软件。这样的“翻译”软件叫做高级语言的编译软件(程序)。在编辑和执行高级语言程序的时候都需要有该种语言的编译软件的参与。

使用高级语言编程的方法和思路很接近于人与人之间的自然语言交流和数字描述，因此编程效率高，编程的时候基本上不涉及计算机的硬件知识，便于普及，程序的通用性好。但是高级语言程序不如机器语言简练，翻译转换后生成的目标程序冗余大，运行时占用内存多，速度较慢。

使用高级语言编写程序时与具体的计算机硬件无关，因此可大大简化程序的编制和调试工作，而且使用高级语言编写的程序通用性强、可移植性好，例如，Pascal 和 C 语言等高级程序设计语言就是典型的代表。

5. 几种典型的高级语言特性

典型的高级语言根据其应用领域、数据类型、语句、程序结构等方面，可分为 Fortran, Algol, Pascal, C, C++, Java, Cobol, Lisp, Prolog 等，而流行的结构化程序设计语言和面向对象程序设计语言是读者需要熟练掌握的两种程序设计语言风格。

(1) Cobol 是面向公共商业，用“自然英语”编程的语言。一个 Cobol 程序由 4 部分组成：PROCEDURE 部分书写算法，DATA 部分描述数据，ENVIRONMENT 说明程序和内部数据文件之间的联系，IDENTIFICATION 用于命名程序和程序作者，并提供其他注释作为程序文档。

(2) Lisp 是函数型语言，计算模型是 λ -演算。最初由 John McCarthy 在 1960 年设计和实现，在人工智能领域(如：机器人、自然语言理解、定理证明、智能系统)被大量使用。Lisp 在许多方面与其他语言不同。最为显著的是，该语言中的程序和数据的形式是相等的，这样数据结构可以作为执行程序，也可以作为数据修改。另一个明显的特征是对递归(recursion)的依赖，而在其他的语言中，循环是常用的。

Lisp 函数完全被作为表达式定义。每一个操作符是一个有返回值的函数，子程序以单一(通常很复杂)表达式写成。为了使这种纯粹的表达式语法看起来更像一般的声明顺序的语法，在语言中加入了各种特殊的构造。但是，表达式形式仍然是基本的。Lisp 程序通常解释执行，用户在终端上将主程序作为一串表达式输入。Lisp 系统计算每一个输入的表达式，自动地在终端上输出结果。

(3) Prolog 是一种使用谓词演算推出解决问题方法的逻辑型语言。Prolog 的目的是推导出解决问题的方法，Prolog 程序由一些 Horn 子句组成，描述一系列事实、数据对象或事实间的具体关系和规则的集合(构成数据库信息)。有了事实和规则后就可以进行查询，数据库里的事实和规则用来决定替换查询中的哪一个变量(叫做合一)。Prolog 具有很强的推理功能，适用于书写自动定理证明、专家系统、自然语言理解等问题的程序。

任何一个程序都可看成是对一组数据及作用于该组数据上操作的描述。数据类型是程序设计语言中的重要概念，是对数据的描述。数据类型的分类是根据数据的书写格式及作用其上的操作进行的。因此，一种数据类型是由集合及在这个值集合上的操作集合构成的。

不同的程序语言提供的数据类型不同，书写方法也不同。可参见第 2 章有关 C 语言的数据类型来理解数据类型的概念。

6. 程序设计语言的基本成分

程序设计语言的基本成分包括数据、运算、控制和传输等。

(1) 数据成分

程序语言的数据成分指的是一种程序语言的数据类型,数据对象总是对应着应用系统中某些有意义的东西。数据类型用于代表数据对象,还用于在基础机器中完成对值的布局,同时还用于检查表达式中对运算的应用是否正确。

数据是程序操作的对象,具有存储类、类型、名称、作用域和生存期等属性,使用时要为其分配内存空间。数据名称由用户通过标识符命名,标识符是由字母、数字和下划线组成;类型说明数据占用内存的大小和存放形式;存储类说明数据在内存中的位置和生存期;作用域则说明可以使用数据的代码范围;生存期说明数据占用内存的时间范围。从不同角度可将数据进行不同的划分。

① 常量和变量

按照程序运行时数据的值能否改变,将数据分为常量和变量。程序中的数据对象可以具有左值和右值。左值指存储单元(或地址、容器),右值是值(或内容)。变量具有左值和右值,在程序运行的过程中其右值可以改变;常量只有右值,在程序运行的过程中其右值不能改变。

② 全局量和局部量

按数据的作用域范围,可分为全局量和局部量。系统为全局变量分配的存储空间在程序运行的过程中一般是不改变的,而为局部变量分配的存储单元在程序运行的过程中是动态改变的。

③ 数据类型

按照数据组织形式的不同可将数据分为基本类型、用户定义类型、构造类型及其他类型。相关内容可见第2章。

(2) 运算成分

程序语言的运算成分指明允许使用的运算符及运算规则。大多数高级程序语言的基本运算可以分成算术运算、关系运算和逻辑运算等,有些语言如C(C++)还提供位运算。运算符的使用与数据类型密切相关。为了确保运算结果的惟一性,运算符要规定优先级和结合性,必要时还要使用圆括号。

(3) 控制成分

控制成分指明语言允许表述的控制结构,程序员使用控制成分来构造程序中的控制逻辑。理论上已经证明,可计算问题的程序都可以用顺序、选择和重复这三种控制结构来描述。有关内容可见第2章。

(4) 传输成分

程序语言的传输成分指明语言允许的数据传输方式,如数据的输入和输出等。

注意:大纲要求考生掌握过程(函数)的调用,此部分内容请读者务必理解。详见第2章介绍。

1.1.2 典型例题分析

例1: (2000年) 从供选择的答案中, 选出应填入下面叙述中___?___内的最确切的解答, 把相应编号写在对应栏内。

软件语言是用于书写计算机软件的语言。它主要包括需求定义语言、___(1)___、___(2)___、程序设计语言及___(3)___等, 适用于软件开发的各个阶段。程序设计语言的基本成分是数据成分、运算成分、控制成分及___(4)___。程序设计语言有多种分类法, 例如, 按成分性质分, 有顺序语言, 并发语言, 并行语言, ___(5)___。

供选择的答案:

- | | | | |
|---------------|----------|-----------|----------|
| (1) A. 数据定义语言 | B. 功能性语言 | C. 面向对象语言 | D. 函数式语言 |
| (2) A. 设计性语言 | B. 结构性语言 | C. 命令式语言 | D. 申述式语言 |
| (3) A. 过程语言 | B. 非过程语言 | C. 逻辑式语言 | D. 文档语言 |
| (4) A. 对象成分 | B. 变量成分 | C. 语句成分 | D. 传输成分 |
| (5) A. 交互式语言 | B. 分布语言 | C. 面向对象语言 | D. 高级语言 |

分析: 进行软件设计的过程大致经过需求分析、功能模块划分分析、详细设计、程序设计语言编码及文档维护和软件测试; 而软件设计中的数据流图是其设计的核心内容之一, 因而数据传输不可避免。鉴于软件实现的环境要求, 常采用不同的程序设计语言, 除了常规的顺序语言外, 还经常采用并发语言、并行语言及分布语言。

答案: (1) B (2) A (3) D (4) D (5) B

例2: (2000年) 从供选择的答案中, 选出应填入下面叙述中___?___内的最确切的解答, 把相应编号写在对应栏内。

编制一个好的程序首先要确保它的正确性和可靠性, 除此以外, 通常更注重源程序的___(1)___。此外, 还应强调良好的编程风格, 例如, 选择标识符的名字时应考虑___(2)___; 在书写语句时应考虑___(3)___; 在书写功能性注解时应考虑___(4)___。

源程序中应包含一些内部文档, 以帮助阅读和理解源程序, 源程序的内部文档通常包括选择合适的标识符、注解和___(5)___。

供选择的答案:

- | | |
|---------------------|-------------------|
| (1) A. 易使用性、易维护性和效率 | B. 易使用性、易维护性和易移植性 |
| C. 易理解性、易测试性和易修改性 | D. 易理解性、安全性和效率 |
- (2) A. 名字长度越短越好, 以减少源程序的输入量
- B. 多个变量共用一个名字, 以减少变量名的数目
- C. 选择含义明确的名字, 以正确提示所代表的实体
- D. 尽量用关键字作名字, 以使名字标准化
- (3) A. 把多个短的语句写在同一行中, 以减少源程序的行数
- B. 尽量使用标准文本以外的有特殊功效的语句, 以提高程序的功效
- C. 尽量消除表达式中的括号, 以简化表达式
- D. 避免使用测试条件“非”, 以提高程序的可读性

- (4) A. 仅为整个程序作注解 B. 仅为每个模块作注解
 C. 为程序段作注解 D. 为每个语句作注解
- (5) A. 程序的视觉组织 B. 尽量不用或少用 GOTO 语句
 C. 检查输入数据的有效性 D. 设计良好的输出报表

分析: 软件质量的好坏常采用可靠性、正确性、易使用性、规范性、易理解性或可读性、易维护性、执行效率和可移植性来判断。

易使用性是指用户操作是否简便、直观和易掌握等; 规范性包括系统的划分、书写的格式和变量的命名等规范要求, 如变量的命名采用其特定含义的名字, 这将有利于程序今后的阅读、修改和维护。

可读性是指程序清晰, 没有太多繁杂的技巧, 使他人能够容易读懂。可读性对于大规模工程化地开发软件是非常重要的。因而可读性程序是今后维护和修改程序的基础, 如果程序很难读懂, 则无法修改, 而无法修改的程序是没有生命力的程序。此外, 程序常采用嵌套等结构, 方便阅读。

易维护性是指程序各部分相互独立, 不会出现在维护时牵一发而动全身的连锁反应。

易移植性是指编写的软件可从一个运行环境移植到另一个运行环境, 希望软件系统能够适应不同的平台, 即跨平台运行。

答案: (1) B (2) C (3) D (4) C (5) A

例3: (1999年) 程序设计语言可划分为低级语言和高级语言两大类。与高级语言相比, 用低级语言开发的程序, 其(1), 但在(2) 的场合, 还经常全部或部分地使用低级语言。在低级语言中, 汇编语言与机器语言十分接近, 它使用了(3) 来提高程序的可读性。高级语言有很多种类, 其中, Prolog 是一种(4) 型语言, 其具有很强的(5) 能力。

- (1) A. 运行效率低, 开发效率低 B. 运行效率低, 开发效率高
 C. 运行效率高, 开发效率低 D. 运行效率高, 开发效率高
- (2) A. 对时间和空间有严格要求 B. 并行处理
 C. 事件驱动 D. 电子商务
- (3) A. 简单算术表达式 B. 助记符号
 C. 伪指令 D. 定义存储语句
- (4) A. 命令 B. 交互 C. 函数 D. 逻辑
- (5) A. 控制描述 B. 输入/输出 C. 函数定义 D. 逻辑推理

分析: 相对高级语言而言, 用低级语言开发的程序执行效率高, 可有效地降低时间和空间复杂度, 但开发周期长, 因而开发效率低。为了改进低级语言开发程序的可读性, 常采用助记符, 如汇编语言就采用助记符。为了提高逻辑表达能力, 人们除了设计出函数语言 Lisp 外, 还开发了逻辑性语言 Prolog, 它具有很强的逻辑表达能力。

答案: (1) C (2) A (3) B (4) D (5) D

例4: (1997年) 从供选择的答案中, 选出应填入下面叙述中? 内的最确切的解答, 把相应的编号写在对应栏内。

计算机程序设计语言经历了近 50 年的发展, 出现了许多不同的程序设计语言。