

高等院校信息类专业校企衔接创新实践系列教材

C++ 项目实训

□ 主 编 李渤 安海龙
□ 副主编 刘冰 罗来俊 徐青翠

```
#include <iostream>
using namespace std;
class Base
{
public:
    Base(int i)
    {
        Value = i;
    }
    int Value;
    void func()
    {
        cout<<"member of Base "<<Value<<endl;
    }
}
```

```
Base1:virtual public Base
```

```
public:
    Base1(int a):Base(a)
    {}
    int Value1;
};
class Base2:virtual public Base
```



中南大学出版社
www.csupress.com.cn

高等院校信息类专业校企衔接创新实践系列教材

C++ 项目实训

主 编 李渤 安海龙
副主编 刘冰 罗来俊 徐青翠

```
#include <iostream>
using namespace std;
class Base
{
public:
    Base(int i)
    {
    }
    // _____
    void func()
    {
        cout<<"member of Base "<<Value<<endl;
    }
};

Base1:virtual public Base
{
public:
    Base1(int a):Base(a)
    {}
    int Value1;
};

class Base2:virtual public Base
```



中南大学出版社
www.csupress.com.cn

· 长沙 ·

图书在版编目 (C I P) 数据

C + + 项目实训 / 李渤, 安海龙主编. --长沙: 中南
大学出版社, 2018. 11

ISBN 978 - 7 - 5487 - 3243 - 3

I. ①C… II. ①李… ②安… III. ①C + + 语言—程序
设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 245332 号

C + + 项目实训

C + + XIANGMU SHIXUN

李 渤 安海龙 主编

☐责任编辑 胡小锋

☐责任印制 易建国

☐出版发行 中南大学出版社

社址: 长沙市麓山南路

邮编: 410083

发行科电话: 0731 - 88876770

传真: 0731 - 88710482

☐印 装 长沙市宏发印刷有限公司

☐开 本 787 × 1092 1/16 ☐印张 21 ☐字数 536 千字

☐版 次 2018 年 11 月第 1 版 ☐2018 年 11 月第 1 次印刷

☐书 号 ISBN 978 - 7 - 5487 - 3243 - 3

☐定 价 59.00 元

图书出现印装问题, 请与经销商调换

内容简介

本书基于 CDIO 工程教育理念,结合应用型人才培养的目标和要求,针对读者“重理论,轻实践”的特点,在阐述程序语言基本范式及算法的基本理论之后,注重通过实例培养读者动手编程的能力。全书从功能应用的角度分为 3 篇、18 章,覆盖了 C++ 应用开发全部基本知识点,每个知识点都配备了一定的示例,并在第三部分给出 8 个具有代表性的综合应用实例。实例从需求分析、设计、编码、测试等软件开发各阶段进行了详细说明,并给出了全部的程序代码。本书能够帮助读者解决只能做简单片段程序不能开发完整工程的问题。每个实例都具有一定的通用性,能够快速移植,帮助学习者更好地学以致用。

本书适合作为高等院校开设的面向对象程序设计、算法与数据结构等应用开发课程的教科书,也可作为计算机程序设计培训班的教材或计算机程序设计人员的参考书。

前 言

众所周知,当前社会需求和高校课程设置严重脱节,一方面企业找不到可迅速上手的人才,另一方面大学生就业难。如果有一些面向工作应用的案例参考书,让广大学生得以参考,并能亲手去做,势必能缓解这种矛盾。本书就是这样一本书:项目开发案例型的、面向工作应用的软件开发类图书。编写本书的目的就是架起让学生从学校走向社会的桥梁。

本书以完成小型项目为目的,让学生切身感受到软件开发给工作所带来的实实在在的用处和方便,激发学生学习软件的兴趣,让学生变被动学习为自主自发学习。本书的项目开发案例过程完整,不但适合在学习软件开发时作为小型项目开发的参考书,而且可以作为毕业设计的案例参考书。

本书共有3篇、18章。第1篇只有第1章,概述了C++的基础知识。第2篇共9章,从第2章至第9章,分别介绍数据结构和算法的基本概念,包括线性结构的线性表、栈与队列、串、多维数组、树与二叉树、图、查找和内排序等。并在此部分的第10章对贪心法、分治法、动态规划法、回溯法等经典算法作了描述。第3篇共8章,包括基于Qt的电子点餐系统、多文本编辑器的设计与实现、俄罗斯方块、基于Qt的画图板功能的实现、视频监控系统的设计与实现、基于Qt的扫雷游戏设计与实现、基于Qt的图书管理系统和网络版中国象棋等8个综合实例。

本书作者有从事多年C++程序开发经验的一线教育工作者,他们具有丰富的教学实践经验,还有上海杰普软件公司一线软件工程师,编程实例由他们撰写,实例全部来源实际案例。本书总体呈现出以下特点:

(1)内容编写既考虑经典范例,也吸收了最新应用内容。由浅入深,循序渐进,层次分明;语言讲解通俗易懂、突出重点。

(2)每章节都配有精心设计的应用例题,用以帮助读者更好地理解和掌握该章节知识点,例题的代码已作了详细的注释。第2篇每章末配有精选习题,用以强化C++应用程序设计知识和技能的训练。

(3)结合每章的内容,编写了综合运用实例,既可作为各章教学的参考,也可作为该章知识点应用的综合实训项目。

(4)本书的例题和习题,都已在VS的开发环境下调试与运行。

本书由江西理工大学李渤、上海杰普软件科技有限公司安海龙担任主编。第1章由江西

理工大学徐青翠组织编写，第2、4、6章由江西理工大学罗来俊组织编写，第3、5、7、10章由江西理工大学刘冰组织编写，第8、9章由李渤组织编写，第11章至第18章由安海龙组织编写。全书由李渤统稿并定稿。

在本书的编写过程中，得到了许多老师和同学的大力支持和热情帮助。本书的出版得到了江西省教育科学“十二五”规划项目“基于CDIO-CMM理念的螺旋式软件人才培养模式的探索与效应研究”(项目编号:12YB125)的基金项目资助。在此表示衷心的感谢!同时，编者参阅了大量的C++应用程序设计的书籍和网上资源。在此，对它们的作者和提供者一并表示衷心的感谢。

由于编者水平有限，书中一定存在着错误或陈述不妥之处，恳请读者批评指正，以便再版时修改完善。

编 者

2017年12月

目 录

第 1 篇 基础知识

第 1 章 C + + 编程基础	(3)
1.1 C + + 简介	(3)
1.2 数据类型	(6)
1.3 常量和变量	(8)
1.4 运算符和表达式	(9)
1.5 const 关键字	(11)
1.6 控制语句	(15)
1.7 类	(21)
1.8 流类库	(58)
1.9 模板	(59)
1.10 例题解析	(63)

第 2 篇 数据结构与算法基础

第 2 章 线性表	(81)
2.1 基础知识	(81)
2.2 存储结构和基本运算	(81)
2.3 例题解析	(86)
2.4 习题 1	(92)
第 3 章 栈与队列	(93)
3.1 基础知识	(93)
3.2 存储结构和基本运算	(93)
3.3 例题解析	(105)
3.4 栈与队列实践	(109)
3.5 习题 2	(115)

第4章 串	(116)
4.1 基础知识	(116)
4.2 串的存储结构和基本运算	(116)
4.3 串的典型算法(模式匹配算法)	(118)
4.4 例题解析	(120)
4.5 习题3	(122)
第5章 多维数组	(123)
5.1 基础知识	(123)
5.2 存储结构和基本运算	(124)
5.3 例题解析	(129)
5.4 多维数组实践——对称矩阵的压缩存储	(131)
5.5 习题4	(132)
第6章 树与二叉树	(133)
6.1 基础知识	(133)
6.2 存储结构	(136)
6.3 树与二叉树的遍历	(139)
6.4 例题解析	(140)
6.5 习题5	(143)
第7章 图	(144)
7.1 基础知识	(144)
7.2 存储结构和基本运算	(147)
7.3 例题解析	(159)
7.4 图实践	(162)
7.5 习题6	(168)
第8章 查找	(169)
8.1 基础知识	(169)
8.2 查找的概念	(169)
8.3 查找方法评价指标	(170)
8.4 静态查找表	(170)
8.5 动态查找	(173)
8.6 习题7	(174)
第9章 内排序	(175)
9.1 基础知识	(175)

9.2	插入排序	(176)
9.3	希尔排序	(179)
9.4	冒泡排序	(180)
9.5	快速排序	(181)
9.6	简单选择排序	(182)
9.7	习题 8	(183)
第 10 章 经典算法分析与实现		(184)
10.1	贪心法	(184)
10.2	分治法	(193)
10.3	动态规划法	(203)
10.4	回溯法	(214)
10.5	习题 9	(223)
 第 3 篇 软件项目实习		
第 11 章 基于 Qt 的电子点餐系统		(227)
11.1	点餐系统功能说明	(227)
11.2	电子点菜系统设计方案	(227)
11.3	相关技术点拨	(232)
11.4	电子点菜系统的实现与程序代码	(233)
第 12 章 多文本编辑器的设计与实现		(245)
12.1	多文本编辑器系统功能说明	(245)
12.2	多文本编辑器系统设计方案	(245)
12.3	相关技术点拨	(247)
12.4	系统实现与程序代码	(247)
第 13 章 俄罗斯方块		(254)
13.1	俄罗斯方块功能说明	(254)
13.2	俄罗斯方块系统设计方案	(254)
13.3	相关技术点拨	(257)
13.4	俄罗斯方块实现与程序代码	(257)
第 14 章 基于 Qt 的画图板功能的实现		(262)
14.1	画图板功能说明	(262)
14.2	画图板系统设计方案	(262)
14.3	相关技术点拨	(264)

14.4	画图板系统设计与程序代码	(265)
第 15 章	视频监控系统的设计与实现	(274)
15.1	视频监控系统功能说明	(274)
15.2	视频监控系统设计方案	(274)
15.3	相关技术点拨	(276)
15.4	视频监控系统实现与程序代码	(279)
第 16 章	基于 Qt 的扫雷游戏设计与实现	(289)
16.1	扫雷游戏的系统需求说明	(289)
16.2	扫雷游戏的系统设计方案	(289)
16.3	相关技术点拨	(292)
16.4	实现与程序代码	(293)
第 17 章	基于 Qt 的图书管理系统	(296)
17.1	图书管理系统功能说明	(296)
17.2	图书管理系统设计方案	(296)
17.3	相关技术点拨	(298)
17.4	图书管理系统实现与程序代码	(299)
第 18 章	网络版中国象棋	(308)
18.1	中国象棋系统功能说明	(308)
18.2	中国象棋系统设计方案	(308)
18.3	相关技术点拨	(313)
18.4	中国象棋实现与程序代码	(314)
参考文献		(327)

第 1 篇 基础知识

第1章 C++编程基础

C++语言(简称C++)是一门高效实用的程序设计语言,它既可进行过程化程序设计,也可进行面向对象程序设计。C++语言强调对高级抽象的支持,实现了对类的封装、继承和多态,使得程序代码容易维护并可高度重用。

1.1 C++简介

1.1.1 ANSI/ISO 标准

C++是从C语言发展演变而来的一门优秀的程序设计语言。它在保持C语言的简洁、高效和接近汇编语言等特点的基础上,对C语言的类型系统进行了改革和扩充,同时,它还支持面向对象的程序设计。因此,最初的C++称为“带类的C”,1983年正式取名为C++。C++语言的标准化工作从1989年开始,经历了3次修订后,1994年制定了ANSI C++标准草案。之后又经过多次完善,1998年11月被国际标准化组织(ISO)批准为国际标准,2003年10月ISO又发布了第二版的C++标准,成为目前的C++。C++还在不断地发展完善。

1.1.2 名称/标识符

C++中的名称由字符集构成。字符集是构成C++语言的基本元素,包含三个部分:

- ①大小写的英文字母: A~Z, a~z。
- ②数字字符: 0~9。
- ③特殊字符: ! # % ^ & * _ + = - ~ < > \ / ' " ; , : ? () [] { } 。

C++中的标识符指的是由程序员定义的字符集合,如变量名、类名、函数名等,标识符的命名必须满足相应的命名规则,具体如下:

- ①以大写字母、小写字母或下划线(_)开始。
- ②可以由大写字母、小写字母、下划线(_)或数字0~9组成。
- ③大写字母和小写字母代表不同的标识符。即C++程序中的标识符对大小写是敏感的。

④不能是C++中的关键字。关键字是C++预定义的单词,在程序中有着特定的含义,主要包含以下关键字:

```
asm auto bool break case catch char class const const_cast continue default
delete do double dynamic_cast else enum explicit export false float for friend goto
if inline int long mutable namespace new operator private protected public
register reinterpret_cast return short signed sizeof static static_cast struct switch
```

```
template this throw true try typeid typename union unsigned using virtual
void volatile wchar_t while
```

下面,通过几个例子来说明 C++ 中哪些是合法标识符,哪些是非法标识符。

```
Clock _line 3th num_4 auto clock 5-1
```

根据标识符的命名规则可知, Clock, _line, num_4, clock 都是合法的标识符,并且 Clock 和 clock 代表两个不同的标识符; 3th, auto, 5-1 都是非法的标识符。

1.1.3 命名空间

使用命名空间就是为具有相同名称的实体划分其所属范围的区间,从而避免歧义或错误。举例来说,假设一个学校有两位同学都叫刘明,一位是经济系的,另一位是计算机系的。那么在区分的时候就需要说明所要找的到底是经济系的还是计算机系的刘明。

命名空间的语法规则如下:

namespace 命名空间名 {

各种声明(函数声明、类声明、...)

}

例如:

```
namespace ns{
    int fun1();
    float fun2();
    int num;
}
```

在命名空间内部可以直接引用当前命名空间中声明的标识符,如果需要引用其他命名空间的标识符,则需采用如下语法:

命名空间名::标识符名

如果需要调用 ns 命名空间中的 num,则必须使用这种方式:

```
ns:: num;
```

为了更为有效地使用命名空间中的内容,C++ 还提供了两种方式:

using 命名空间名::标识符名;

using namespace 命名空间名;

第一种形式将该标识符暴露在当前作用域内,使得当前作用域中可以直接引用该标识符;第二种形式将指定命名空间内的所有标识符暴露在当前的作用域内,使得在当前作用域中可以直接引用该命名空间内的任何标识符。以 ns 命名空间为例,如果要使用其中的 count 标识符,可以使用

```
using ns:: count;
```

也可以使用

```
using namespace ns;
```

通过这两种方式,都可以直接使用 count 变量,只是使用前种方式的话只能直接使用 ns 中的 count,其他声明不可直接使用,如果还需使用其中的函数 fun1,还需在函数名前添加 ns 区分;而如果使用后面这种方式,ns 中的所有声明都可以在该作用域内直接使用。

允许嵌套使用命名空间，例如：

```
namespace n1 {  
    namespace n2 {  
        int count;  
    }  
}
```

引用其中的标识符，可以通过这种方式实现：`n1::n2::count`。

下面简单介绍两种特殊的命名空间：全局命名空间和匿名命名空间。

全局命名空间是默认的命名空间，如果某个标识符在没有具体说明的显式命名空间，那么该标识符就在全局命名空间中。

匿名命名空间是需要显式说明的没有名字的命名空间，其主要作用是保护该命名空间中的内容不被其他源文件访问。声明方式如下：

```
namespace {  
    各种声明(函数声明、类声明、...)  
}
```

1.1.4 代码注释

代码注释，顾名思义，就是对所写的代码进行解释说明。编译系统在对源程序进行编译时忽略注释部分，注释内容不会对可执行文件的大小产生影响。所以，适当地添加注释，能够提高程序的可读性。这个可读性，不仅是针对编写程序的程序员，更多的是针对团队开发的其他成员以及用户，并对后期维护产生影响。所以，对于一个具有良好编程习惯的程序员而言必须养成及时为代码添加相应注释的习惯。

代码的注释方式有两种：注释行和注释块。

注释行是使用“//”的方式，从“//”开始，直到其所在行尾，所有字符都被作为注释处理。举例如下：

```
int count;    //count is an integer for count
```

通过注释，读者可以知道 `count` 是一个用来统计个数的整型变量。

如果注释比较长，可以选择采用注释块来写，使用“/*”和“*/”括起注释文字。例如：

```
int count;    /* first, count is an integer;  
                second, count is for count; */
```

1.1.5 标准库

C++标准库(C++ Standard Library)是类库和函数的集合，其使用核心语言写成，由C++标准委员会制定，并不断维护更新。

C++强大的功能来源于其丰富的类库及库函数资源。C++标准库的内容在50个标准头文件中定义。在C++开发中，应尽可能地利用标准库来完成。这样做的好处如下：

(1) 节约成本

已经作为标准提供，不必再花费时间、人力重新开发。

(2) 提高质量

标准库都是经过严格测试的,正确性有保证。

(3) 提高效率

使用标准库中的类和函数能有效地提高代码的执行效率。

(4) 良好的编程风格

采用行业中普遍的做法进行开发。

1.2 数据类型

C++ 中的数据类型主要包含两种:基本数据类型和自定义的数据类型(非基本数据类型)。

1.2.1 基本数据类型

C++ 的基本数据类型有 bool(布尔型)、char(字符型)、int(整型)、float(单精度浮点型)以及 double(双精度浮点型)。基本数据类型具体见表 1-1。

表 1-1 基本数据类型表

类型名	长度(字节)	取值范围
bool	1	false, true
char	1	-128 ~ 127
signed char	1	-128 ~ 127
unsigned char	1	0 ~ 255
short (signed short)	2	-32768 ~ 32767
unsigned short	2	0 ~ 65535
int (signed int)	4	-2147483648 ~ 2147483647
unsigned int	4	0 ~ 4294967295
long (signed long)	4	-2147483648 ~ 2147483647
unsigned long	4	0 ~ 4294967295
float	4	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
double	8	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$
long double	8	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$

关键字 signed 和 unsigned 以及关键字 short 和 long 被称为修饰符。类型修饰符 signed 和 unsigned、short 和 long 用于修饰字符型和整型。当用 signed 和 unsigned、short 和 long 修饰 int 整型时, int 可省略。long 还可以修饰 double。

两种浮点类型 float 和 double 除了取值范围不同之外,所取数据的精度也不同, float 可以保存 7 位有效数字, double 可以保存 15 位有效数字。

1.2.2 非基本数据类型

C++中的非基本数据类型主要包括：枚举类型、结构体类型、联合体类型、数组类型、指针类型以及类类型。

1. 枚举类型

将变量的可取值一一列举出来，便构成了一个枚举类型。枚举类型的声明形式为：

enum 枚举类型名 {变量值列表};

例如：一场比赛的结果有胜、负、平局和取消四种，可以通过这种方式声明：

```
enum game { win, lose, tie, cancel };
```

枚举类型应用说明：

——对枚举元素按常量处理，不能对它们赋值。例如，不能写：`win = 0;`

——枚举元素具有默认值，它们依次为：`0, 1, 2, ...`。

——也可以在声明时另行指定枚举元素的值，如：

```
enum game { win = 3, lose, tie = 0, cancel };
```

——枚举值可以进行关系运算。

——整数值不能直接赋给枚举变量，如需要将整数赋值给枚举变量，应进行强制类型转换。

2. 数组类型

数组是具有一定顺序关系的若干相同类型变量的集合体，组成数组的变量称为该数组的元素。

3. 结构体类型

由不同的数据类型构成的一种混合的数据结构，构成结构体的成员的数据类型一般不同，并且在内存中分别占据不同的存储单元。

4. 联合体类型

联合体类型类似于结构体的一种构造类型，与结构体不同的是构成联合体的数据成员共用同一段内存单元。

5. 指针类型

指针类型变量用于存储另一变量的地址，而不能用来存放基本类型的数据。它在内存中占据一个存储单元。

6. 类类型

类是体现面向对象程序设计的最基本特征，也是体现C++与C最大的不同之处。类是一个数据类型，它定义的是一种对象类型，由数据和方法组成，描述了属于该类型的所有对象的性质。