

<<Borland C++4.0 程序设计>>

图书基本信息

书名：<<Borland C++4.0 程序设计>>

13位ISBN编号：9787810125383

10位ISBN编号：7810125389

出版时间：1995-01

出版时间：北京航空航天大学出版社

作者：方旭

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Borland C++4.0 程序设计>>

书籍目录

目录

编者前言

第一部分 集成环境用法

第一章 IDE概述

1.1 安装BorlandC++

1.1.1 对硬件和软件的要求

1.1.2 安装步骤

1.1.3 启动BorlandC++

1.1.4 获取帮助信息

1.2 配置IDE

1.2.1 改变SpeedBars

1.2.2 设置IDE preference

1.2.3 保存IDE 的设置

1.3 使用编辑器

1.3.1 配置IDE的编辑器

1.3.2 语法制导彩色显示 (SyntaxHighlighting)

1.4 使用消息窗口 (Message) 工作

1.5 浏览代码

1.5.1 浏览对象的类

1.5.2 过滤器

1.5.3 查看所列符号的声明

1.5.4 浏览全局符号

1.5.5 在浏览器中使用一般通配符

1.5.6 浏览源代码中的符号

1.6 使用命令行工具

1.6.1 DPMI和命令行工具

1.6.2 内存和MAKESWAP.EXE

1.6.3 运行时间管理器和工具

1.6.4 控制RTM占用的内存

1.7 在IDE中运行其他程序

第二章 IDE编辑器

2.1 编辑器命令

2.2 编辑器激活及文本键入

2.3 字符、字和行的删除

2.4 移动、拷贝文本和块移动

2.5 使用剪贴板

2.6 进一步说明如何移动光标

2.7 字符系列的搜索和替换

2.8 位置标识的设置和搜索

2.9 存储和装入文件

2.10 自动缩进

2.11 磁盘文件中文本块的移入和移出

2.12 对匹配

2.13 其它有关命令

2.14 命令综述

<<Borland C++4.0 程序设计>>

2.15编辑器缺省的修改

2.16编辑器的帮助

第三章 IDE工程管理器

3.1什么是工程管理器

3.2建立一个工程

3.2.1建立一个多任务工程

3.2.2转换旧的工程

3.2.3将工程转化为制作文件 (makefiles)

3.2.4改变工程视图

3.3建立一个工程

3.3.1建立工程的一部分

3.4编辑工程树

3.4.1使用TargetExpert编辑任务属性

3.4.2编辑节点属性

3.4.3增加和删除一个节点

3.4.4增加和删除任务

3.4.5移动节点和任务

3.4.6拷贝节点

3.5使用SourcePool

3.6设置工程选项

3.6.1LocalOverride

3.6.2使用StyleSheet

3.6.3将一个StyleSheet与一个节点相连

3.6.4生成一个StyleSheet

3.6.5编辑StyleSheet

3.6.6共享StyleSheet

3.6.7查看工程中的选项

3.7转换器

3.7.1安装一个转换器

3.7.2使用SpeedMenu中的Special命令

3.7.3安装观察器和工具

第四章 编译

4.1在IDE 中编译

4.1.1使用IDE编译器选项

4.2使用命令行编译器

4.2.1配置文件

4.2.2应答文件

4.2.3选项的优先级规则

4.3编译选项参考

4.4目录选项

4.4.1查找文件算法

4.5编译器宏定义选项

4.6编译器代码生成选项

4.7编译器浮点处理选项

4.8编译器输出选项

4.8.1编译源文件代码选项

4.9编译调试选项

<<Borland C++4.0 程序设计>>

- 4.10预编译头文件选项
- 4.1116位编译处理器选项
- 4.11.116位编译器调用约定选项
- 4.1216位编译存储模式选项
- 4.1316位编译数据段命名选项
- 4.1416位远程数据段命名选项
- 4.1516位编译代码段命名选项
- 4.15.116位编译入口出口代码选项
- 4.1632位编译处理器选项
- 4.1732位编译调用约定选项
- 4.18C++选项成员指针
- 4.19C++兼容性选项
- 4.20C++虚表选项
- 4.21C++模板生成选项
- 4.22C++例外控制选项
- 4.22.1优化选项
- 4.23优化声明选项
- 4.24优化代码长度选项
- 4.25优化速度选项
- 4.26消息选项
- 4.27移植性警告选项
- 4.28ANSI违例警告
- 4.29C++ 错误 (Message|Obsolete C++)
- 4.30潜在的c++ 错误 (Message|Potential C++ Errors)
- 4.31无效C++ 代码 (Messages|Inefficient C++ Coding)
- 4.32潜在错误 (Messages|Potential errors)
- 4.32.1无效代码 (Messages|Inefficient coding)
- 4.33一般错误 (Message|General)
- 4.34 Make选项
- 4.35命令行选项
- 第五章 集成调试器
- 5.1错误的类型
- 5.1.1编译时间错误
- 5.1.2运行时间错误
- 5.1.3逻辑错误
- 5.2生成调试信息
- 5.3指明程序参数
- 5.4控制程序执行
- 5.4.1监视程序的输出
- 5.4.2单步执行程序 (Step over code)
- 5.4.3跟踪执行代码
- 5.4.4单步跳过一段程序代码
- 5.4.5停止程序运行
- 5.4.6重新开始
- 5.5检查变量的值
- 5.5.1什么是表达式
- 5.5.2监视表达式

<<Borland C++4.0 程序设计>>

5.5.3计算和修改表达式

5.5.4检查数据元素

5.5.5检查寄存器的值

5.6使用断点

5.6.1设置断点

5.6.2使用断点工作

5.6.3用户定做断点和执行点

5.7解决一般保护错误

5.8使用 Event Log窗口

5.9调试动态连接库

第六章 连接程序：TLINK

6.1 TLINK基础

6.1.1 TLINK.CFG

6.1.2响应文件

6.1.3用 BCC.EXE使用 TLINK

6.1.4连接库

6.2TLINK选项

6.3模块定义文件参考

6.3.1CODE语句

6.3.2DATA语句

6.3.3DESCRIPTION语句

6.3.4EXETYPE语句

6.3.5EXPORT语句

6.3.6IMPORTS语句

6.3.7LIBRARY语句

6.3.8NAME语句

6.3.9SEGMENTS语句

6.3.10STACKSIZE语句

6.3.11STUB语句

6.3.12缺省的模式定义文件

第二部分 C和C++编程基础

第七章 C和C++基础

7.1C的发展史

7.1.1与其他语言的关系

7.1.2C语言的优势

7.1.3C的弱点

7.1.4编程的规程

7.2ANSIC标准

7.3C++的进化和面向对象的编程

7.4C++的历史

7.4.1使用C++对象来进行流线代码设计

7.4.2对C的提高

7.4.3对C的主要提高

7.5C程序的基本元素

7.5.1程序的五个基本成分

7.5.2第一个C程序

7.5.3第一个C++程序

<<Borland C++4.0 程序设计>>

7.5.4第二个C程序

7.5.5第二个C++程序

7.5.6文件

第八章 数据

8.1标识符

8.2关键字

8.3标准C和C++数据类型

8.3.1字符

8.3.2三种整数

8.3.3unsigned修饰符

8.3.4浮点数

8.3.5枚举型

8.4访问修饰符

8.4.1const修饰符

8.4.2 # define常量

8.4.3Volatile修饰符

8.4.4const和volatile一起使用

8.5pascaldec1、near、far和huge修饰符

8.5.1pascal

8.5.2cdecl

8.5.3nearfar和huge

8.6数据类型转换

8.6.1显示类型转换

8.7存储类型

8.7.1外层的变量说明

8.7.2内层的变量说明

8.7.3变量作用域

8.7.4外层函数说明

8.8操作符

8.8.1位操作符

8.8.2左移和右移

8.8.3增量与减量

8.8.4算术操作符

8.8.5赋值操作符

8.8.6复合赋值操作符

8.8.7关系和逻辑操作符

8.8.8条件操作符

8.8.9逗号操作符

8.9理解操作符优先顺序

8.10 标准C和C++库函数

第九章 控制

9.1条件语句

9.1.1if语句

9.1.2if - else语句

9.1.3嵌套if - else

9.1.4if - else - if语句

9.1.5 ?

<<Borland C++4.0 程序设计>>

条件语句

9.1.6switch语句

9.1.7混合使用if - else - if和Switch语句

9.2循环语句

9.2.1for循环

9.2.2while循环

9.2.3do - while循环

9.2.4break语句

9.2.5使用CodeView

9.2.6continue语句

9.2.7使用TurboDebugger

9.2.8一起使用break和continue

9.2.9exit () 语句

9.2.10atexit () 语句

第十章 编写和使用函数

10.1函数原型和风格

10.1.1构造原型

10.1.2通过值调用和通过引用调用

10.1.3存储类型

10.1.4作用域

10.1.5递归

10.2函数参数

10.2.1形式参数和实际参数

10.2.2void类型的参数

10.2.3字符参数

10.2.4整数参数

10.2.5浮点参数

10.2.6双精度参数

10.2.7数组参数

10.3函数类型

10.3.1void函数类型

10.3.2char函数类型

10.3.3int函数类型

10.3.4long函数类型

10.3.5float函数类型

10.3.6double函数类型

10.4main () 的参数

10.4.1字符串

10.4.2整数

10.4.3浮点数

10.5重要的C++特性

10.5.1直接插入

10.5.2重载

10.5.3省略号 (.....)

10.6作用域规则遇到的问题

10.6.1C++程序中一个未定义的符号

10.6.2使用具有文件作用域的变量

<<Borland C++4.0 程序设计>>

10.6.3用局部变量屏蔽全局变量

10.6.4C++的作用域问题

10.6.5C++作用域限定操作符

第十一章 数组

11.1什么是数组

11.2数组和C

11.3数组说明

11.4数组初始化

11.4.1缺省初始化

11.4.2显示初始化

11.4.3不定长初始化

11.5访问数组元素

11.6计算数组大小 (sizeof ())

11.7数组下标越界

11.8字符串输入输出

11.9多维数组

11.10数组作为函数参数

11.10.1将数组传给C函数

11.10.2数组传给C++函数

11.11字符串函数和字符数组

11.11.1gets () puts () , fgets () , fputs () 和sprintf ()

11.11.2strcpy , strcat , strlen

第三部分 C++面向对象编程基础

第十二章 指针

12.1定义指针变量

12.1.1指针变量说明

12.1.2简单语句使用指针变量

12.1.3指针变量初始化

12.1.4地址操作符的错误使用

12.1.5数组指针

12.1.6指针的指针

12.1.7串的指针

12.1.8指针运算

12.1.9指针运算和数组

12.1.10操作符++和--的有关事项

12.1.11比较指针

12.1.12可移植性

12.1.13指针的大小

12.2函数的指针

12.3动态存储

12.3.1使用空指针

12.4指针和数组

12.4.1串 (字符型的数组)

12.4.2指针数组

12.4.3更多的有关指针的指针的内容

12.4.4串指针的数组

12.5C++引用类型

<<Borland C++4.0 程序设计>>

12.5.1函数返回地址

12.5.2使用TurboDebugger

12.5.3何时使用引用类型

第十三章 C语言中的I/O

13.1流函数

13.1.1打开流

13.1.2输入和输出的重定向

13.1.3改变流缓冲区

13.1.4关闭流

13.2C中的低级输入和输出

13.3字符输入和输出

13.3.1使用getc ()、putc ()、fgetc ()和fputc ()

13.3.2使用getchar ()、putchar ()、fgetchar ()和fputchar ()

13.3.3使用getch ()和putch ()

13.4串输入和输出

13.4.1使用gets ()、puts ()、fgetx ()和fputs ()

13.5整数输入和输出

13.5.1使用getw ()和putw ()

13.6格式化输出

13.6.1使用printf ()和fprintf ()

13.7使用fseek ()、ftell ()和rewind ()

13.7.1使用TurboDebugger

13.8格式化输入

13.8.1使用scanf ()、fscanf ()和sscanf ()

第十四章 C++的输入、输出介绍

14.1使用C++精简I/O

14.1.1cin、cout和cerr

14.1.2抽取 (> >) 和插入 (< <) 操作

14.2从stream.h到iostream.h

14.2.1操作符和成员函数

第十五章 结构、联合和其它数据结构

15.1C和C++结构

15.1.1C和C++结构：语法和规则

15.1.2C++结构：语法和规则的扩展

15.1.3访问结构成员

15.1.4构造一个简单的结构

15.1.5将结构传递给函数

15.1.6构造一个结构数组

15.1.7使用结构指针

15.1.8传递结构数组给函数指针

15.1.9C++中结构的使用

15.1.10对结构的附加操作

15.2联合

15.2.1联合：语法和规则

15.2.2构造一个简单的联合

15.3其它数据类型

15.3.1使用typedef

<<Borland C++4.0 程序设计>>

15.3.2使用枚举类型

第十六章 高级C和C++编程专题

16.1类型兼容

16.1.1ANSIC对于类型兼容的定义

16.1.2什么是同一类型

16.1.3枚举类型

16.1.4数组类型

16.1.5函数类型

16.1.6结构和联合类型

16.1.7指针类型

16.1.8多源文件的相容性

16.2宏

16.2.1定义宏

16.2.2宏和参数

16.2.3宏扩展的问题

16.2.4生成和使用自己的宏

16.2.5编译器预置的宏

16.3高级预处理语句

16.3.1 # ifdef和 # endif

16.3.2 # undef

16.3.3 # ifndef

16.3.4 # if

16.3.5 # else指令

16.3.6 # elif指令

16.3.7 # line指令

16.3.8 # error指令

16.3.9 # pragma指令

16.4条件编译

16.5高级预处理操作

16.5.1 # stringizeoperator (串化操作)

16.5.2连接操作 (# #)

16.5.3字符操作 (# @)

16.6头文件的正确使用

16.7有效地使用头文件

16.8预编译的头文件

16.8.1产生预编译的头文件

16.8.2在PWB (程序员工作台) 上使用预编译头文件

16.9limits.h和float.h

16.10错误处理 : perror ()

16.11存储模式

16.11.1Tiny (微)

16.11.2small (小)

16.11.3medium (中)

16.11.4Compact (紧缩)

16.11.5Large (大)

16.11.6Huge (巨)

16.12动态存储分配 : 链表

<<Borland C++4.0 程序设计>>

- 16.12.1使用链表时需要考虑的事项
- 16.12.2一个简单的链表
- 第十七章 面向对象的编程介绍
- 17.1并非是新内容
- 17.2传统结构化设计
- 17.3面向对象的程序设计
- 17.3.1C++和面向对象设计
- 17.4面向对象的术语
- 17.4.1封装
- 17.4.2类的层次性
- 17.5认识C++类
- 17.5.1一个作为私有类的结构
- 17.5.2C++类的语法规则
- 17.5.3一个简单的C++类
- 第十八章 C++类
- 18.1附加的类特性
- 18.1.1一个简单的类
- 18.1.2嵌套类
- 18.1.3构造函数和析构函数
- 18.1.4重载类的成员函数
- 18.1.5使用友元函数来存取私有类变量
- 18.1.6使用出is指针
- 18.2使用操作符重载
- 18.2.1操作符重载和函数调用
- 18.2.2重载的语法
- 18.3派生类
- 18.3.1派生类的语法
- 18.3.2产生派生类
- 第十九章 C++的I/O
- 19.1枚举类型
- 19.2引用变量
- 19.3缺省参数
- 19.4memset ()
- 19.5格式化的输出
- 19.6C/C++I/O选择
- 19.7iostream类列表
- 19.7.1输入流类
- 19.7.2输出流类
- 19.7.3缓冲流类
- 19.7.4串流类
- 19.8二进制文件
- 19.9使用extern “ C ” 来结合C和C++代码
- 19.10编写自己的操纵符
- 19.10.1无参数的操纵符
- 19.10.2带有一个参数的操作器
- 19.10.3多参数的操纵符
- 第二十章 基于面向对象环境的工作

<<Borland C++4.0 程序设计>>

- 20.1C++中一个面向的对象的栈
 - 20.2C++中面向对象的连接表
 - 20.2.1父类的产生
 - 20.2.2派生子类
 - 20.2.3使用友元类
 - 20.2.4测试一个完整的程序
 - 20.2.5连接表输出
 - 20.3面向对象C++的其它内容
- 第四部分 DOS图形、系统访问库及混合语言接口
- 第二十一章 屏幕文本和图形程序设计
- 21.1图形系统和要素
 - 21.1.1视频模式
 - 21.1.2窗口和视区
 - 21.1.3在文本模式下编程
 - 21.1.4在图形模式下编程
 - 21.2BorlandC++图形程序设计
 - 21.2.1基本正文模式函数
 - 21.2.2BorlandC++的图形子系统简介
 - 21.3IBM/PC的文本方式
 - 21.3.1PC显示器适配器和屏幕
 - 21.3.2视频缓冲区I/O
 - 21.4控制文本屏幕
 - 21.4.1使用文本方式控制函数
 - 21.4.2使用直接控制台I/O以获得高性能
 - 21.5使用窗口函数
 - 21.6了解IBM PC的图形方式
 - 21.6.1像素点与调色板
 - 21.6.2控制图形屏幕
 - 21.7介绍BGI图形库
 - 21.7.1使用画图和填充函数
 - 21.7.2控制屏幕和视口
 - 21.8在图形方式下显示文本
 - 21.8.1BGI字库
 - 21.8.2使用图形方式下的文本函数
- 第二十二章 更具威力的编程：选择重要的C和C++库
- 22.1TurboC和C++头文件
 - 22.2标准库函数 (stdlib.h)
 - 22.2.1实现数据转换
 - 22.2.2实现查找和排序
 - 22.2.3混合操作
 - 22.3字符函数 (ctype.h)
 - 22.3.1查测是否是字母/数字、字母、ASCII值
 - 22.3.2检测是否是控制、空白和标点字符
 - 22.3.3转换为ASCII码、小写字母、大写字母
 - 22.4字符串函数 (string.h)
 - 22.4.1内存函数
 - 22.4.2使用字符串函数

<<Borland C++4.0 程序设计>>

22.5数学函数 (math.h)

22.5.1建立三角函数值的表

22.6时间函数 (time.h)

22.6.1时间和日期的结构和语法

22.7依赖于系统的函数

22.7.1bios.h头文件

22.7.2dos.h头文件

第二十三章 BorlandC++和汇编语言的混合编程

23.1直接插入汇编语言代码

23.1.1产生音响

23.1.2并行端口作为通用I/O端口

23.1.3控制LED发光管

23.2创立C++和汇编语言模块

23.2.1传递函数参数

23.2.2传递不同数据类型的参数

23.2.3简单的C和汇编语言连接

23.2.4用C和汇编语言连接硬件接口

23.2.5由C向汇编语言传递数组

第五部分 设计Windows应用程序界面

第二十四章 Windows编程概述

24.1Windows程序设计语言

24.1.1Windows环境快速浏览

24.1.2使用Windows的优越性

24.1.3新的Windows可执行文件的格式

24.2Windows编程：概念和词汇

24.2.1Windows的窗口

24.2.2Windows布局

24.2.3C++中的Windows类

24.2.4OOP及Windows

24.2.5发送和接收Windows消息

24.2.6访问Windows函数

24.2.7Windows头文件：Windows.h

24.2.8Windows应用程序的组成成分

24.3BorlandC++Windows工具

24.3.1使用make文件

24.3.2创建资源

24.4使用资源工具

24.4.1BRCC.EXE：资源编译器

24.4.2RLINK：资源连结器

24.4.3BRC.EXE资源外壳

24.5启动

24.5.1安装ResourceWorkshop

24.5.2启动ResourceWorkshop

24.5.3退出ResourceWorkshop

24.5.4获得帮助

第二十五章 Windows程序设计技术

25.1应用程序框架

<<Borland C++4.0 程序设计>>

- 25.1.1 有效地使用句柄
- 25.1.2 考察Windows应用程序中的成分
- 25.2 一个简单的应用程序和示例
 - 25.2.1 画椭圆
 - 25.2.2 画弦
 - 25.2.3 画饼图瓣
 - 25.2.4 画矩形
- 25.3 使用SWA开发一个正弦波应用程序
- 25.4 创建Windows饼图应用程序
 - 25.4.1 PIE和PIE.DEF文件
 - 25.4.2 pie.h头文件
 - 25.4.3 PIE.RC资源文件
 - 25.4.4 PIE.C程序
- 第六部分 附录
 - 附录A 扩展ASCII表
 - 附录B DOS10H, 21H和33H中断参数
 - B.1 由BIOS10H中断获得屏幕控制
 - B.2 CRT的接口控制
 - B.3 字符处理
 - B.4 图形接口
 - B.5 ASCII电传类型输入
 - B.6 DOS21H中断
 - B.7 通过中断33H进行存取鼠标控制函数
 - 附录C BorlandC++4.0实用程序
 - C.1 运行MAKE
 - C.1.1 MAKE基础
 - C.1.2 使用makefile
 - C.1.3 显式和隐式规则
 - C.1.4 使用MAKE宏
 - C.1.5 使用MAKE指令
 - C.2 使用库管理程序
 - C.2.1 使用IMPLIB: 导入库
 - C.2.2 使用IMPDEF: 模块定义文件管理器
 - C.2.3 使用TLIB: Turbo库
 - C.2.4 例子
 - C.3 运行WinSpector
 - C.3.1 使用WinSpector
 - C.3.2 处理WinSpector数据
 - C.3.3 其它WinSpector工具
 - C.4 WinSight
 - C.4.1 开始启动
 - C.4.2 选择视窗
 - C.4.3 ClassList
 - C.4.4 WindowTree
 - C.4.5 选择跟踪消息

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>