



软件项目开发实例丛书



实例源代码
免费下载

Visual C++

软件项目开发实例

尹立民 王兴东 等编著

详述基于**Visual C++**的软件项目
设计总体思路，通过完整的通用实例
讲解软件项目开发的过程与方法

- MyDraw绘图软件
- 键盘奏鸣曲软件
- 网络数据管理软件
- .NET特性演示软件



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



软件项目开发实例丛书

Visual C++

软件项目开发实例

尹立民 王兴东 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

SJS180/02

内 容 提 要

本书完整地介绍了几个基于 Visual C++ 的软件项目开发实例, 包括 MyDraw 绘图软件、键盘奏鸣曲软件、网络实时数据管理软件和.NET 特性演示软件。本书深入剖析了软件项目的开发思路、方法和技巧, 详细地讲述了基于 Visual C++ 的软件项目开发的全过程。对于系统设计、开发环境选择、系统功能模块设计、各个功能模块的创建、系统的编译和发行等软件项目开发过程进行了详细的讲解。为方便广大读者使用, 并最大限度地节约读者的购书成本, 本书实例源代码可以免费从网上下载。

本书注重工程实践性, 实用性强, 可供广大程序员和编程爱好者进行类似软件项目开发时学习参考, 也可相关专业大学生毕业设计提供经典项目案例借鉴。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

Visual C++ 软件项目开发实例 / 尹立民等编著. —北京: 电子工业出版社, 2004.11

(软件项目开发实例丛书)

ISBN 7-121-00488-7

I. V... II. 尹... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 109145 号

责任编辑: 张瑞喜

印 刷: 北京市天竺颖华印刷厂

出版发行: 电子工业出版社出版

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 19.75 字数: 454 千字

印 次: 2004 年 11 月第 1 次印刷

印 数: 6000 册 定价: 28.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。
联系电话: (010)68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

丛 书 前 言

《软件项目开发实例丛书》由一批长期从事相关软件项目开发的资深软件工程师、项目经理和专业老师共同编写。丛书深入剖析了软件项目的开发思路、方法和技巧，结合具体的项目详细地讲述了软件项目开发的全过程。对系统设计、开发环境选择、系统功能模块设计、功能模块的创建、系统的编译和发行等软件项目开发过程进行了详细的讲解。本套丛书注重工程实践性，实用性强，可以帮助读者深入掌握软件项目开发的方法和过程，对于从事各类软件项目的开发人员有一定参考价值，对广大计算机及相关专业大学生进行毕业设计、自主开发小型商业项目具有很好的指导意义。

本套丛书共有以下 4 种，分别讲解了 4 至 5 个完整的小型通用软件项目实例。

《Visual Basic 软件项目开发实例》

- ◆企业信使软件
- ◆小秘书办公软件
- ◆网络图像认证软件
- ◆MDBManager 数据库管理软件

《Visual C++软件项目开发实例》

- ◆MyDraw 绘图软件
- ◆键盘奏鸣曲软件
- ◆网络数据管理软件
- ◆.NET 特性演示软件

《Visual FoxPro 软件项目开发实例》

- ◆名片管理软件
- ◆网络图书信息浏览系统
- ◆API 特效演示软件
- ◆教学信息管理软件

《Delphi 软件项目开发实例》

- ◆网络数据采集与存储(NetData)软件
- ◆TestForNetData 测试软件
- ◆图书管理系统
- ◆远程借阅系统
- ◆桌面精灵软件

每本书均在第 1 章对一般项目开发的总体设计思路进行了概括的讲解，旨在使学习者能够提纲挈领地把握一个项目开发的全过程。不论使用哪种开发工具，首先都要掌握软件项目开发的总体思路，这一点是相通的。然后通过几个完整的商用项目实例，具体讲述了项目开发的全过程。这些实例都具有很好的通用性和典型性。

本套丛书在策划的过程中充分考虑到了在软件开发市场上软件版本的适用性和实际需要。每本书在开发工具版本的选择上，是从系统开发所需要的软件角度出发来考虑的，不会因为软件版本的升级而盲目跟风。例如，在《Visual Basic 软件项目开发实例》一书中，使用 Visual Basic 6.0 开发企业信使软件，而采用 Visual Basic .NET 开发网络图像认证软件项目(因为这样才能充分发挥.NET 平台的网络支持特性)等。

在讲解开发实例的过程中，增加了系统设计和测试发行两个板块，特别是测试发行板块，这是决定软件质量尤其是商业软件非常重要的一个步骤。这样，实例项目的开发流程就非常完整了。

虽然每本书都是针对几个具体的软件项目实例来设计的，但是它却讲述了软件项目设计中非常具有通用性的内容和技术，如数据环境的设计、软件项目的模块化设计、软件项目的测试等，所以它不仅对进行类似项目设计的读者有很直观的指导作用，对其他软件项目设计的读者来说也有很好的参考价值。

本套丛书特别适合于对软件功能有初步了解，但需要提高实际动手能力的读者阅读。

本套丛书的作者将通过编读交流网站<http://phei.xicp.net>，为广大读者提供在线指导和更多的超值网络服务。

编者

2004 年 10 月

前 言

Visual C++是运行于 Windows 平台上的可视化集成开发环境,它是 Microsoft 公司开发的 Microsoft Visual Studio 系列的一部分。Visual C++集程序的界面设计、代码编辑、编译、连接和调试等功能于一体,为编程人员提供了一个方便而又完整的开发界面和许多有效的辅助开发工具。

本书共 5 章,主要包括以下内容。

第 1 章介绍了软件项目开发的相关内容,如软件开发的过程、系统设计的步骤、软件开发工具的选择、程序测试,并且提供了一个 Visual C++编程规范的示例。通过本章的学习,可以从系统的角度去理解软件项目,增强软件项目开发过程中的全局观。

第 2 章介绍了两个版本的 MyDraw 绘图软件的开发过程。在工程软件中,经常需要绘图表示工业现场的布置图,而且要求绘图软件能够将绘图元素对象化,这样有利于修改。MyDraw 软件就可以满足这样的要求,它是一个用于对象化绘图的软件,用于辅助客户完成常用图形的绘制、存储等操作的软件。

第 3 章介绍了键盘奏鸣曲软件。在许多系统中,出于安全或其他原因,常常要求随时对键盘进行监控,利用 Hook(即钩子)技术编写的应用程序能够很好地达到这个目的。本软件制作了一个键盘奏鸣曲软件,即在软件运行过程中,用户在键盘上的按键操作就会播放声音文件。对本软件稍加扩展就可以变成一个完整的键盘监控软件。

第 4 章介绍了网络数据管理软件的开发过程。目前网络数据采集在工程项目中极为普遍。例如,小区或大厦环境和安全检测,使用各种感应器和摄像装置采集数据,将采集的数据输入到计算机里,而计算机实时地采集从网络中传输的数据并进行处理,并以一种易于用户观察的方式显示出来,便于用户管理和控制。网络数据管理软件是一个集网络技术与数据库技术于一体的软件,用于完成网络数据的实时采集并且动态存储,形成实时数据和历史数据。本软件的开发过程,对于其他需要从网络采集数据并进行处理的软件的开发具有比较普遍的借鉴意义。

第 5 章介绍了 .NET 特性演示软件,它演示了 Visual C++ .NET 三个方面的新特性——Windows 窗体、组件和数据库。它的大部分功能采用 Visual Studio .NET 的组件自动完成,开发人员只需要添加少量的代码就可以完成所需要的功能,代码量非常少,大大减轻了程序设计人员的工作量。

本书的实例源代码可从网上下载,具体说明参见附录 A。

本书的作者将通过编读交流网站<http://phei.xicp.net>,为广大读者提供在线指导和更多的超值网络服务,具体说明参见附录 B。

参加本书编写的主要人员有：尹立民、王兴东、赵万军、关晓平、管海涛、王喜章、冯欢、王春侠、刘仙、张洁、弓胜民、高卫平、金海涛、姜竞秋、董丽等。

由于作者水平有限，书中难免存在一些不足之处，恳请读者批评指正。

作者的电子邮箱：yinlimin@sina.com

作者

2004年10月

目 录

第 1 章 基于 Visual C++ 的软件项目设计	1
1.1 软件开发过程	1
1.1.1 项目计划阶段	1
1.1.2 需求分析阶段	3
1.1.3 软件开发阶段	4
1.1.4 测试阶段	4
1.2 系统设计的步骤	4
1.3 软件开发工具的选择	5
1.4 测试与改错	6
1.4.1 对测试的理解	6
1.4.2 测试方式	6
1.4.3 测试的主要内容与常用方法	7
1.4.4 改错	8
1.5 Visual C++ 编程规范的制定	8
1.5.1 基本要求	8
1.5.2 可读性要求	8
1.5.3 结构化要求	9
1.5.4 正确性与容错性要求	9
1.5.5 编程规范示例	9
1.6 软件文档	11
第 2 章 MyDraw 绘图软件	13
2.1 MyDraw 绘图软件的介绍	13
2.1.1 MyDraw 绘图软件的特色	13
2.1.2 MyDraw 绘图软件的功能	14
2.1.3 MyDraw 软件的客户定位	16
2.2 系统设计与选型	16
2.2.1 软件工具的选型	16

2.2.2	MyDraw 绘图软件 1.0 版模块分工设计	16
2.3	MyDraw 绘图软件 1.0 版功能模块的创建	18
2.3.1	CLine 类记录直线信息	18
2.3.2	CArc 类记录弧线信息	21
2.3.3	CEllipse 类记录椭圆信息	23
2.3.4	CRectangle 类记录矩形信息	25
2.3.5	CText 类记录文本信息	28
2.3.6	CLineProperties 类显示和修改 CLine 和 CArc 类信息	31
2.3.7	CEllipseProperties 类显示和修改 CEllipse 和 CRectangle 类信息	34
2.3.8	CTextProperties 类显示和修改 CText 类信息	38
2.3.9	CMyDrawView 类实现绘图操作	41
2.3.10	CMyDrawDoc 类实现文档存取操作	73
2.4	测试 MyDraw 绘图软件 1.0 版	78
2.5	对 MyDraw 绘图软件 1.0 版的修改设计	81
2.6	MyDraw 绘图软件 2.0 版功能模块的创建	84
2.6.1	CDrawElement 类记录绘图元素信息	84
2.6.2	对 CLine 类的修改	86
2.6.3	对 CArc 类的修改	89
2.6.4	对 CEllipse 类的修改	92
2.6.5	对 CRectangle 类的修改	96
2.6.6	对 CText 类的修改	99
2.6.7	对 CMyDrawView 类的修改	103

第 3 章 键盘奏鸣曲软件 113

3.1	键盘奏鸣曲软件的介绍	113
3.1.1	键盘奏鸣曲软件的特色	113
3.1.2	键盘奏鸣曲软件的功能	113
3.1.3	键盘奏鸣曲软件的客户定位	113
3.2	系统设计与选型	114
3.2.1	软件工具的选型	114
3.2.2	功能分析	114
3.2.3	模块分工设计	114
3.2.4	技术问题分析	114
3.3	功能模块的创建	119
3.3.1	设计 DLL 模块	119
3.3.2	创建钩子可执行程序	129
3.4	运行程序	130

第 4 章 网络数据管理软件131

4.1 网络数据管理软件的介绍.....	131
4.1.1 网络数据管理软件的特色.....	131
4.1.2 网络数据管理软件的功能.....	132
4.2 服务器端系统设计与选型.....	133
4.2.1 软件工具的选型.....	133
4.2.2 功能分析.....	133
4.2.3 模块分工设计.....	133
4.3 网络协议的设计.....	135
4.4 服务器端数据环境的设计.....	136
4.5 服务器端各个功能模块的创建.....	137
4.5.1 程序的初始化.....	137
4.5.2 内存链表类模块 ControlUnit	137
4.5.3 通信类模块.....	204
4.5.4 界面左侧视类 CLeftView.....	218
4.5.5 界面右侧视类 CDataManagerView.....	228
4.5.6 设置“段”对话框.....	242
4.5.7 设置“单元”对话框.....	242
4.5.8 设置“点”对话框.....	243
4.6 客户端软件的介绍.....	243
4.6.1 客户端软件的特色.....	243
4.6.2 客户端软件的功能.....	244
4.7 客户端的系统设计.....	244
4.7.1 模块分工设计.....	244
4.7.2 ini 配置文件的设计	245
4.8 各个功能模块的创建.....	246
4.8.1 内存数据结构.....	246
4.8.2 客户连接 Socket 类 CClientSocket	247
4.8.3 “界面”对话框.....	252
4.9 软件运行.....	267
4.9.1 运行服务器.....	267
4.9.2 客户端的运行.....	268

第 5 章 .NET 特性演示软件271

5.1 .NET 特性演示软件的介绍.....	271
5.1.1 .NET 特性演示软件的特色.....	271

5.1.2 .NET 特性演示软件的功能 271

5.2 系统设计 273

5.2.1 功能分析 273

5.2.2 模块分工设计 273

5.2.3 数据环境的设计 273

5.3 功能模块的创建 274

5.3.1 Windows 窗体功能演示 274

5.3.2 组件功能演示 281

5.3.3 数据库特性演示 284

5.4 运行程序 299

附录 A 实例源代码下载说明 305

附录 B 书网互动学习说明 306

第1章 基于 Visual C++ 的软件项目设计

本章主要从软件工程的角​​度介绍了软件项目开发的各个过程及各个过程中的主要任务，主要包括需求分析、系统设计、软件开发工具的选择、测试与改错、Visual C++编程规范、软件文档等内容。

通过深入细致的需求分析产生“需求分析报告”后，进入软件开发阶段。首先需要进行系统设计，而将系统分层是简化系统的好方法，针对每一层可以设计一个或多个模块，每个模块完成相对独立的功能。

软件开发工具的选择主要决定于两个因素：所开发系统的最终用户和开发人员。Visual C++以其极大的软件自由度，较少的封装和充分的功能暴露成为 Microsoft 对其操作系统功能的“权威”封装，至今在 Windows 系统级开发中占据主流地位。

很多人认为软件开发阶段是最重要的，相比之下，测试阶段就无足轻重了。实际上，测试也是一个很重要的阶段。测试的目的是为了发现尽可能多的缺陷，并期望通过改错来消除缺陷，以期提高软件的质量。一般首先在模块内进行单元测试，通常由开发人员完成，然后各个模块组合起来进行连调，再对整个软件项目进行综合测试，包括正确性测试、容错性测试、性能与效率测试、易用性测试、文档测试。

1.1 软件开发过程

软件项目开发的总体流程如图 1-1 所示。

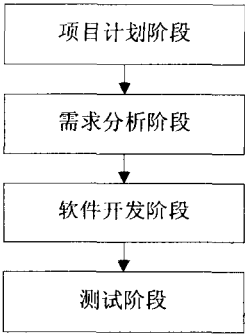


图 1-1 软件项目开发的总体流程

1.1.1 项目计划阶段

项目计划草案和风险管理计划是项目开发的第一步。当有一个商业机会后，制定初步商业计划书来完成项目的计划草案，确定、分析项目风险并确定其优先级，还要制定风险解决方案等。当确定了开发的项目之后，则需要制定软件开发计划、确定人员组织结构等。



1. 项目计划草案

项目计划草案应包括产品简介、产品目标及功能说明、开发所需的资源、开发时间等。

2. 风险管理计划

风险管理计划是将有可能出错或现在还不能确定的东西列出来，并制定出相应的解决方案。风险发现得越早对项目越有利。

3. 软件开发计划

软件开发计划的目的是收集控制项目时所需的所有信息，项目经理根据项目计划来安排资源需求，并根据时间表跟踪项目进度。项目团队的成员根据项目计划以了解他们的工作任务、工作时间以及他们所依赖的其他活动。

可将计划分成总体计划和详细计划，总体计划中每个任务为一个里程碑，详细计划中必须将任务落实到个人。

软件开发计划还应包括产品的验收标准。

4. 人员组织结构

(1) 项目经理的职责有以下几项。

- 制定产品的目标。
- 制定各个工作的详细任务表，跟踪这些任务的执行情况，进行控制。
- 组织会议对程序进行评审。
- 根据具体的情况，对各种不同方案进行取舍并做出决定。
- 协调各项目参与人员之间的关系。

(2) 对项目经理的要求：对产品有激情，具有领导才能；对问题能正确而迅速地做出决定；能充分利用各种渠道和方法来解决问题；能跟踪任务，有很好的日程观念。

(3) 系统分析员的职责有以下两项。

- 了解用户需求，写出软件需求分析。
- 建立用户界面原型。

系统分析员应该善于协调，具有良好的沟通技巧，并且具备业务和技术领域的知识。

程序员的职责是按项目的要求进行编码和单元测试。要求具有良好的编程技能和测试技术。

测试员的职责是执行测试，描述测试结果，提出问题解决方案。要求了解被测试的系统，具备诊断和解决问题的技能，以及编程技能等。

根据每个人的特长来担任其中的一个或多个角色，最好是每个人都能参与到设计或编码工作中，这样有利于建立起个人的系统全局观。

5. 过程控制计划

过程控制计划的目的是收集项目计划正常执行所需的所有信息，用来指导项目进度的监控、计划的调整，以确保项目按时完成。

1.1.2 需求分析阶段

需求分析阶段的目的是系统开发方与用户方达成一致。

软件需求规约：详细说明系统将要实现的所有功能。

用户界面原型：可以有三种表示方法，即图纸(在纸上)、位图(绘图工具)和可执行文件(交互式)。

需求分析阶段的工作必须做得到位，否则软件很可能一而再、再而三地根据用户新的需求更改，而这些新的需求往往都是当初没有提及的，问题就在于需求分析阶段双方没有进行充分的交流。弄清用户真正的需求(包括细节)，正是软件开发成功的关键所在。

当与客户交流的时候，谈话往往是这样的。

经理：“我们要建立一套完整的商业管理软件系统，包括商品的进、销、调、存管理，是总部—门店的连锁经营模式。通过通信手段门店自动订货，供应商自动结算，卖场通过扫描条码实现销售，管理人员能够随时查询门店商品销售和库存情况。另外，我们也得为政府部门提供关于商品营运的报告。”

分析员：“我已经明白这个项目的大体结构框架，这非常重要，但在制定计划之前，我们必须收集一些需求。”

经理：“我不是刚告诉你我的需求吗？”

实际上，经理只说明了整个项目的概念和目标。这些高层次的业务需求不足以提供开发的内容和时间，需要与将要实际使用系统的业务人员进行讨论，然后才能真正明白达到业务目标所需功能和用户要求。

如果只是凭空猜想用户的要求，结果总是不会令人满意的。因为软件开发人员并不是采购专家、营运专家或是财务专家，不能真正明白这个企业内部运营所需要做的事情。

在项目开发中，所有的项目风险承担者都对需求分析阶段感兴趣。这里所指的风险承担者包括客户方的项目负责人和用户，开发方的需求分析人员和项目管理者。做好这部分工作，就完成了开发优秀的软件产品的第一步，同时也会令客户满意。若处理不好，则会导致误解、挫折、障碍，以及潜在的质量和业务价值上的威胁。由此可见——需求分析奠定了软件工程和项目管理的基础。

下面分析一下需求的具体内容。

(1) 业务需求：反映了组织机构或客户对系统、产品高层次的目标要求，通常在项目定义与范围文档中予以说明。

(2) 用户需求：描述了用户使用产品必须完成的任务，这在使用实例或方案的脚本中予以说明。

(3) 功能需求：定义了开发人员必须实现的软件功能，使用户利用系统能够完成他们的任务，从而满足业务需求。

(4) 非功能性的需求：描述了系统展现给用户的行为和执行的操作等，它包括产品必须遵从的标准、规范和约束，以及操作界面的具体细节和构造上的限制。

(5) 需求分析报告：报告所说明的功能需求充分描述了软件系统所应具有的外部行为。“需求分析报告”在开发、测试、质量保证、项目管理以及相关项目功能中起着重要作用。



前面提到的经理通常阐明产品的高层次概念和主要业务内容，为后继工作建立了一个指导性的框架。其他任何说明都应遵循“业务需求”的规定，然而“业务需求”并不能为开发人员提供开发所需的许多细节说明。

下一个层次需求——用户需求，必须从使用产品的用户处收集。因此，这些用户构成了另一种软件客户，他们清楚要使用该产品完成什么任务和一些非功能性的特性需求。例如：程序的易用性、健壮性和可靠性，而这些特性将使用户更好地接受具有该特点的软件产品。

用户需求来自产品的真正使用者，必须让实际用户参与到收集需求的过程中。如果不这样做，产品很可能会因缺乏足够的信息而遗留不少隐患。

开发人员与实际用户交流需要好的方法。首先分析人员要使用符合用户语言习惯的表达。

需求讨论应集中于业务需求和任务，因此要使用术语。客户应将有关术语(例如：采价、印花商品等采购术语)教给分析人员，而客户不一定要懂得计算机行业的术语。

分析人员要了解用户的业务及目标。只有分析人员更好地了解用户的业务，才能使产品更好地满足需要。这将有助于开发人员设计出真正满足用户需要并达到期望的优秀软件。

分析人员应将从客户那里获得的所有信息进行整理，以区分业务需求及规范、功能需求、质量目标、解决方法和其他信息。通过这些分析，用户就能得到一份“需求分析报告”，这份报告使开发人员和客户之间针对要开发的产品内容达成协议。

报告应以一种用户认为易于翻阅和理解的方式组织编写。用户要评审此报告，以确保报告内容准确完整地表达其需求。一份高质量的“需求分析报告”有助于开发人员开发出真正需要的产品。

1.1.3 软件开发阶段

本阶段从物理上实现目标系统，包括的内容如下。

- (1) 软件框架设计：说明软件的组织结构及运行环境。
- (2) 数据环境设计：定义数据之间的关联及含义。
- (3) 编码和单元测试：按照设计文档进行编码，每完成一个模块应进行单元测试。
- (4) 集成系统：按软件组织结构的要求将各个子系统组合起来。

1.1.4 测试阶段

测试的目的是在发布之前找出并修正程序的错误。

1.2 系统设计的步骤

将系统分层是简化系统的好方法，而且已经得到了很好的证实，如 OSI(Open System Interconnection)七层参考模型，就是一种通信协议的七层抽象参考模型。其中每一层执行某一特定任务。该模型的目的是使各种硬件在相同的层次上相互通信。这七层是：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。分层的思路是将系统按功能职责进行划分，将同一类职责的功能抽象为一层。下面以一个信息系统的设计来说明如何进行分层结构设计。

某系统中包含三层结构：表示层、业务层和数据层。

与传统的两层结构相比，它的最大特征是将业务层独立出来，从而提高了业务层的可

复用性。在两层结构中，用户界面和业务处理流程放在一起，因此无法直接复用业务处理的相关功能，也无法将业务处理功能进行灵活的部署。在三层结构中，表示层只处理用户界面相关的功能，业务层专心处理业务流程，可以对业务层进行灵活的部署，开发时也便于业务处理的开发和用户界面的开发同时进行。

OSI 中要求高层只能调用它的下一层提供的接口，我们设计接口时也尽量遵守这样的约束。

数据层在业务层中是可见的，业务层在表示层中也是可见的，反之则是不可见。为什么在业务层中不能直接访问表示层呢？因为业务层要相对独立，它不能依赖于任何表示层，以至于一个业务层可以对应多个表示层。业务层可以间接与表示层通信，这种通信方式根据实际需要来确定。

针对每一层可以设计一个或多个模块，每个模块完成相对独立的功能。

如表示层中用户界面模块的功能如下。

- (1) 与用户的交互，接收用户的各种输入以及输出各种提示信息或处理结果。
- (2) 对于输入的数据进行数据校验，过滤非法数据。
- (3) 向业务层发送处理请求。

业务层中业务处理模块的功能如下。

- (1) 实现各种业务处理逻辑或处理算法。
- (2) 验证请求者的权限。
- (3) 向数据层发送数据操作的请求。
- (4) 向用户层返回处理结果。

数据层中数据数据访问模块的功能如下。

- (1) 实现数据的读取与存储操作。
- (2) 实现事务处理。

1.3 软件开发工具的选择

软件开发已经逐步从原来的手工作坊式发展到了软件工程的阶段，软件开发本身也在不断发展。开发工具的选择，已经成为软件开发成功的要素之一。

开发工具的选择主要决定于两个因素：系统的最终用户和开发人员。最终用户需求是一切软件的来源和归宿，也是影响开发工具的决定性因素；开发人员的爱好、习惯、经验也影响着开发工具的选择。

程序的最终使用群体是软件开发的服务对象，他们影响着开发工具的选择。从计算机使用的程度分类，最终的使用者可以分为 IT 人员、各行业的专业人员以及普通用户。由于使用者的不同，所以对于软件的需求就不会相同。IT 人员自然需要更多的功能、更自由的设置以及二次开发空间；行业用户往往需要一个整体的解决方案，从而提升其整体竞争力；普通用户显然要求更方便简单地使用软件。

1. 软件自由度

为了扩展软件的自由度，较少的封装和充分的功能暴露是必然的趋势。Visual C++ 成为 Microsoft 对其操作系统功能的“权威”封装，至今在 Windows 系统级开发中占据着主



流地位; Delphi 扩充的 Pascal 语法, 提供了 RAD(Rapid Application Development)的支持, 而且还可以通过很容易修改 VCL(Visual Component Library)得到新的组件, 具有很强的灵活性; Visual Basic 语法结构较为简单, 同样可以非常轻松地调用 API 函数扩展自身的功能; Visual FoxPro 也可以通过 ODBC 和 ASP 等技术很容易地与其他程序交互。

2. 针对性各有特色

在针对性上, 各个工具都具备各自的优势。在单机应用上, Visual FoxPro 具有全球最快的数据访问引擎; 而使用 Visual Basic 或 Visual C++再使用 SQL Server, 就能在性能、开发效率、稳定性上都有所保证。

3. 开发者的偏爱

开发工具是给开发者用的, 开发人员是这些工具的用户。不同的开发人员对工具的偏爱也不同。Pascal 程序员一般都会钟爱 Delphi; Windows 的 C++程序员则会选择 Visual C++; Basic 程序员会更加喜欢 Visual Basic; 而在早期大量应用 FoxPro 的程序员会偏爱采用 Visual FoxPro。

1.4 测试与改错

测试的目的是为了发现尽可能多的缺陷, 并期望通过改错把缺陷统统消灭, 以获得更高的软件质量。

1.4.1 对测试的理解

以下是关于测试的几个重要观念。

1. 测试的目的

测试的目的是为了发现尽可能多的缺陷。

这里缺陷是一种泛称, 它可以指功能的错误, 也可以指性能的低下, 易用性差等。测试总是先假设程序中存在缺陷, 再通过执行程序来发现并最终改正缺陷。

2. 测试与质量的关系

测试有助于提高软件的质量, 但是提高软件的质量不能依赖于测试。归根结底, 软件的高质量是设计出来的, 而不是靠测试修补出来的。

1.4.2 测试方式

内部对产品的测试(称为 α 测试), 需要开发人员与独立的测试小组共同参与。开发人员应该执行“白盒测试”, 即测试源程序的逻辑结构以及实现细节(“白盒”是指看得见程序的内部结构)。而独立测试小组应该执行“黑盒测试”, 即按照规格说明来测试程序是否符合要求(“黑盒”是指看不见程序的内部结构)。比如在测试一个模块时, “白盒”测试方法要对模块的所有代码进行单步跟踪测试。而“黑盒”测试方法只需测试模块的接口是否符合要求, 它关心程序的外部表现而不是内部的实现细节。

这里要强调的是, α 测试不能依赖于开发人员或者测试小组中的任何一方, 必须是双方