

高等学校规划教材

安徽省高等学校“十二五”规划教材



C/C++规范设计简明教程 ——思维训练、上机实验指导

C/C++ GUIFAN SHEJI JIANMING JIAOCHENG
——SIWEI XUNLIAN SHANGJISHIYAN ZHIDAO

李祎 编著



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
安徽大学出版社

C/C++ 规范设计简明教程

——思维训练、上机实验指导

李 祎 编著



北京师范大学出版集团

BEIJING NORMAL UNIVERSITY PUBLISHING GROUP

安徽大学出版社

图书在版编目(CIP)数据

C/C++规范设计简明教程:思维训练、上机实验指导/李祎编著.
—合肥:安徽大学出版社,2015.3
ISBN 978-7-5664-0899-0

I. ①C… II. ①李… III. ①C语言—程序设计—高等学校—
教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 028094 号

C/C++规范设计简明教程

——思维训练、上机实验指导

李祎 编著

出版发行:北京师范大学出版集团
安徽大学出版社
(安徽省合肥市肥西路3号 邮编 230039)
www.bnupg.com.cn
www.ahupress.com.cn

印 刷:合肥现代印务有限公司
经 销:全国新华书店
开 本:184mm×260mm
印 张:16.5
字 数:402千字
版 次:2015年3月第1版
印 次:2015年3月第1次印刷
定 价:33.00元
ISBN 978-7-5664-0899-0

策划编辑:李 梅 蒋 芳
责任编辑:蒋 芳
责任校对:程中业

装帧设计:李 军
美术编辑:李 军
责任印制:赵明炎

版权所有 侵权必究

反盗版、侵权举报电话:0551-65106311

外埠邮购电话:0551-65107716

本书如有印装质量问题,请与印制管理部联系调换。

印制管理部电话:0551-65106311

前言

学而不思则罔,思而不学则殆,学思不行则废,基于这一思想,本课程采用“学思行”三位一体的教学模式,本手册提供“思行”练习依据。

“思”指思维训练,这是一个重要的教学环节,是“学——理论教学”和“行——上机实验”联系的桥梁和缓冲。本教学环节通过精心设计不同题型的练习,不仅巩固和加深课堂理论教学,同时提供了一个以学生为主的师生双向交流平台,学生可利用此平台充分表达观点、展示思路、提出疑问。思维训练主要包括以下三个部分:

(1)自测部分

这个部分需学生个人独立完成,练习题型主要包括与教材相关度高的简答题、选择题、判断题、画图题、改错题、同型题,自测练习是度量学生掌握基础知识的必要手段。

(2)答辩部分

这个部分需学生以小组为单位合作完成,练习题主要包括理论教学衍生出来的变式题,通过多角度观察并解决问题,从而真正掌握基础概念,培养学生扎实的编程能力。

答辩部分教学,是学生经过自我思索、合作交流后并用语言表达出来,是实现知识同构的强有力手段,同时合作学习的方式能够更加有效地促进问题的解决和团队凝聚力的提升。

(3)提高部分

这个部分供学有余力的学生主动自我学习,练习题主要包括一些难度较高的题目或综合性的题目,培养学生全面应用知识和深度思考问题的能力。

“行”指上机实验,实验题目与教学内容相对应,并形成梯度。以面向过程、面向对象的编程思想为指导,围绕“学生成绩管理系统”的构建,不断代入新的环境,反复锤炼,做到代码复用,让学生更深刻地体会编程过程。

为了不限制学生思维,本书不提供答案,需验证结果可访问 <http://hfxyliyi.ysl68.com>, 下载各章节参考答案。

编者

2014年11月

目 录

答辩要求与答辩示例	1
-----------------	---

答辩要求	1
------------	---

答辩示例	1
------------	---

实验要求与实验示例	2
-----------------	---

实验要求	2
------------	---

实验示例	2
------------	---

第 1 章 模型与模块概述	7
---------------------	---

1.1 目标与要求	7
-----------------	---

1.2 解释与扩展	7
-----------------	---

1.3 思维训练题—自测练习	9
----------------------	---

1.4 思维训练题—答辩练习	14
----------------------	----

1.5 思维训练题—阅读提高	15
----------------------	----

1.6 上机实验	15
----------------	----

第 2 章 调试技术	18
------------------	----

2.1 目标与要求	18
-----------------	----

2.2 解释与扩展	18
-----------------	----

2.3 思维训练题—自测练习	20
----------------------	----

2.4 思维训练题—答辩练习	25
----------------------	----

2.5 思维训练题—阅读提高	25
----------------------	----

2.6 上机实验	26
----------------	----

第 3 章 基本数据类型	27
--------------------	----

3.1 目标与要求	27
-----------------	----

3.2 解释与扩展	27
-----------------	----

3.3 思维训练题—自测练习	31
----------------------	----

3.4 思维训练题—答辩练习	34
----------------------	----

3.5 思维训练题—阅读提高	35
----------------------	----

3.6 上机实验	35
----------------	----

第 4 章 高级数据类型	37
--------------------	----

4.1 目标与要求	37
-----------------	----

4.2 解释与扩展	37
-----------------	----

4.3 思维训练题—自测练习	38
----------------------	----

4.4 思维训练题—答辩练习	40
----------------------	----

4.5 思维训练题—阅读提高	43
----------------------	----

4.6 上机实验	43
----------------	----

第 5 章 结构编程之顺序与选择	46
------------------------	----

5.1 目标与要求	46
-----------------	----

5.2 解释与扩展	46
-----------------	----

5.3 思维训练题—自测练习	50
----------------------	----

5.4 思维训练题—答辩练习	54
----------------------	----

5.5 思维训练题—阅读提高	55
----------------------	----

5.6 上机实验	55
----------------	----

第 6 章 结构编程之循环	57
---------------------	----

6.1 目标与要求	57
-----------------	----

6.2 解释与扩展	57
-----------------	----

6.3 思维训练题—自测练习	65
----------------------	----

6.4 思维训练题—答辩练习	70
----------------------	----

6.5 思维训练题—阅读提高	73
----------------------	----

6.6 上机实验	76
----------------	----

第 7 章 数 组	77
-----------------	----

7.1 目标与要求	77
-----------------	----

7.2	解释与扩展	77	11.3	思维训练题—自测练习 ..	168
7.3	思维训练题—自测练习	92	11.4	思维训练题—答辩练习 ..	174
7.4	思维训练题—答辩练习	95	11.5	思维训练题—阅读提高 ..	175
7.5	思维训练题—阅读提高	96	11.6	上机实验	178
7.6	上机实验	103	第 12 章 继 承	180	
第 8 章 字符串	105		12.1	目标与要求	180
8.1	目标与要求	105	12.2	解释与扩展	180
8.2	解释与扩展	105	12.3	思维训练题—自测练习 ..	203
8.3	思维训练题—自测练习	106	12.4	思维训练题—答辩练习 ..	205
8.4	思维训练题—答辩练习	112	12.5	思维训练题—阅读提高 ..	206
8.5	思维训练题—阅读提高	113	12.6	上机实验	207
8.6	上机实验	126	第 13 章 多态转型	209	
第 9 章 结构体	128		13.1	目标与要求	209
9.1	目标与要求	128	13.2	解释与扩展	209
9.2	解释与扩展	128	13.3	思维训练题—自测练习 ..	213
9.3	思维训练题—自测练习	132	13.4	思维训练题—答辩练习 ..	216
9.4	思维训练题—答辩练习	136	13.5	思维训练题—阅读提高 ..	217
9.5	思维训练题—阅读提高	137	13.6	上机实验	218
9.6	上机实验	141	单元总结与自检测试	220	
第 10 章 文件操作	143		单元 1	模型模块、数据类型总结与 自检测试	220
10.1	目标与要求	143	单元 2	结构编程总结与自检测试	226
10.2	解释与扩展	143	单元 3	构造类型总结与自检测试	236
10.3	思维训练题—自测练习 ..	154	单元 4	封装、继承、多态总结与自 检测试	242
10.4	思维训练题—答辩练习 ..	158	课程结束总结与模拟考核	248	
10.5	思维训练题—阅读提高 ..	159	附录 课程设计	257	
10.6	上机实验	161			
第 11 章 类和对象	163				
11.1	目标与要求	163			
11.2	解释与扩展	163			

答辩要求与答辩示例

◆ 答辩要求

(1) 答辩小组

根据班级人数成立 3~6 人为一组的答辩小组。

(2) 答辩思路

答辩以模块分析为核心,遵循模块设计的基本步骤:模块功能、输入输出、解决思路、算法步骤、模块代码。

(3) 强调合作

答辩小组答辩前先做好分工,每位同学在对整体结构认识的基础上,各有分工,都要有表达机会。如有的同学重点在模块分析,有的同学重点在代码描述,有的同学重点在演示程序,有的同学重点在模型设计等。

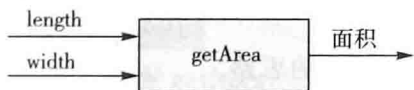
◆ 答辩示例

答辩题目:编写模块,根据长方形的长和宽(均为整数),返回长方形的面积。

模块设计(本题只要求编写自定义模块,所以不从主模块开始分析):

① 模块功能:求长方形的面积。

② 输入输出:



形式: `int getArea(int length, int width)`

归属: `int`

③ 解决思路:根据长方形的面积计算公式进行求解,面积=长*宽。

④ 算法步骤:首先计算面积,然后返回面积。

⑤ 模块代码:

```
int getArea(int length, int width)
{
    return length * width;
}
```

实验要求与实验示例

◆实验要求

(1) 实验前准备

实验前要求:实验题目、概要分析、模型设计、模块设计、程序代码、数据预测。

实验题目:按实验指导书里给定的题目进行实验。

概要分析:①解决的问题是什么,解决这个问题的关键是什么;②需要采用什么方案,是单文档还是多文档;③写出程序实现所需要建立的各文件,并指出文件之间的关系;④程序使用何种数据类型。

模型设计:按要求绘制模型结构图,清楚表达文件结构、模块结构。

模块设计:要求至少给出一到两个核心模块的设计,每个模块包括:模块功能、模块参数、解决思路、算法步骤。

程序代码:按模型设计中设计的模块撰写代码。

数据预测:给出实验数据,并预测结果。

(2) 实验中调试

实验中要求:调试、纠错步骤方法。

(3) 实验后总结

实验后要求:实验总结、实验思考、实验自评。

实验总结:总结要真实地反映实验过程的心得。

实验思考:实验指导书后面的思考题,要给出答案或者解决的思路。

实验自评:①项目完整;②版面整洁;③设计合理;④调试详实;⑤总结深刻。

◆实验示例

实验一 模型模块

[实验题目]

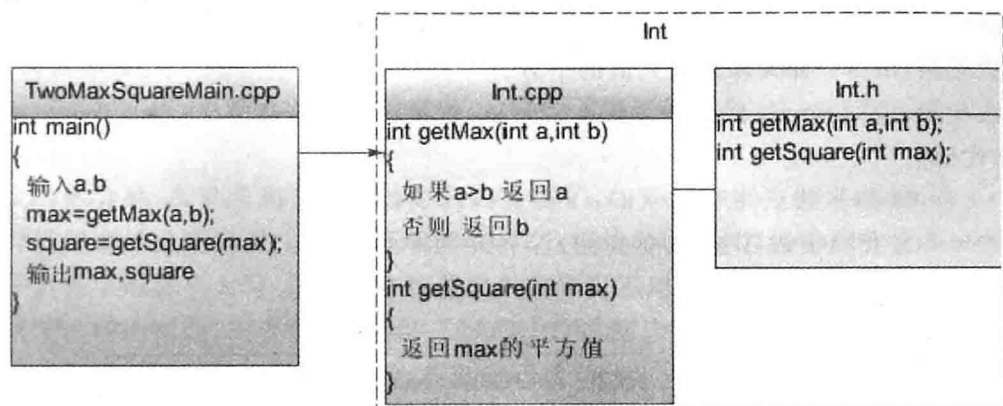
采用多文档模型结构实现如下要求:输入 2 个整数,编程求这 2 个数中较大数的平方。

[概要分析]

本程序要实现一个简单数运算的问题,求两个整数的较大数只需要用一个判断即可完成,得到较大值之后再求平方值。

本程序采用多文档多模块模型结构,建立的主模块的文件名为 TwoMaxSquareMain.cpp;自定义模块是针对简单整数的运算,所以起名为 Int,自定义模块所在的文件名为 Int.cpp,而相应的声明文件是 Int.h,主模块与自定义模块之间的关系是依赖的关系,主模块要依赖自定义模块来实现。

[模型设计]



[模块设计]

文件 TwoMaxSquareMain.cpp

主模块 main

①模块功能:这是主模块,要求两个整数较大数的平方。

②输入输出:系统模块,暂不考虑模块级别的输入输出,输入输出在内部完成。

形式: `int main()`

归属: TwoMaxSquareMain

③解决思路:先求出 max,再根据 max 求出 square。

④算法提纲:

第一步:输入两个整数 a,b。

第二步:通过自定义模块 `getMax(a,b)`,求出较大数赋给 max。第三步:通过自定义模块 `getSquare(max)`,求出平方值赋给 square。

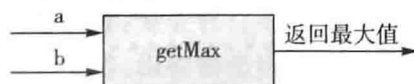
第四步:输出 square。

文件 Int.cpp

自定义模块 getMax

①模块功能:求两个整数的较大值。

②输入输出:

形式: `int getMax(int a,int b)`

归属: int

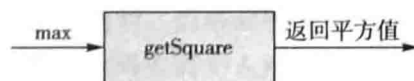
③解决思路:比较 a 与 b,得到较大值。

④算法步骤:如果 $a > b$,返回 a;否则,返回 b。

自定义模块 getSquare

①模块功能:求一个整数的平方值。

②输入输出:



形式: `int getSquare(int max)`

归属: `Int`

③解决思路: `max * max` 即为较大值的平方。

④算法步骤: 返回 `max * max`。

[程序代码]

```

/1 ****
filename:F:\c++\chapt1\TwoMaxSquareProj\TwoMaxSquareMain.cpp
author:李伟
****/

int main ()
{
    int a,b,max,square;
    cin>>a>>b;
    max=getMax(a,b);
    square=getSquare(max);
    cout<<"最大值的平方是"<<square;
    return 0;
}

/2 ****
filename:F:\c++\chapt1\TwoMaxSquareProj\Int.cpp
author:李伟
****/

int getMax(int a,int b)
{
    if(a>b) return a;
    else return b;
}

int getSquare (int max)
{
    return max * max;
}

/3 ****
filename:F:\c++\chapt1\TwoMaxSquareProj\Int.h
author:李伟
****/

int getMax(int a,int b);
int getSquare(int max);

```

现象: error C2065: 'cout': undeclared identifier

原因: 没有声明的标识符cout

改正: 模块前面加上声明, 见#include <iostream.h>

现象: error C2065: 'getMax': undeclared identifier

原因: 没有声明的标识符'getMax',

模块前面加上声明

改正: #include "Int.h"

[数据预测]

输入 3 5

结果: 最大值的平方是 25

输入 -3 -5

结果: 最大值的平方是 9

[实验调试]

出错信息、原因及改正代码见程序代码右边。

[实验总结]

通过本次实验,既饱尝了编程的辛苦,也体会了编程的乐趣,经过反复地调试,程序终于运行成功。

我的体会是:文件名和变量名的命名太重要了,如果没有事先做好规划,输入这些标识符的时候很容易出现错误。首先,不管是文件名、模块名、变量名都要用英文单词来表达,其次文件名首字母都是大写,模块名和变量名的首字母不用大写,其他一样。

对于出现的错误,不要慌张,找到一个 ERROR,阅读并分析错误原因,双击之后就能找到错误所在行,再针对错误进行修改,直到运行成功。

[实验思考]

1. 按“Ctrl+F5”相当于同时执行了哪几步?

答:编译+连接+运行。

2. 模块或文件的命名规范是什么?

答:模块或文件的命名规范:模块名或文件名要有意义,并用动词或动词词组表示,首字母小写,其余单词开头大写。

[自我评价]

1. 项目完整-----A

2. 版面整洁-----A

3. 设计合理-----A

4. 调试细致-----A

5. 总结深刻-----A

◆基于合作方式的电子实验报告

实验报告可采用“小组合作+个人任务”的电子稿方式,这样安排,可减少学生书写负担,同时更强调设计与合作。

1. 报告栏目

实验题目:来源于“上机实验部分”。

模型设计:专业软件绘制结构清晰的模型图。

分工安排:根据模型图,确定小组中个人的工作内容(具体到各模块)。

共享资料:确定实验共享资料,包括共享位置、共享文档、共享代码。

模块描述:负责模块功能、输入输出、设计思路、算法步骤、模块代码。

预测结果:根据事先给定的数据,预测可能产生的结果。

实验调试:明确错误及原因,并改正。

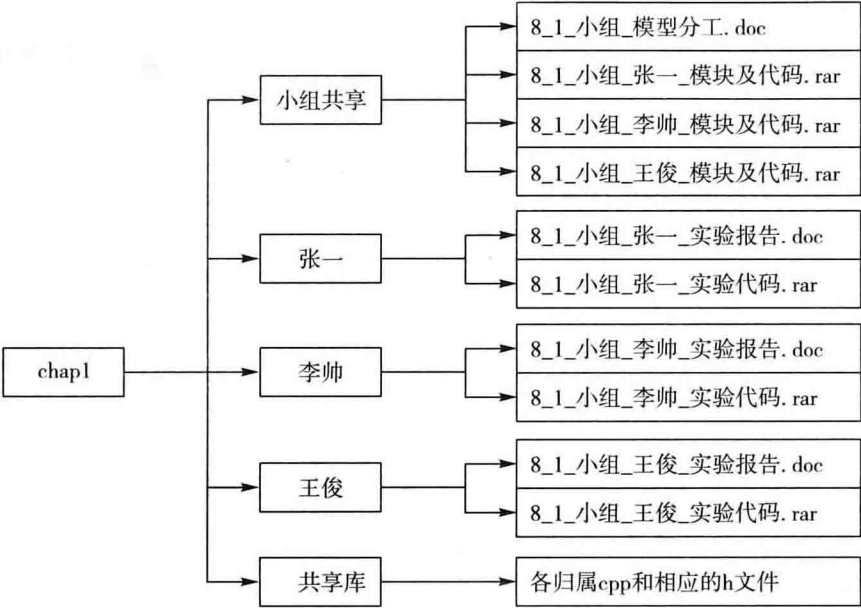
总结思考:总结分析,并完成思考练习题。

2. 关于共享资料

小组共享提供了一个交流的平台,这是合作编程能力培养的前提,同时为个人电子报告的撰写提供了便利。但必须明确本课程的所有课内实验,个人分工外的模块务必全部掌握(4个步骤),模块代码也需个人亲自编写。个人报告可综合小组共享中提供的内容进行编写。

3. 上传文档代码结构与格式

以第 8 小组做第一章实验为例说明上传结构与文档格式



4. 实验考核

考核标准:达到下述标准为合格,否则为不合格。

- ①准备充分,按规定格式要求上传资料。
- ②个人报告栏目完整。
- ③实验过程认真。
- ④实验结果现场验收,接受教师提问并正确回答。

第1章 模型与模块概述

1.1 目标与要求

1. 了解计算机语言的发展。
2. 了解面向过程和面向对象的本质区别。
3. 初步掌握模块化设计思想。
4. 初步掌握模块的组织形式,即模型结构。
5. 初步掌握简单函数的调用规则。
6. 初步了解 VC 开发环境。

1.2 解释与扩展

1. 程序世界的世界观、方法论、表达建模、语法规则

(1) 世界观

① 科学的极致是哲学。

“艺术的极致是科学,科学的极致是哲学”,此话不无道理,牛顿、爱因斯坦等科学界泰斗,晚年都不约而同地转向哲学研究。

② 世界观是哲学的根本。

马克思曾说过:“世界观是人们对世界的总的最根本的看法。任何哲学问题的探讨,归其出发点和本源,都是世界观的问题。什么样的世界观决定了什么样的哲学观点”。

古往今来,有许多截然不同的世界观,例如,唯物主义与唯心主义、形而上学与辩证法、可知论与不可知论等。

世界只有一个,但每个人的观点是不同的。孔子的世界观即孔子的“道”。孔子曾说:“朝闻道,夕死可矣。”这个“道”就是“精气为物,游魂为变”,说明了万物、生命的起源。那么“游魂”又从何而来?孔子进一步指出:“《易》有太极,是生两仪。”太极天然而有,寂静光明,无生无死。太极动则显象起用,所生之象,其数无穷,但始动之际,阴阳消长,于是太极失明,转变为游魂,流转生死。所以,儒家认为太极生阴阳,阴阳生五行,五行生万物。五行体现于人则是五常,即仁、义、礼、智、信,做到仁、义、礼、智、信就是符合“道”。

③ 程序世界的世界观。

程序世界的世界观包括过程观和对象观。

不论是过程观还是对象观,都承认一点,那就是程序世界本质上只有两种东西——数据和逻辑。数据天性喜静,构成了程序世界的本体和状态;逻辑天性好动,作用于数据,推动程序世界的演进和发展。尽管上述观点是统一的,但是在数据和逻辑的存在形式和演进形式

上,过程论和对象论的观点截然不同。

过程观认为:数据和逻辑是分离的、独立的,各自形成程序世界的一个方面。所谓“世界的演变”,是在逻辑作用下,数据改变的一个过程。这种过程有明确的开始、输入、输出、结束,步骤间存在严格的因果关系。过程是相对稳定的、明确的和预定义的,小过程组合成大过程,大过程还可以组合成更大的过程。所以,程序世界本质是可控的过程世界,数据作为过程的处理对象,逻辑作为过程的形式定义。

对象观认为:数据和逻辑不是分离的,而是相互依存的。相关的数据和逻辑形成个体,这些个体叫做对象(Object),世界就是由一个个对象组成的。对象具有相对独立性,对外提供一定的服务。所谓“世界的演进”是指在某个“初始作用力”作用下,对象间通过相互调用而完成的交互;在没有初始作用力下,对象保持静止。这些交互并不是完全预定义的,不一定有严格的因果关系,对象间交互是“偶然的”,对象间联系是“暂时的”。世界就是由各色对象组成,然后在初始作用力下,对象间的交互完成了世界的演进。所以世界的本质是对象及其相互联系。世界的本质是一个不可控的对象世界。

(2) 方法论

世界观决定方法论,面向过程的方法论是:自顶而下,逐步分解。例如,烧排骨的过程是先买,再洗,再切,再烧,最后装盘。面向对象的方法论是:抽象,精化,联系。例如,一个大学的管理,抽象出院长、系主任、教研室主任等对象,在初始作用力“提高教学,培养合格人才”的作用下,各对象自行联系解决问题,实现目标。

(3) 表达建模

设计出一套相应的规范结构以保证方法论顺利执行,结构设计也称为“建模”。

过程方法论的模型结构通常用层次图、数据流图来表达。“模型模块”设计,是认识现实世界和解构世界、解决问题的一种简单建模方案,绘制模型模块结构图的过程是设计思路不断清晰、深化的过程,也是具体编程(包括合作编程)的依据。

对象方法的模型结构通常用例图、类图、序列图等方法来表示。

(4) 语法规则

有了世界观、方法论及具体的表达方案后才可以去编程,不同的计算机语言虽然格式上有些区别,但语法规则是相同的。

2. C/C++两种语言和面向过程、面向对象两种思想

(1) 是否应该先学C再学C++

有不少人认为C是基础,C++是提高,所以应该先学C再学C++。C++之父Bjarne Stroustrup认为先学C没有必要,这是混淆了认识、解决问题的两种思维(世界观方法论)和两种描述语言。

(2) 思想和语言

解决问题有面向过程、面向对象两种思想。通常人们认为C是面向过程的开发语言,而C++是面向对象的开发语言,所以,用C来开发的程序就是面向过程的,而用C++开发的程序就是面向对象的。

实际上,如果从面向对象的角度去考虑问题,C也可以编写出面向对象的程序;如果出发点是面向过程,编写的程序中即使包含了C++的语法特点(如类C++语言的特性,C语言里没有),依然是面向过程。

(3) 面向过程和面向对象各有优势

在一些实时反应,效率要求很高的问题中,用面向过程解决直接快捷。比如说,单片机的控制、信号的读取等。如果解决复杂的系统,要考虑系统里各个对象的权利、通信、协作等,用面向过程不是不可以,但用面向对象的思想会促使我们去建立各种模型、各种对象、各种联系,会使得我们对结构把握得更清楚。

这就像盖一个狗棚,就不需要画图纸,第一步,找些泥来;第二步,拿点水;第三步,把泥和起来;第四步,垒泥巴完事。狗棚倒了也没有关系,还可以重修重盖。如果盖一幢 100 层的大楼,首先要进行设计规划,然后选出各方面的项目经理,再将设计思想跟他们讲清楚,并向他们提供很多的图纸,包括电气分配图、材料图、立体模型图等。

(4) 面向过程和面向对象需要相互融合

面向过程要求分步骤:第一步,第二步,第三步……第 n 步,做后一步的依据是前一步做好了,每一步你都要事必躬亲,如果一步做不下来,就无法再做下一步;面向对象的处理方式是:将事件流按功能相近的事件交给一个专门的对象(人)去处理,不需要管细节,只要管那个人就可以了,他做好了,向你汇报,你再去看看第二个人工作有没有做好。你是从管理者的角度来管理对象的,不用再辛苦地管理细节,你是面向对象的,而被管理者去做事情的时候,他会按照步骤一步步地向下做,他是面向过程的。

可见,面向对象和面向过程并不冲突,在一个大型的系统设计中两者是紧密结合在一起的,只是侧重点不一样。在做整体架构的时候多用面向对象的思想去考虑,而在实现细节上大都用面向过程的思想去解决。面向对象里包括了面向过程,不应截然分开,掌握面向对象的基础是面向过程,面向过程和面向对象需要相互融合。

(5) 借助C/C++语言学习面向过程和面向对象的思想

C 和 C++ 是两种语言,C++ 虽然主要是以 C 为基础发展起来的一门新语言,但它不是 C 的替代品,从语言的角度来看没有谁比谁先进的说法,这两门语言各有各的应用范围,没有先学 C 再学 C++ 的说法。目前,C 和 C++ 各自的标准委员会是独立的,最新的 C++ 标准是 C++98,最新的 C 标准是 C99。

我们将 C/C++ 放在一起学习,是基于这两种语言的特点优势及语法规则的关联,更重要的是借助这两种语言学习面向过程和面向对象两种处理问题的方法。

1.3 思维训练题—自测练习

1. 简答题

- (1) 面向过程结构化编程的宗旨是什么?
- (2) 模块又称什么?
- (3) 模块和文件的关系是什么?
- (4) 程序运行从哪个模块开始,至哪个模块结束?
- (5) 单文档模型和多文档模型写程序都有哪 3 个部分?
- (6) 为什么 `sqrt` 函数前面必须加上 `#include <math.h>`?
- (7) 程序编译之后生成的目标文件为什么不能执行?
- (8) 以下是求 3 个整型数平方和的部分代码。哪个写得较好,为什么?

第一种写法:主模块所在的文件 SquareSumMain. cpp。

```

/*****
    created: 2008/02/08
    file base: SquareSumMain
    file ext: cpp
    author: 李祎
    purpose: 这是求 3 个整型数的平方和的主模块所有的文件
*****/
int main()
{
    int data1,data2,data3,result;        //定义变量
    cin>>data1>>data2>>data3;          //输入 3 个变量的值
    result=squareSum(data1,data2,data3); //调用函数求平方和,返回值给 result
    cout<<result;                        //输出最后的平方和
    return 0;
}

```

第二种写法:主模块所在的文件 Test. cpp。

```

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    int d;
    d=f(a,b,c);
    cout<<f;
    return 0;
}

```

选择较好的方案并说明理由:

(9)请分析下面程序运行后的结果,代码如下:

```

#include <iostream.h>
int add(int a,int b);
int main()
{
    int a,b,c;
    a=2;
    b=3;
    c=add(a,b);
    cout<<"两个变量的和是:"<<c;
    return 0;
}

int add(int a,int b)
{
    return a+b;
}

```


(10)根据第(8)题中调用的 squareSum 模块,及源代码:“result = squareSum(data1, data2, data3);”,画出 squareSum 的模块图。

2. 不定项选择题

(1)如果源文件是 MaxTwoMain. cpp,经过编译后生成的文件是()。

- (A)MaxTwoMain. exe
- (B)MaxTwoMain. lnk
- (C)MaxTwoMain. obj
- (D)MaxTwoMain. dsw

(2)Matlab 属于()。

- (A)系统软件
- (B)应用软件
- (C)嵌入软件
- (D)语言软件

(3)下列关于 main 模块的描述,正确的是()。

- (A)程序中必须有的模块
- (B)可有可无
- (C)视程序大小可以有多个
- (D)这是程序的开始模块,但不是终点

(4)“成绩管理系统”需要编写若干个自定义模块,如“输入分数模块”、“查询分数模块”、“显示分数模块”,那么这些模块归属下面()名称是合适的。

- (A)Elephant
- (B)ScoreManager
- (C)Student
- (D)StudentManager

(5)以下()符号是程序的注释符号。

- (A){}
- (B)//
- (C)* / / *
- (D)remember

(6)程序中使用 cin 输入数据,使用前必须加()。

- (A)#include <iostream. h>
- (B)#include <cin. h>
- (C)直接使用,表示输入
- (D)#include"cin. h"

(7)建立控制台程序必须先建立一个项目,如果建立的项目名为 SimpleProj. dsp,项目下建立 3 个文件:SimpleMain. cpp、Int. cpp、Int. h,编译连接后生成的可执行文件名为()。

- (A)SimpleProj. exe
- (B)SimpleProj. obj
- (C)SimpleMain. exe
- (D)SimpleMain. obj

(8)下面()描述是正确的。

- (A)输入、输出模块功能决定了模块头部的写法
- (B)模块名通常是动词,因为它表达某一个具体执行的功能
- (C)使用一个模块,要在前面加上这个模块的说明清单
- (D)使用一个模块,必须要定义这个模块,否则不能使用

(9)编写模块的思路应该按如下()步骤进行。

- (A)模块功能、输入与输出、解决思路、算法步骤、模块代码
- (B)自顶而下、逐步求精、面向过程、结构编程
- (C)建立对象、发生关联、不断迭代
- (D)顺序、选择、循环

(10)自定义模块格式为:int getMax(int a,int b,int c),请选择其相对应的模块结构图()。