

# Visual Basic程序设计 项目教程

郭晓平 朱鸣华 著



014932017

计算机基础课程系列教材

TP312VB  
32

# Visual Basic程序设计 项目教程

郭晓平 朱鸣华 著



北航 C1720034

TP312VB

32



机械工业出版社  
China Machine Press

710880910

## 图书在版编目 ( CIP ) 数据

Visual Basic 程序设计项目教程 / 郭晓平, 朱鸣华著. —北京: 机械工业出版社, 2014.2  
(计算机基础课程系列教材)

ISBN 978-7-111-45733-6

I. V… II. ①郭… ②朱… III. BASIC 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2014) 第 024352 号

本书是非计算机专业人员学习 Visual Basic 编程的入门教程, 通过完整地介绍一个时钟软件项目, 简明扼要地介绍了 OOP 的程序设计理念, 并应用这一理念对时钟问题进行分析及分类程序设计。书中较为系统地介绍了 Visual Basic 2010, 一方面作为语言工具用以实现 OOP 分类设计, 另一方面则作为用户界面开发平台对分类程序进行组装。在此过程中, 较为详细地讲解了相关的语句结构、程序技巧及如何使用 VB 提供的多媒体功能。希望通过本书的学习, 读者可以了解利用 VB2010 开发 Windows 软件的基本方法, 并能有效地利用 VB 自身的帮助功能完成简单软件的独立制作。另外, 考虑到目前国家计算机等级考试二级 VB 的需要, 本书还介绍了 VB6.0 与 VB2010 的主要差别, 并提供了配套的实例项目与练习教程, 便于读者加深理解及增强学习效果。

本书可作为高等院校非计算机专业或培训机构计算机程序设计基础课程的教材, 也可以作为广大计算机爱好者的自学教程或参加计算机等级考试的参考用书。



出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 刘立卿

印刷: 北京市荣盛彩色印刷有限公司

版次: 2014 年 3 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 14.25

书号: ISBN 978-7-111-45733-6

定价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

# 前 言

Visual Studio 2010 是由微软公司开发的一套功能强大的程序编辑平台，Visual Basic 2010（简称 VB2010）是其重要的组成部分。VB2010 建立在新的 OOP 理念之上，但同时又继承了传统 Visual Basic（简称 VB）易学易用的优点，使程序设计与其他 OOP 语言例如 C++、Java 等一致，但又较其他程序语言易于掌握，便于非计算机专业人员在较短的时间内编写实用程序。

本书是学习 VB 编程的入门教程，读者对象为非计算机专业本科学生或从业人员。VB 语法知识是本书的重要但非基本内容。全书以分析、模拟到最终实现一个电子时钟系统为主线，通过实例向读者解释如何利用计算机程序语言（例如 VB）解决生活和工作中遇到的实际问题。书中的章节按照提出问题、分析问题、解决问题的思路顺序展开。通过阅读本书，希望读者不仅可以学会 VB2010 的基本知识，而且更重要的是能够掌握利用计算机程序解决实际问题的基本思想方法。

本书共分 12 章。第 1 章“项目需求与分析”是程序开发的基础，目的是使读者在学习 VB 前知道自己将要干什么以及怎么干的基本框架；第 2 章、第 3 章和第 10 章介绍 VB 界面设计技术；第 4 ~ 8 章介绍 VB 语言基本语法；第 9 章介绍多媒体编程；第 11 章介绍文件；第 12 章介绍 VB6.0 与 VB2010 的差别。

尽管 VB6.0 已经在世界范围内退出了软件开发舞台，但考虑到目前国家计算机等级考试二级 VB 仍然使用 VB6.0 环境，因此本书对 VB6.0 与 VB2010 做了比较性的说明。作为开发平台，Visual Studio 2010 更为成熟，自动化、智能化程度较 Visual Studio 6.0 更高，因而对初学者更为方便。学了 VB2010，VB6.0 则是非常易于理解和掌握的。

本书图文并茂，所有过程都依实际屏幕显示逐步讲述，便于读者跟进上机操作。为方便读者学习和上机，书中所用图像和每个项目设计程序，读者可以从机械工业出版社华章公司网站 [www.hzbook.com](http://www.hzbook.com) 下载。

为解释 VB 语法或强调某些程序使用技巧，本书在电子时钟系统项目以外还加入了一些独立的小段程序。

本书的配套用书《Visual Basic 程序设计项目教程实例与练习》提供了很多练习，对书中的基本概念做了进一步的解释，并在每章增添了实例项目，配套使用能帮助读者加深对相关知识理解，使学习效果更佳。

本书可作为高等院校非计算机专业的计算机语言课程教材，也可作为初学编程人员的参考书，同时也可作为广大计算机爱好者的自学教程。

感谢张启晖对本书的指导性建议。由于作者水平有限，书中难免有错误之处，敬请读者批评指正。

作者

2013 年 12 月于大连理工大学



# 教学建议

章 节	教学要求	课 时
第 1 章 项目需求与分析	了解 OOP 特点 了解 VB 语言功能特点	2
第 2 章 Visual Basic 集成开发环境	掌握 VB 集成开发环境的组成 熟练使用集成开发环境中的常用操作 掌握项目资源管理器中的主要文件类型	1
第 3 章 窗体和基本控件	掌握窗体和基本控件的创建, 以及它们的常用属性、方法和事件 掌握标准事件的组成	3
第 4 章 Visual Basic 语言基础	掌握 VB 基本结构和书写格式 掌握数据类型的概念和常用数据类型 了解数据类型间的转换 掌握 VB 运算符和表达式 掌握常用内部函数的使用方法 掌握 VB 语法检查功能的使用	4
第 5 章 控制结构	了解用控制台应用程序编写 VB 程序 掌握用 If-Then 结构和 Select 结构控制程序运行流程 掌握用 For-Next 结构、While 结构和 Do-Loop 结构控制循环	3
第 6 章 数组、枚举与集合	掌握数组的概念和基本算法 了解枚举的概念和用法 掌握集合的概念与用法	3
第 7 章 过程	掌握过程的作用和构造原则 掌握值传递与地址传递的概念 掌握函数过程与子过程的构造与调用 掌握作用域的概念和使用 掌握静态变量的概念和使用	3
第 8 章 自定义类	掌握构造类的语法及类成员的访问方式 掌握如何设计类的继承 掌握如何实现类的多态性	4
第 9 章 多媒体编程	掌握如何绘制直线, 了解如何在窗体中写字 了解如何播放多媒体	2
第 10 章 常用控件	掌握常用控件的用法	3
第 11 章 文件	掌握数据文件的分类 掌握顺序文件的访问方法 了解随机文件的访问方法	2
第 12 章 Visual Basic 6.0 与 Visual Basic 2010 的差别 (选讲、自学)	了解 VB2010 与 VB6.0 在控件名与事件参数上的差别	2
总课时		32

说明: 1. 建议采用课堂讲授与课后作业、上机操作相结合的教学方法, 调动学生的学习兴趣。

2. 建议利用本教程的配套习题和上机指导教材来巩固各章节的重点知识内容。

3. 对于本书的学时数, 可根据具体情况调整。

# 目 录

前言	
教学建议	
项目设计清单	
第1章 项目需求与分析	1
1.1 提出问题	1
1.1.1 问题起源	1
1.1.2 项目需求	1
1.2 分析问题	3
1.2.1 OOP 概述	4
1.2.2 为时钟系统建模	6
1.3 解决问题	12
1.4 Visual Basic 简介	14
1.5 程序设计流程	15
小结	15
习题	15
第2章 Visual Basic 集成开发环境	16
2.1 Visual Studio 2010 的启动与保存	16
2.2 Visual Basic 集成开发环境简介	18
2.2.1 主窗口	18
2.2.2 设计窗口	20
2.2.3 工具箱窗口	22
2.2.4 属性窗口	23
2.2.5 解决方案资源管理器	23
2.2.6 代码窗口	25
2.3 Visual Basic 程序的运行与退出	25
小结	26
习题	26
第3章 窗体和基本控件	27
3.1 创建控件	27
3.2 控件的属性、方法和事件	28
3.2.1 控件的属性	28
3.2.2 控件的方法	29
3.2.3 控件的事件	29
3.3 窗体	31
3.3.1 设置启动窗体	32
3.3.2 窗体的常用成员	32
3.3.3 设计窗体界面的技巧	36
3.4 Label 控件	37
3.5 Button 控件	38
3.6 PictureBox 控件	41
3.7 ToolTip 控件	42
3.8 容器	43
3.9 TextBox 控件	46
3.10 Timer 控件	50
小结	52
习题	52
第4章 Visual Basic 语言基础	53
4.1 编码规则	53
4.2 语句	54
4.3 基本数据类型	55
4.3.1 常用标准数据类型	55
4.3.2 数据类型的转换	56
4.4 常量与变量	58
4.4.1 常量	58
4.4.2 变量	60
4.5 运算符	61
4.5.1 算术运算符	61
4.5.2 关系运算符	62
4.5.3 逻辑运算符	62
4.5.4 赋值运算符	63
4.5.5 字符串运算符	64

4.6 表达式	64	7.2 Sub 子过程	105
4.7 常用内部函数	65	7.3 过程的参数传递	106
4.7.1 数学函数	65	7.4 过程的嵌套与递归调用	108
4.7.2 字符串函数	66	7.5 模块	109
4.7.3 日期时间函数(属性)	67	7.6 作用域	111
4.7.4 随机函数	68	7.7 静态变量	116
4.7.5 转换函数	69	小结	116
4.7.6 格式化输出函数	69	习题	117
小结	71	第8章 自定义类	118
习题	71	8.1 创建类和类的变量	118
第5章 控制结构	72	8.1.1 创建类	118
5.1 顺序结构	72	8.1.2 命名空间	118
5.1.1 赋值语句	72	8.1.3 设置属性和字段	120
5.1.2 控制台输入/输出	74	8.1.4 建立成员函数	121
5.2 选择结构	76	8.1.5 创建类的变量	122
5.2.1 If 块结构	76	8.1.6 构造函数与析构函数	123
5.2.2 Select Case 语句	80	8.1.7 重载	127
5.2.3 其他选择结构	82	8.1.8 共享成员	132
5.3 循环结构	83	8.1.9 类变量的生命周期	132
5.3.1 For-Next 循环	83	8.2 继承、多态性	133
5.3.2 While 循环	85	8.2.1 继承	133
5.3.3 Do-Loop 循环	85	8.2.2 多态性	136
5.4 其他辅助控制语句	88	小结	143
5.5 应用举例	89	习题	143
小结	90	第9章 多媒体编程	144
习题	90	9.1 绘图	144
第6章 数组、枚举与集合	91	9.1.1 绘图基础	144
6.1 数组	91	9.1.2 绘制形状	147
6.1.1 数组的维数	91	9.1.3 绘制文本	154
6.1.2 数组的声明	92	9.1.4 设置填充属性	154
6.1.3 Array 类	93	9.1.5 清除绘图表面	160
6.1.4 创建控件数组	93	9.2 多媒体播放	160
6.1.5 数组的基本操作	94	小结	161
6.2 枚举	97	习题	161
6.3 集合	98	第10章 常用控件	162
6.4 数组、枚举与集合的比较	101	10.1 菜单控件	162
小结	102	10.1.1 MenuStrip 控件	162
习题	102	10.1.2 ContextMenuStrip 控件	162
第7章 过程	103	10.2 单选按钮和复选框	164
7.1 Function 函数过程	103	10.3 列表框工具	167

10.3.1 列表框	167	12.2 窗体与基本控件	199
10.3.2 复选列表框	169	12.2.1 Visual Basic 6.0 的常用属性 和事件	199
10.3.3 组合框	170	12.2.2 窗体的属性和事件	199
10.4 与用户交互	172	12.2.3 基本控件的属性和事件	200
10.4.1 InputBox 函数	172	12.3 Visual Basic 语言基础	201
10.4.2 MessageBox 消息框	173	12.3.1 基本数据类型	201
10.5 Shell 函数	177	12.3.2 常用内部函数	202
小结	178	12.4 数组	203
习题	178	12.4.1 数组的声明	203
第 11 章 文件	179	12.4.2 控件数组	204
11.1 文件分类	179	12.5 常用控件	205
11.2 访问文件	179	12.5.1 菜单	205
11.2.1 访问顺序文件	179	12.5.2 单选按钮、复选框和框架	206
11.2.2 访问随机文件	182	12.5.3 列表框和组合框	206
11.3 OpenFileDialog 控件和 SaveFileDialog 控件	182	12.5.4 滚动条	207
11.3.1 OpenFileDialog 控件	182	12.6 文件	209
11.3.2 SaveFileDialog 控件	184	12.6.1 文件系统控件	209
11.4 用 Stream 读写文本文件	185	12.6.2 数据文件	210
小结	195	小结	213
习题	195	习题	213
第 12 章 Visual Basic 6.0 与 Visual Basic 2010 的差别	196	习题参考答案	214
12.1 Visual Basic 6.0 的启动与保存	196		



# 第 1 章 项目需求与分析

编写程序是为了解决现实世界中的问题。因此，在编写程序之前，必须搞清楚要干什么，需要达到什么目的，以怎样的技术路线来达到所设计的目的。

人类在不断地提出问题、分析问题和解决问题中进步。无论是科学家、工程技术人员，还是高层管理人员，都遵循这个思维模式。软件开发也是如此。本章以人们熟悉的时钟为背景，介绍“时钟系统”项目开发中如何提出问题、分析问题，以及基于 OOP 的常用分析方法。在软件项目开发中，Visual Basic 是用于解决问题的一种工具。本章最后对 Visual Basic 的发展史和特点做了简要介绍。

## 1.1 提出问题

提出问题是软件项目开发的基础。想干什么，要达到什么目的，将问题尽可能描述完整，这样才能减少软件项目的维护工作量。

### 1.1.1 问题起源

时间是事件发生到结束的时刻间隔。时刻是对时间的一种描述。不同的时刻显示出时间的流动性。因此，相同的时间可以对应两个不同的时刻。人们除了需要知道时间，更需要掌握时刻。

人们为了方便相互间的交流和活动需要“对时”，即对各个事务的先后次序进行对比；还需要“计时”，即比较各个事物存在过程的长短。计时器是通过某个变化事物的存在过程（尤其是周期性的）来衡量其他事物存在过程长短的装置。我们比较熟悉的计时器有机械钟表、石英钟等。

人们把一个昼夜轮回的时间定义为一天，并把一天划分为 24 小时。由于计量时间的起算点是天体过子午圈的时刻，而对地面上不同地理经圈的两地来说，它们的子午圈是不同的，使得各地时刻的起算点各不相同，这就形成了各自的时间系统——地方时。

时间是每个人的最大资本和财富。时间是有限而宝贵的。为了方便人们掌握时间，要求制作一个多功能计时器——时钟（Clock）系统。

### 1.1.2 项目需求

制作一个“Clock”系统，系统外观如图 1-1 所示。

将图 1-1 称为“Clock”界面。之所以用英文“Clock”而不是中文“时钟”，是为了提高项目软件系统的可移植性。

要求 1：在“Clock”界面中，中间的大钟称为“主钟”，用于显示指定城市的当地时间。在默认情况下显示北京时间。位于“主钟”左边的文字分别用中文和英文说明城市名称。

要求 2：位于“Clock”界面下半部的 11 个指针式时钟分别显示世界主要城市的当地时间，并在每个时钟的正下方显示该城市所在国家的国旗。当鼠标指向某个时钟画面时，用中文显示所指时钟的城市名和用括号括起来的国家名称，如图 1-2 所示。

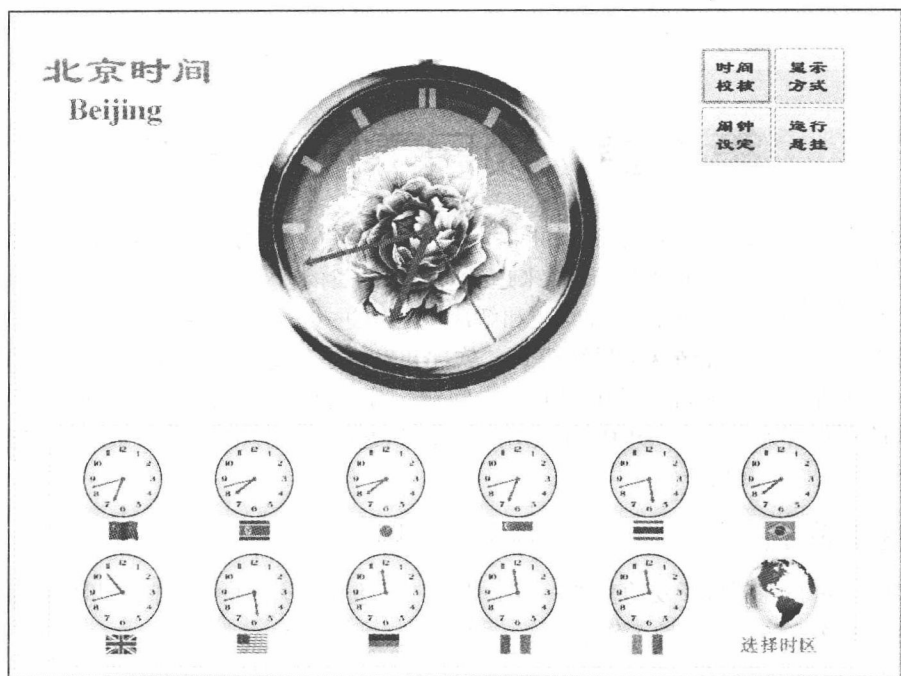


图 1-1 Clock 界面

要求 3：位于“Clock”界面右下角的“选择时区”，用于查看世界各国的当地时间。当单击此图像后，进入“各地时间”界面，如图 1-3 所示。用户可以输入城市名称来查询当地时间，也可以利用菜单选择所关心时区的时间。

说明：1884 年 10 月 13 日，在华盛顿召开的国际天文学家代表会议决定，以经过英国伦敦东南格林尼治的经线为本初子午线，作为计算地理的起点和世界标准“时区”的起点。每隔经度  $15^\circ$  划分一个时区，全球共 24 个理论时区。

根据东早西晚的规律，已知一个时区的区时，求另一个时区的区时的方法是：东加西减。

为了节省能源，世界上有些国家实行夏时制。本系统暂时不考虑夏时制。

要求 4：在“Clock”界面的右上方设置 4 个功能按钮，分别为：

①时间校核：“Clock”界面所显示的时间是计算机系统时间。当计算机的系统时间和实际时间出现偏差时，需要进行调整。“时间校核”按钮就是用来调整系统以

显示正确的时间的（但不修改计算机的系统时间）。这项操作至关重要，因此要设权限。单击“时间校核”按钮，弹出口令窗口（窗口也称窗体，这两个术语经常互换使用。更准确地说，



图 1-2 显示城市名和国家名

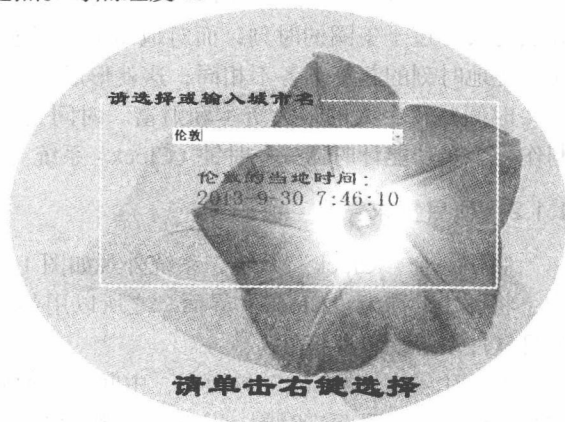


图 1-3 查询城市的当地时间

窗口指程序运行时用户可见并可与之交互的界面；窗体是开发设计时看到的窗口)：

- 在 1 分钟内完成口令检查，即开启口令窗口时间不得超过 1 分钟，过时自动关闭口令窗口，“时间校核”按钮失效。
- 连续重复试探口令最多 3 次。
- 除非退出系统后再重新启动系统，累计试探口令不得超过 10 次。超过 10 次后，“时间校核”按钮失效。
- 当用户忘记口令时，系统提供“提示口令”。只要经过简单的计算（向用户提供计算器）就能得到口令。

②显示方式：主钟的显示方式有两种，即指针式和数字式。图 1-1 是指针式，单击“显示方式”按钮改成数字式，如图 1-4 所示。反复单击该按钮，在“指针式”和“数字式”两种显示方式之间切换。

③闹钟设定：单击该按钮，弹出闹钟设定界面，如图 1-5 所示。当主钟显示的时间与“闹钟设定”的时间相等时，系统发出由用户选择的音乐。

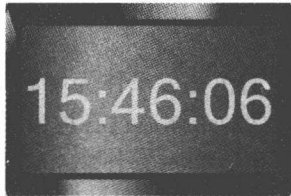


图 1-4 数字式显示时间

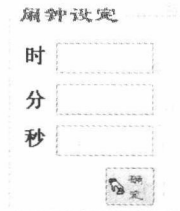


图 1-5 闹钟设定界面

④运行悬挂：进入程序运行时，时钟运行，“Clock”界面上的全部时钟显示各自的当地时间。单击“运行悬挂”按钮，系统处于悬挂状态，全部时钟停顿在单击该按钮的时刻。反复单击该按钮，系统在“运行”和“悬挂”之间切换。

要求 5：在“Clock”界面上不设退出系统的明显标志。若单击“Clock”界面的空白处会弹出一个对话框，询问是否要终止程序运行。

## 1.2 分析问题

分析问题是软件项目开发的第二步，也是重要的一步。它可以为确定软件的解决方案提供依据。

项目需求已提出，希望用计算机来实现。但用 1.1.2 节的方式阐述问题，计算机无法理解。所以必须对问题进行提炼、抽象，最终用计算机能理解并处理的语义来表达。这种无歧义的书面描述称为模型，从现实问题到模型的转化过程称为创建模型，简称建模。

因此，软件项目开发的第二步也可以说是建模。

建模过程会因分析问题的视角不同而不同。目前的软件方法有两种：面向过程和面向对象。

- 面向过程：以事件发生的先后顺序为导向，用推理机制解决问题。
- 面向对象：以个体为中心（导向），不关心事件发生的先后顺序，不存在逻辑推理。

下面以学生交作业为例说明面向过程与面向对象的分析方法。

面向过程分析法：

①发生什么事：学生交作业

②事情发生过程：先有交作业指令，然后交作业

③谁发指令：李老师

谁交作业：1 班学生

④整个过程：李老师向 1 班学生发出交作业指令→1 班学生收到指令后交作业→李老师收作业

面向对象分析法：

①事件中的人物：李老师，1 班学生

②李老师做什么：发交作业指令，收作业

1 班学生做什么：交作业

③李老师与 1 班学生有什么关系：李老师向 1 班学生发出交作业指令→1 班学生收到指令后交作业→李老师收到作业。

本教程介绍面向对象方法。

## 1.2.1 OOP 概述

### 1. 类与个体

OOP (Object Oriented Programming, 面向对象程序设计) 是以个体为中心 (导向) 的设计方法。从 OO 的角度来说, 大千世界本质上是由各自独立的个体通过各种各样的关系组合而成的, 每个个体都有自己的特征和行为。在本项目中, 主钟是一个个体, 显示主要城市的 11 个时钟就是 11 个个体。时钟的外观样式是时钟的特征之一, 在图 1-1 中, 12 个时钟有 12 种各自的样式。此外, 这 12 个时钟都有一个“时间”特征, 它们彼此之间有的相同有的不同。时间变化是时钟的行为。将这 12 个时钟按一定的规则摆放在一起, 形成了所需的“时间显示”系统。

在 OOP 中, 把个体所具有的特征 (通常称为属性) 用数据来描述; 把个体所具有的行为 (称为方法和事件) 用函数来体现。不同的个体有不完全相同的属性、方法和事件。

在 OOP 中, 把具有相同特征和相同行为的一组相似个体组合在一起, 构成类。类所包含的个体范围可大可小, 完全由程序开发人员自己来决定, 其衡量的标准是所开发的软件的质量, 包括可靠度和稳定性, 是否易于扩充和维护等。例如, 研究“小学一年级学生的上网”问题, 若构造“人类”来研究, 会使类中包含大量与研究课题无关的信息; 若构造“学生类”, 就要比“人类”更易于研究所定问题; 若将“小学生类”从“学生类”中独立出来, 就更切合所要研究的问题了。

类和个体关系密切, 但并不相同。实际上, “类”是对个体的抽象和描述, “个体”是类的实例化。

在一般讨论 OO 的书籍中, 也将类实例化出来的个体叫做该类的 Instance (实例、实体或运行个体)。

从程序设计的观点, 类只是一种抽象的数据类型, 而类的实例化就是该类的一个变量。例如在 Visual Basic 中, Integer 可以看做是整数类, 不能直接用 Integer 类来做加减运算。举例:

```
Dim A As Integer      ' 声明变量 A 是一个整数
Dim B As Integer      ' 声明变量 B 是一个整数
```

```

A=10           ' 正确
B=A+8         ' 正确
Integer=10    ' 错误!

```

从以上的程序片段看出：类（Integer）可以通过声明来实例化个体（A 或 B），但是类本身不可用来运行。

### 2. 类的成员

通常，类中的数据称为类的数据成员或属性，函数称为类的成员函数或方法。所以，类包含以下要素：

- ①属性：（可选）数据。
- ②方法：（可选）函数或过程。
- ③事件：（可选）类的实例化可以识别的操作（例如单击鼠标），也可以为其编写代码进行响应。在 OOP 理论中并没有事件的概念，事件是可视化编程语言特有的方法。

关于事件，本书只要求读者掌握 Visual Basic 系统中控件的事件的使用方法，并不关心如何在自定义类中创建和使用自定义事件。

### 3. OOP 的三个基本特征

#### (1) 封装性

封装是 OO 方法的一个重要原则。封装性有两个含义：第一，把类的全部属性和全部方法结合在一起，形成一个不可分割的独立单位（例如图 1-6）；第二，尽可能隐藏类的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系，该含义也称做“信息隐蔽”。

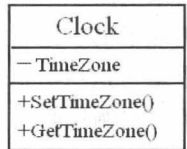


图 1-6 封装性

在 OOP 中，类的封装性把类的属性声明为私有的，即只能被类内的成员函数访问。类外要访问类的数据，必须经过类所定义的外部接口成员函数来访问，从而实现了数据存储和数据操作的分离。类内部数据存储方式的改变所产生的影响被控制在类内，对程序的其他部分没有影响。以后，即使数据存储方式发生改变，也不必为此重写类外使用这些数据的函数，从而使得程序的维护更容易。

理解封装的另一种方式是想象个体是“谦恭的”。如果想从同事那里借钱（在超市购物忘带钱包），不能抢夺同事的钱包，大翻一通，看看里面是否有足够的钱，而应询问他是否可以借你一些钱，让同事自己翻钱包。

在本项目中，时钟 Clock 有时区“TimeZone”特征。把“TimeZone”定义成私有变量（图 1-6 中，“TimeZone”前面有一个“-”符号，表示是私有的），类外不可直接访问。同时定义一个公有“SetTimeZone”函数（在图 1-6 中，“SetTimeZone”前面有一个“+”符号，表示是公有的）和一个公有“GetTimeZone”函数。类外通过“SetTimeZone”函数对 Clock 类的个体的“TimeZone”赋值，或者通过“GetTimeZone”函数读取“TimeZone”的值。

#### (2) 继承性

在原有类的基础上派生出新类，新类将继承原有类的公有属性和方法，并且还可以添加新的特性。派生出来的新类称为派生类或子类；而被派生的类称为基类或父类。本书使用术语“子类”和“父类”。派生出来的子类也可以用作另外一个子类的父类。

一个父类可以派生出多个子类，一个子类也可以从多个父类派生而来。通常把从一个父类派生出一个子类的过程称为单继承；把从多个父类派生出一个子类的过程称为多继承。本书只介绍单继承。

在本项目中，时钟有两种形态：数字式与指针式。因此，可以定义“数字式时钟”类和“指针式时钟”类，而将数字式和指针式共同的成员提取出来组建“时钟”类。时钟类（Clock）是父类，而数字式时钟（DigitalClock）和指针式时钟（AnalogClock）是子类，如图 1-7 所示。

从图 1-7 可见，数字式时钟的属性有时间、时区、时钟中心坐标和放大系数，这些属性都是从它的父类“时钟”类继承来的。

而指针式时钟的属性，除了从父类继承而来的时间、时区、时钟中心坐标和放大系数以外，还增加了时针、分针和秒针三个属性，共有 7 个属性。

采用继承，可以使得代码复用，从而减轻程序的开发和维护成本。

### （3）多态性

多态性是指定义名称相同但具有不同功能的方法或属性的多个类。通常有两种不同的方法来实现多态性：继承和接口。本书只介绍基于继承的多态性：在父类中定义方法并在子类中重写这些方法。这样，当通过父类实例化或子类实例化调用方法时，将会产生不同的行为。

在图 1-7 中，父类时钟“Clock”定义了一个方法“画图”。这是个虚函数，其中并没有指定应该如何画图。在“数字式时钟”类中，将其父类的“画图”方法重新定义成“写时间”；而在“指针式时钟”类中，重新定义成“画指针”。因此，同一个“画图”方法，在不同的类中有不同的行为，这就是 OOP 中的“多态性”特征。

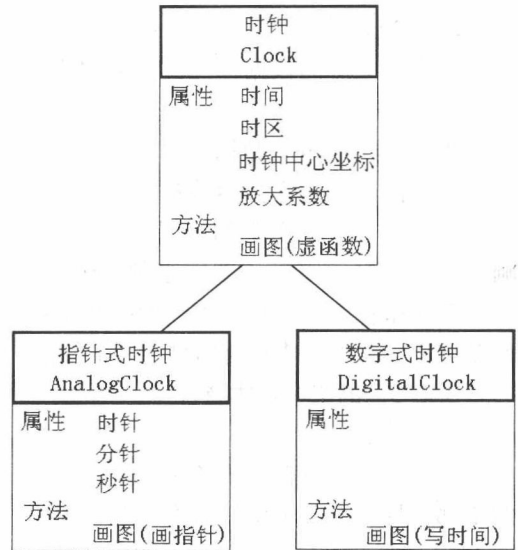


图 1-7 继承关系

## 1.2.2 为时钟系统建模

概括起来，一个项目的模型是一些有意义的静态模型和动态模型的组合。

- 静态模型描述系统的逻辑或物理部分，以及它们如何连接在一起。或者说，静态模型描述如何构建和初始化系统；
- 动态模型描述静态部分如何相互协作，系统在运行时如何动作。

静态模型可以使用类图来描述。类图显示了系统要处理的个体和这些个体之间的相互关系。动态模型可以使用通信图来表达。

统一建模语言（Unified Modeling Language, UML）是软件业的标准语言。这里不涉及 UML 表示法的所有复杂性，只介绍最基本的部分，满足研究本项目所需即可。

### 1. 静态分析

#### （1）确定类

对系统功能进行描述时，其中的名词可以作为候选类。在候选类中删除诸如“系统”、“参与者”、“边界”之类的名词，然后对剩余的候选类进行简短描述。如果不能给类写出简短描述，就可能希望它表示更多的内容，此时应把它分解为多个类。



例如，“时钟”自然是备选类，但时钟中有“数字式时钟”和“指针式时钟”，在“时钟”类中很难简短地描述“数字式时钟”和“指针式时钟”，因此应该分别建立“数字式时钟”类和“指针式时钟”类。

## (2) 标识类的关系

类之间的常用关系包括：

- 继承：子类继承父类的属性和行为。例如，“数字式时钟”类和“指针式时钟”类分别继承了“时钟”类的所有属性和行为。
- 关联：一种类型的个体与另一种类型的个体之间的结构关系。比如，公民与身份证之间的一对一关系；公司与员工之间的一对多关系。指针式时钟与指针之间的一对三关系。
- 组合：类的数据成员是其他类的类型。例如，“指针式时钟”的“指针”属性是“指针”类的变量。

## (3) 绘制类图和关系

类图显示了系统中存在哪些类，以及这些类之间的关系。

图 1-8 显示了如何在类图中描述继承：实线上的空心箭头从子类指向父类。为了强调子类的层次结构，箭头可以以图 1-8b 的样式合并。因此，“指针式时钟”和“数字式时钟”都是“时钟”的子类。

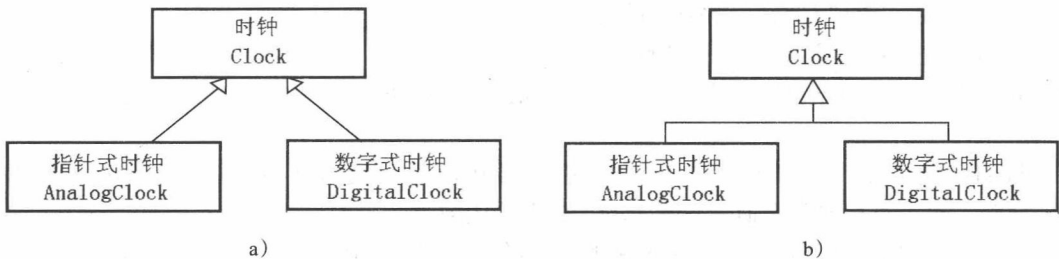


图 1-8 用 UML 描述继承

两个类之间的组合关系表示为组合端带有一实心菱形的线条。所以图 1-9 显示“点”类是“时钟”类的一部分。

根据 UML 规则，可以在关系线端表示运行时变量的数量（关系中的多重性）。在图 1-9 中，数字 1 表示时钟中有一个点类型变量（用来表示时钟的中心坐标点）。

## (4) 属性

属性是类的一个特性。在分析阶段应该为属性提供简短的描述。如果不能提供简短的描述，该属性就应分拆为几个属性，或者自成一类。

例如，“指针式时钟”中有“指针”属性，但“指针”属性本身包含粗细、长短、颜色等特征，为“指针”属性提供简短描述有困难，因此应该创建“指针”类，而“指针式时钟”中的“指针”属性是“指针”类的变量。

在类名的下面添加一个分隔线，就可以在类图中显示属性，且在属性名称右面的冒号之后指定属性的类型，如图 1-10 所示。在图 1-10 中，clockArm 是“指针”类的名称。

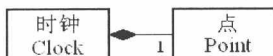


图 1-9 用 UML 描述组合

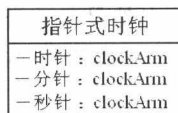


图 1-10 用 UML 描述属性

为简单起见，应避免包含派生属性。例如，圆的属性包括半径、直径、周长和面积。但是，只要存储其中的一个属性，就可以在运行期间计算出其余属性。所以只需在类图中存储四个属性中的一个（通常存储半径）。

### (5) 给类添加操作

由于动态分析中的消息都对应类中的一个操作（方法），所以应在类图中体现出来。在类图上，方法显示在属性下面的单独分隔区中，如图 1-11 所示。

使用 UML 表示法的类图由三个方框组成：类名、属性和方法。方法名称旁边的圆括号表示需要的参数，即使没有参数，也最好包含括号，以使方法名称与属性名称明显地区分开（当省略方框中的一个或多个部分时，给方法加上圆括号将非常重要）。

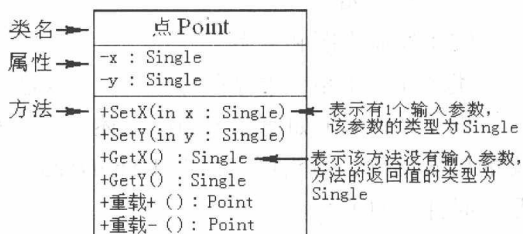


图 1-11 用 UML 描述方法

## 2. Clock 系统静态图

图 1-12 是一个“Clock”系统的静态图。实际上对任何系统来说都没有固定的静态图，程序开发人员可以根据自己的常识、经验、直觉，以不同的方式表达系统，找出成功的实现方式。

说明：创建类时，若两个以上的类中包含相同的成员，通常就把相同部分提出来构建它们的父类，这个过程非常类似于数学上的“提取公因数”。

在图 1-12 中，时针类、分针类和秒针类的成员好像完全相同。但是，注意到这三个类的属性都是静态变量。对于静态属性，类的所有实例化个体的值相同。也就是说，所有时钟的指针属性（长度、宽度、颜色）相同。但是，时针、分针、秒针彼此之间的属性值不相同：通常时针短而宽；秒针细而长，且秒针的颜色有别于时针和分针。最主要的是，在同一时刻，时、分、秒的数值不同，例如时刻 10:32:22，时=10，分=32，秒=22。如果把时针类、分针类和秒针类的属性合并到指针类中，或者无法实现静态参数，或者无法区别时针、分针、秒针之间的差异。而按照图 1-12 所示设计，可以达到预期目标。

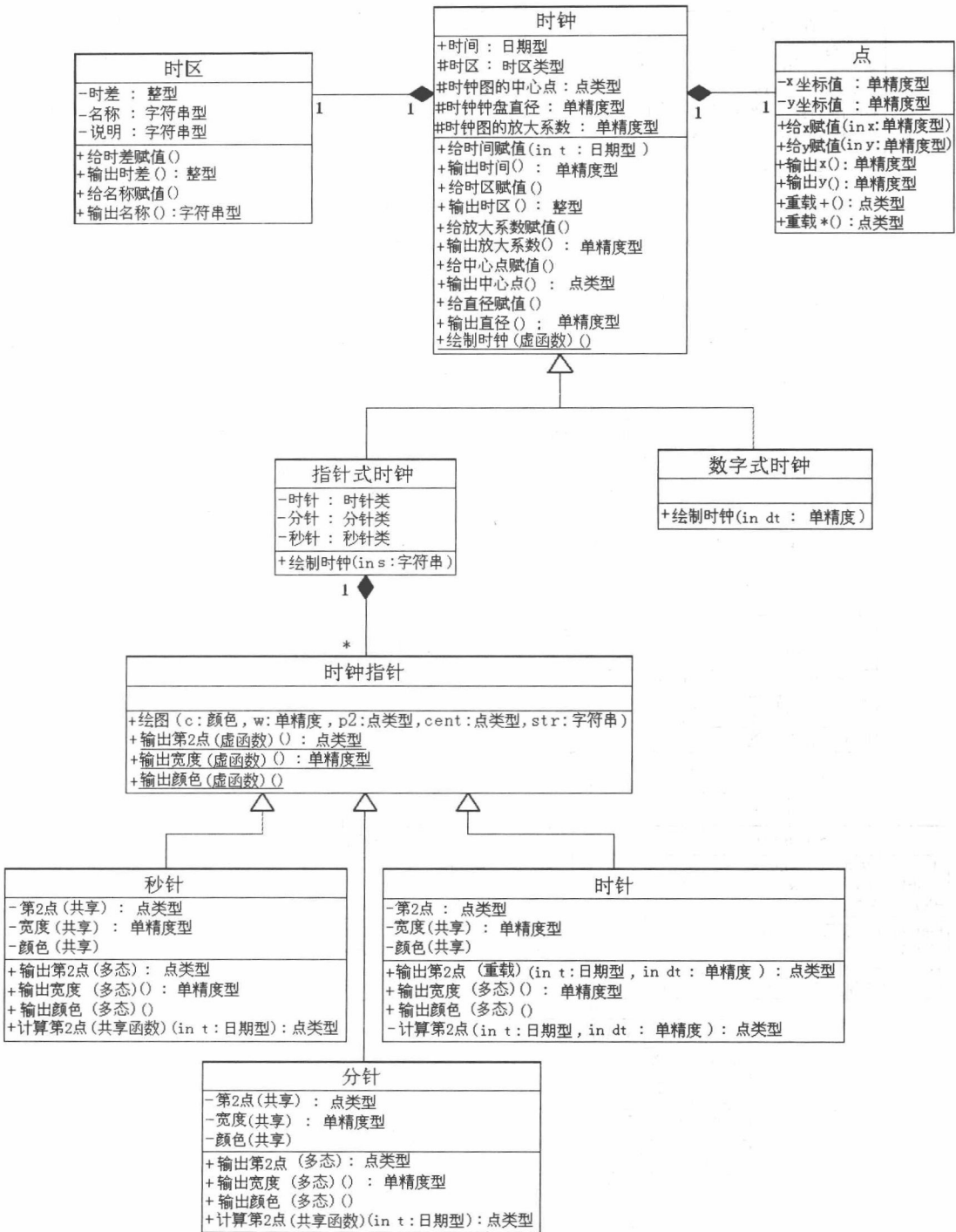
考虑到系统的移植性，开发软件代码尽量采用英文，尽管汉化的版本允许使用汉字。图 1-12b 中的类及其属性和方法都用英文命名，以便和程序代码对应。

## 3. 动态分析

系统的行为可以采用通信图（亦称协作图）来描述。通信图通过个体之间的连接和它们相互发送的消息来显示参与交互的个体。通信图由活动者、个体、链和消息组成。

- 活动者。活动者可以是人、计时器或另一个计算机系统，负责发送初始消息，启动一个操作。
- 个体。负责发送和接收消息。在通信图中，个体用矩形框表示。有三种表示法，如

图 1-13 所示。在图 1-13 中，“张晖”指明个体，“学生”定义“张晖”的类型。无论采用何种表示法，名称下面都需带有下划线。



a) 中文版

图 1-12 系统静态图