高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

Delphi 软件开发技术

张世明 编

免费提供教学相关资料

人民邮电出版社
POSTS & TELECOM PRESS

高职高专计算机技能型紧缺人才培养规划教材
计算机软件技术专业

Delphi 软件开发技术

张世明 编

人民邮电出版社

图书在版编目（CIP）数据

Delphi 软件开发技术 / 张世明编. —北京：人民邮电出版社，2005.7

ISBN 7-115-13316-6

Ⅰ. D... Ⅱ. 张... Ⅲ. 软件工具—程序设计 Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字（2005）第 070029 号

内 容 提 要

为了帮助学生将 Delphi 学通、学透，使学生真正能用 Delphi 开发应用程序，本套教材将 Delphi 作为一个模块按开课顺序分为 3 个层次，依次是 Delphi 程序设计基础→Delphi 软件开发技术→软件项目开发综合实训——Delphi 篇，并将每个层次各成一书，本书是其中的第 2 本，即《Delphi 软件开发技术》。

本书共分 10 章，分别介绍面向对象程序设计、Delphi 自定义组件的开发、Delphi 异常处理与调试、动态链接库（DLLs）编程、Windows API 与资源调用、Delphi 图形及多媒体应用、多线程、文件操作、Delphi 网络组件编程及实训。

本书是高职高专 Delphi 课程的教材，也可作为 Delphi 软件开发人员的自学参考书。

高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

Delphi 软件开发技术

◆ 编　　张世明

责任编辑　潘春燕

执行编辑　韩学义

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

河北人民邮电出版社印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：21.25

字数：507 千字　　2005 年 7 月第 1 版

印数：1－3 000 册　　2005 年 7 月河北第 1 次印刷

ISBN 7-115-13316-6 / TP・4614

定价：28.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

丛书出版前言

目前，人才问题是制约我国软件产业发展的关键。为加大软件人才培养力度和提高软件人才培养质量，教育部继在2003年确定北京信息职业技术学院等35所高职院校试办示范性软件职业技术学院后，又同时根据《教育部等六部门关于实施职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》（教职成［2003］5号）的要求，组织制定了《两年制高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。示范性软件职业技术学院与计算机应用与软件技术专业领域技能型紧缺人才培养工作，均要求在较短的时间内培养出符合企业需要、具有核心技能的软件技术人才，因此，对目前高等职业教育的办学模式和人才培养方案等做较大的改进和全新的探索已经成为学校的当务之急。

据此，我们认为做一套符合上述一系列要求的切合学校实际的教学方案尤为重要。遵照教育部提出的以就业为导向，高等职业教育从专业本位向职业岗位和就业为本转变的指导思想，根据目前高等职业教育院校日益重视学生将来的就业岗位，注重培养毕业生的职业能力的现状，我们联合北京信息职业技术学院等几十所高职院校和普拉内特计算机技术（北京）有限公司、福建星网锐捷网络有限公司、北京索浪计算机有限公司等软件企业共同组建了计算机应用与软件技术专业领域技能型紧缺人才培养教学方案研究小组（以下简称研究小组）。研究小组对承担计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的79所院校的专业设置情况做了细致的调研，并调查了几十所高职院校计算机相关专业的学生就业情况以及目前软件企业的人才市场需求状况，确定首批开发目前在高职院校开设比较普遍的计算机软件技术、计算机网络技术、计算机多媒体技术和计算机应用技术等4个专业方向的教学方案。

同时，为贯彻教育部提出的要与软件企业合作开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的精神，使高等职业教育培养出的软件技术人才符合企业的需求，研究小组与许多软件企业的专家们进行了反复研讨，了解到目前高职院校的毕业生的实际动手能力和综合应用知识方面较弱，他们和企业需求的软件人才有着较大的差距，到企业后不能很快独当一面，企业需要投入一定的成本和时间进行项目培训。针对这种情况，研究小组在教学方案中增加了“综合项目实训”模块，以求强化学生的实际动手能力和综合应用前期所学知识的能力，探索将企业的岗前培训内容前移到学校的教学中的实验之路，以此增强毕业生的就业竞争力。

在上述工作的基础上，研究小组于2004年多次组织召开了包括企业专家、教育专家、学校任课教师在内的各种研讨会和方案论证会，对各个专业按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”一步步进行了认真的分析和研讨：

- 列出各专业的岗位群及核心技能。针对教育部提出的以就业为导向，根据目前高职高专院校日益关心学生将来的就业岗位的现状，在前期大量调研的基础上，首先提炼各个专业的岗位群。如对某专业的岗位群进行研究时，首先罗列此专业的各个岗位，以便能正确了解

每个岗位的职业能力，再根据职业能力进行有意义的合并，形成各个专业的岗位群，再对每个岗位群总结和归纳出其核心技能。

- 根据岗位群及核心技能做出教学方案。在岗位群及核心技能明确的前提下，列出此岗位应该掌握的知识点，再依据这些知识点推出应该学习的课程、学时数、课程之间的联系、开课顺序并进行必要的整合，最终形成一套科学完整的教学方案。

为配合学校对技能型紧缺人才的培养工作，在研究小组开发上述 4 个专业的教学方案的基础上，我们组织编写了这套包含计算机软件技术、计算机网络技术、计算机多媒体技术及计算机应用技术等 4 个专业的教材。本套教材具有以下特点：

- 注重专业整体策划的内涵。对各专业系列教材按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”的思路组织开发教材。
- 按照“理论够用为度”的原则，对各个专业的基础课进行了按需重新整合。
- 各专业教材突出了实训的比例，注重案例教学。每本教材都配备了实验、实训的内容，部分专业的教材配备了综合项目实训，使学生通过模拟具体的软件开发项目了解软件企业的运行环境，体验软件的规范化、标准化、专业化和规模化的开发流程。

为了方便教学，我们免费为选用本套教材的老师提供部分专业的整体教学方案及教学相关资料。

- 所有教材的电子教案。
- 部分教材的习题答案。
- 部分教材中实例制作过程中用到的素材。
- 部分教材中实例的制作效果以及一些源程序代码。

本套教材以各个专业的岗位群为出发点，注重专业整体策划，试图通过对系列教材的整体构架，探索一条培养技能型紧缺人才的有效途径。

经过近两年的艰苦探索和工作，本套教材终于正式出版了，我们衷心希望，各位关心高等职业教育的读者能够对本套教材的不当之处给予批评指正，提出修改意见，也热切盼望从事高等职业教育的教师以及软件企业的技术专家和我们联系，共同探讨计算机应用与软件技术专业的教学方案和教材编写等相关问题。来信请发至 panchunyan@ptpress.com.cn。

目　录

第 1 章

面向对象程序设计

面向对象的程序设计方法是一种以模拟真实世界的概念来组织程序的全新方法，它与传统的结构化程序设计方法有着明显的不同。能够支持面向对象程序设计的语言称为面向对象程序设计语言，面向对象的 Pascal(Object Pascal)语言就是一种面向对象程序设计语言。Object Pascal 扩展了 Pascal 的功能，引入了面向对象的封装、继承、多态等概念。与 C++这和面向对象程序设计语言相比，Object Pascal 更容易学习和使用。

本章详细介绍面向对象程序设计知识，具体包括：

- 面向对象程序设计概念
- 类
- 方法
- 继承和多态
- 对象
- 属性
- 消息

1.1 面向对象程序设计概念

我们眼中的世界是一个充满了个体的世界，个体有自己的特性，个体之间又存在着共性，还存在着千丝万缕的联系，个体又是发展的，个体的发展影响着个性，通过外界的力量，也可以直接改变个体的个性，这就是离散对象组成集合的世界。这种对个体世界的理解启发我们设计程序时，也可以使用抽象和具体相结合的分析方法，在程序里建立类似于现实世界的个体，让这些个体有个性和行为，与其他个体有合理的联系，这就是面向对象的程序设计方法。面向对象的程序设计方法中，把个体命名为对象，把个性命名为数据或者属性，把行为称为方法。

1.1.1 结构化程序设计方法

传统 Pascal 语言支持结构化的程序设计方法，这种程序设计的方法着重于分析应用程序所要实现的功能，根据要解决的问题和程序的功能设计数据结构。分析实际问题时，根据数据流的变化和输入输出划分模块，根据程序功能编写过程。这种编程方法在 Pascal 的语法中得以加强，程序根据模块划分单元，单元内实现数据结构、变量和子程序。主程序内按照实际问题的预定步骤调用各功能模块的子程序，各个程序之间使用协定好的数据结构，共同维护变量的一致性。

用传统的设计方法编写程序，感觉像是在解决一个数学运算的问题。的确我们刚刚学习编写程序的时候也就是编写数学运算。但是很多问题并非像数学运算那样有精确的步骤和一致的数据类型。

1.1.2 面向对象的程序设计方法

尽管面向对象的程序设计方法并不能使代码容易编写，但它使代码容易维护。通过把数据和代码封装在一起，大大简化了定位和修复错误的工作，并最大限度地减少了对其他对象的影响。

面向对象的程序设计方法强调将行为个体对象化，将对象的数据和操作封装在一起，对于一个对象而言，内部数据的更改完全由自己调用操作完成，数据变为对象之后，它就有活动能力，可以主动地表现自己，修改自己，与其他对象联系。对象之间还存在着继承关系，继承其他对象得到的对象具有父对象的全部或者部分数据和操作，表现出来可以和父对象完全相同，也可以部分相同，这就是面向对象的三大特点：封装性、继承性、多态性。

1. 封装性

封装是指通过定义类并且给类的属性和方法加上访问控制来抽象事物的本质特性。类把数据和方法封装在一起，同时隐藏了实现的细节，这样有利于程序的模块化，减少被其他代码干扰的机会。

2. 继承性

继承是指一个对象能够从它的祖先那里获取已有的成员和行为。这样，要创建多层次对象，可以先创建一个通用的对象，然后派生出新的对象。继承的好处是可以共享已有的代码。

被继承的类称为基类，继承下来的类称为派生类，基类的成员自动成为派生类的成员。类的继承具有传递性。例如，假设 T3 继承了 T2，而 T2 又继承了 T1，则可以认为 T3 也继承了 T1。在 Object Pascal 中，TObject 是一个最基本的类，从它派生出了大量形形色色的其他类。

3. 多态性

多态是指程序通过一个一般基类的引用来完成实现在多个派生类中的方法，即当调用一个对象的方法时，实际被调用的代码与对象的实例有关。

与 C++相比，Object Pascal 的面向对象语法适当作了一些简化。例如，Object Pascal 语言中没有运算符重载的概念。Object Pascal 也不支持多继承。所谓多继承，是指一个对象同时继承两个或两个以上的对象，以便同时包含两个祖先对象的代码和数据。

1.1.3 面向对象程序设计的基本概念

类：类描述了具有相似性质的一组对象。这组对象具有相同的数据结构，相同的操作。在 Object Pascal 中类是一种数据结构，定义了该组对象的共同属性和操作。

类要定义公共的属性和方法，根据它创建的对象都可以拥有这些属性，并可使用这些方法。如果有一个类，是其他类的父类，但是它不能创建对象，则称之为抽象类。抽象类就是给相似的类提供一个接口，再定义一些公共的东西。为了实现多态，定义抽象类的方法适用于接口定义。根据继承的关系，类有父类、子类、祖先类、后代类的区分。TObject 就是一个抽象类。

对象：数据和代码封装的统一体。对象与实例的概念等同，实例从类的角度定义对象，表示一个建立在特定类的基础上的对象。对象也可以理解为一个数据和代码封装的变量，而类便定义了这个变量的数据类型。

对象可以被创建，也可以被释放。创建对象时要使用对象所属类的构造方法。对象在创建的时候，要进行对象中数据的初始化工作。如果一个对象所属的类是从其他类继承来的，那么该对象还享有祖先类的属性和方法。

属性：是对对象数据概念的扩展。类的属性，不仅仅是用来赋值和引用的，类可以提供保护方法，使访问对应的对象数据时能够支持自动转换。

属性可暂先理解为数据域。数据域可以设置 3 种保护方法：只允许自己类内访问的私有型，允许所有对象访问的公有型，允许自己的子类访问的保护型。属性还可以有读写设置，可以设置为只读的、只写的、可读可写的。读的时候可以直接读数据，也可以通过子程序转化一下，写也可以有不同方法。

方法：类所提供的操作，可以是过程，也可以是函数。对象可以使用方法以实现对象的操作。

方法就是类的操作。构造方法是类创建对象的方法，析构方法是释放变量的方法，方法可以是函数或过程，也有和数据域相同的 3 种保护方式。

1.2　类

可以认为类就是对象 Pascal 的一个数据类型。类的概念就是抓住对象的相似性，定义它们的共同特征，包括数据和操作。

1.2.1　类的定义

在 Object Pascal 中类被当作一个数据类型来定义，它的语法是：

```
Type 类名=class(父类)
     数据域定义
     过程和函数定义
     属性的定义
End;
```

如果不声明父类，Delphi 自动从 TObject 继承，TObject 是所有类的最终祖先。

在 Delphi 自定义的类中，有一个习惯，即所有定义的类都以 T 开头，建议读者建立类时也使用这个好习惯。

1.2.2　保护方式

Object Pascal 有 4 种类元素的保护方式，类元素包括数据域和方法。这 4 种保护方式为 Published（可视）、Public（公有）、Protected（保护）和 Private（私有）。加上类的保护指令，类定义的语法可改进为：

```
Type 类名=class(父类)
Public
```

```
    类元素定义
Protected
    类元素定义
Private
    类元素定义
Published
    类元素定义
End;
```

1. Private

被声明的类元素仅在该类的方法中被访问，它的子类和实例都无法访问。也就是说，一个私有的属性不可以在所在模块之外的其他模块中读写，一个私有的方法也不可以在所在模块之外的其他模块中被调用。但是，如果在同一个单元文件中定义了两个类（通常把关系非常紧密的两个类定义在同一个单元文件中），那么，在一个类的成员中就可以对另一个类中的私有变量进行访问，或者调用另一个类中的私有方法。

2. Protected

被声明的类元素在该类的方法中可被访问，它的子类和后代类也可访问，除此以外都无法访问。

3. Public

完全可访问，即可以被该类以外的类访问。如果两个类不在同一个单元文件中，则要在 uses 语句中包括被访问的类所在的单元名称。

4. Published

Published 的访问权限基本与 Public 相同，只是在设计期间也可以被访问。通常该类型的成员用在组件类的声明中，这样，就可以在对象编辑器中访问组件的发行类型的成员。

除了在类封装的时候可以限制成员的访问权限外，也可以限制对变量、对象、函数和过程等的访问权限。为了使软件系统具有良好的安全性、健壮性，应该注意这些限制权限的用法。

隐藏程序内部细节、公开接口是类元素保护的意义。如果保护方式不定义，则认为是 Published。

1.2.3 类实例化

定义了类之后，必须对类内的方法进行实现。在单元的实现部分，不必按照定义的顺序，但是必须按照格式“Procedure（或 Function）类名.方法（参数表）”来做。注意在方法内不要引用程序其他部分建立的对象，免得运行过程中对象未建立，访问它时发生保护性的错误。

类的实例化就是利用类的方法创建对象，如果是缺省从 TObject 处继承来的类，它的祖先从未定义过构造方法，那么直接使用如下形式创建。

```
var  对象变量名:类名;
...
对象变量名:=类名.create;
{或者：}  对象变量名:=类名.构造方法名(参数表);...
```