

学习中文编程——易语言（一）：编程的遐想

我是个电脑爱好者，能够操作电脑，但不能说精通。

电脑系统由硬件和软件两部分组成，简单的说：硬件是指电脑设备，软件是指电脑程序。能够操作电脑，主要是说能够在电脑上应用一些常用的电脑软件。

电脑软件分为系统软件和应用软件两类。

系统软件又可分为操作系统和支撑软件：操作系统是管理电脑硬件与软件资源的程序，同时也是计算机系统的内核与基石，如微软的 Windows 系统，使用最广泛；还有 Linux 操作系统，源代码开放，多用于大型服务器。支撑软件是支撑各种软件的开发与维护的软件，又称为软件开发环境，编程软件属于支撑软件，如 C++、VB、Java，还有中文编程软件——易语言。

应用软件是为满足不同领域、不同问题的应用需求而提供的软件，它可以拓宽计算机系统的应用领域，发挥硬件的功能，如 Office 办公组件、Photoshop 图像处理软件等。

能使用电脑操作系统和一些常用应用软件，是电脑爱好者的基本要求；能够使用支撑软件中一种编程软件，自己编程，才能算是电脑高手。

我是 1999 年开始接触电脑的，一开始就对电脑产生浓厚的兴趣。当时工作比较忙，经常加班加点，我利用工作之余和在机关值班时间，制作启动软盘，在 DOS 下格式化硬盘、进行硬盘分区，用光盘装 Windows98 操作系统，安装、使用常用的或自己需要的应用软件。

我也曾经尝试着学习编程，从网上下载 D 版 C++、VB，虽说是中文版，但编写代码还是需要用英文，什么参数、变量、常量更是一头雾水，束手无策。

大约在 2006 年，一个偶然的机会，在阅读《电脑爱好者》杂志时，我得知有一种中文编程软件——易语言。于是我与“大连大有吴涛易语言软件开发有限公司”联系，并邮购了学习教材《易语言编程系统》，开始艰难的编程学习。

2012 年，我正式退休，闲来无事，便又想着学习编程。我从淘宝网上购买了最新的易语言学习教材《中文编程——从入门到精通》，学编程，全当消遣。

“老骥伏枥，志在千里；烈士暮年，壮心不已”。通过努力，应该能学会编程，虽然与高手还有差距……



《易语言编程系统》、《中文编程——从入门到精通》

学习中文编程——易语言（二）：初识易语言

易语言诞生于 2000 年 9 月，由大连飞扬工作室吴涛先生编写，至今最高版本是“易语言 5.11”。

易语言分完全版和精简版：完全版包括主程序、易语言知识库、易语言多媒体教程，安装后大约 500 多 M；精简版只包括主程序，和完全版主程序相比，少了一些命令，安装后只有 100 多 M。

易语言有收费和免费两种：收费版本有全部功能，可配合加密狗使用；免费版本对功能有所限制，不可编译输出，也不能使用加密狗。

通过百度可搜到易语言的 D 版，其中“易语言 v5.11 完美破解版”还是比较好用的。可以试用。

对于初学者来说，易语言有如下特征：

1. 全中文支持：易语言用中文编写代码，不是把现有的编程工具简单进行汉化、封装，而是有自己的编译器，中文汉语源代码被直接编译成电脑的 CPU 指令。不需精通英文，只需用汉语和中文思维方式就能写出程序。易语言支持中文输入，并支持南方方言发音，默认中文首拼+全拼，编写代码时，只要正确输入每字第一个字母，就能智能提示，可快速录入。

2. 可视化编程：易语言在编写程序过程中，可即时看到当前程序运行流程及路线，有助于培养编程思路，提高解决编程问题的能力。而且代码即文档、命令格式统一，语法格式自动检查，当编写代码后回车，无论大小写、标点、空格都会进行预编译处理，转换为标准格式，

如果发现错误，会及时提示。

3. 全程提示帮助：编程时随时查阅相关信息，将光标移动到某程序行上，按 F1 帮助键，能够在状态夹提示中获得有关信息。易语言还有参数引导技术，当编写命令时，按 Alt 键+右方向键，可自动展开该命令的参数，以便书写代码，减少用户记忆。

当然，易语言还存在一些不足，如：用易语言编写的程序相对较大，这可能影响它的效率；另外易语言编写的程序，有可能遭到杀毒软件的误杀等等。

自学易语言，除了《易语言编程系统》和《中文编程——从入门到精通》两本书外，网上还有许多视频教程，其中世恒、世宝、觅风三人的教材很精彩：大连世恒的视频教材《易语言百集培训》，大约录制于 2008 年，是培训班的收费教材，淘宝网上光盘售价 900 多元，不过网上有前 40 集的破解版本。对易语言的组件、命令分集讲解，比较详细，遗憾的是只能看到前 40 集。天津世宝的视频教材《易语言教程》，大约录制于 2009 年，分春、冬两季，共 40 课，第一季 22 课，第二季 18 课，全部免费。第一季课程，通过“泛世纪”软件，实现了远程互动教学，灵活多变，接近现场教学；第二季是录像。不过我们现在看到的都是录像了。觅风的视频教材《零基础易语言入门系列视频》，2012 年录制，讲解的是易语言 5.11 版本，前 30 集是免费的，后 12 集网上也可搜到，42 集后的暂时没有。每集时间不超过 30 分钟，由浅入深，简洁明了，便于思考、理解。

当我们沏一杯热茶，坐在电脑边，看着易语言视频教程，间或打开《中文编程——从入门到精通》查阅，何不悠哉乐哉！



易语言创始人——吴涛

学习中文编程——易语言（三）：易语言工作界面

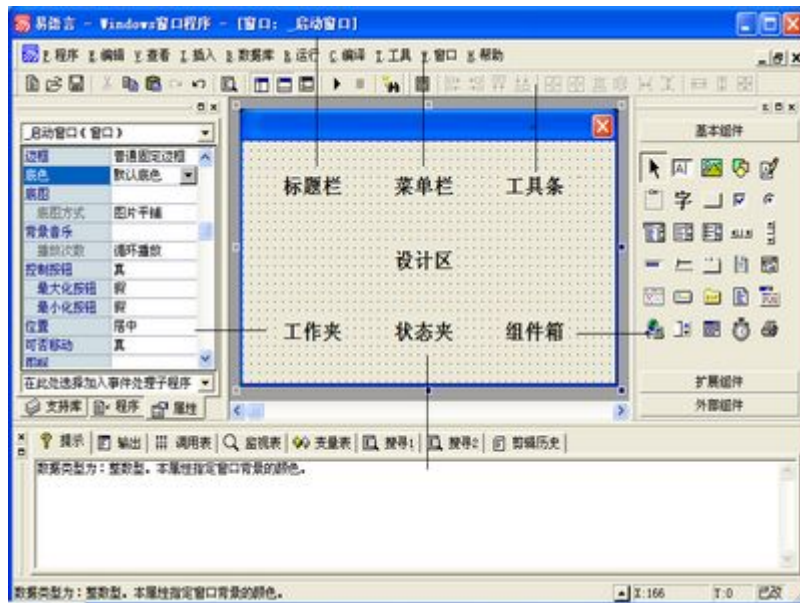
在网上下载“易语言 5.11”，按照软件提示安装。

运行易语言，首先会弹出易语言启动对话框，可创建 4 种 Windows 空白程序：Windows 窗口程序、Windows 控制台程序、Windows 动态链接库、Windows 易语言模块，还可以从左侧选择 Linux 空白程序、通过向导创建、打开易语言例程等选项。勾选“启动后不再显示此窗口”选择框，下次启动就不再显示易语言启动对话框。



易语言启动对话框

默认选择 Windows 窗口程序，单击“确定”按钮，会创建一个 Windows 窗口程序，并打开易语言主界面。



易语言工作界面

易语言主界面最上面是标题栏，显示这个打开窗口的标题。

标题栏下面是菜单栏，有程序、编辑、查看、插入、数据库、运行、编译、工具、窗口、帮助十个菜单，每个菜单下还有子菜单，包含易语言的全部功能。

菜单栏下面是工具条，以图标按钮形式组合，分为标准工具条和对齐工具条二类，是常用功能的快捷按钮。

工具条下面左侧是工作夹，分三个面板：1. 支持库，显示支持库列表，展开后查看各支持库提供的命令、数据类型等信息，编辑状态下，双击某个命令，可将其直接填充到光标处。2. 程序，可以添加程序数据、窗口、资源等。3. 属性，由组件列表、属性列表、事件列表组成，可分别选择列表中的组件、查看或更改组件属性、生成组件事件子程序。

工具条下面右侧是组件箱，分基本组件、扩展组件、外部组件

三部分，如果只显示基本组件，可依次单击菜单工具——支持库配置——全选按钮——确定按钮，就可全部显示。

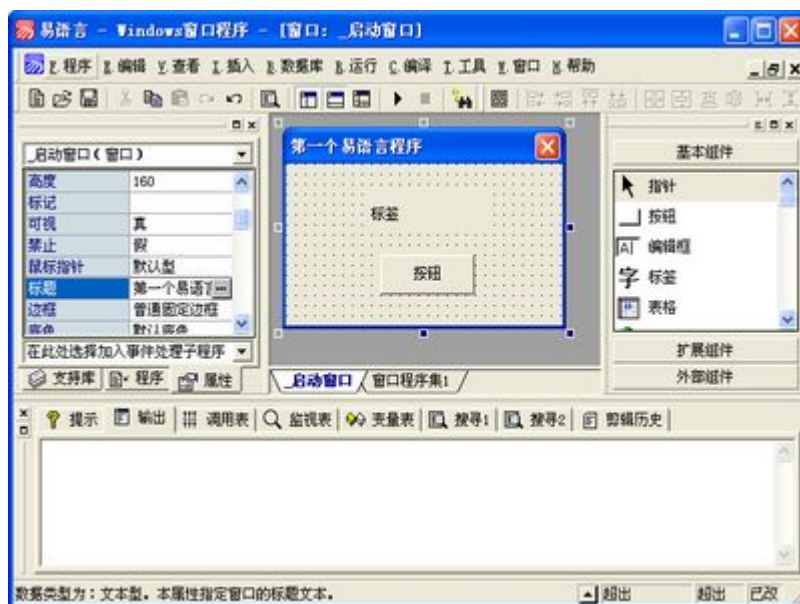
工作夹和组件箱中间是设计区，窗口设计时，可向窗口中添加组件，进行程序界面设计。在程序代码编辑状态下，可录入、修改程序代码。

最下面是状态夹，有提示、输出、调用表、监视表、变更表、搜索 1、搜索 2、剪辑历史八个选项，可以查看帮助信息，查看调试文本等。

熟悉了易语言编程软件的界面，就可以开始编写程序了。

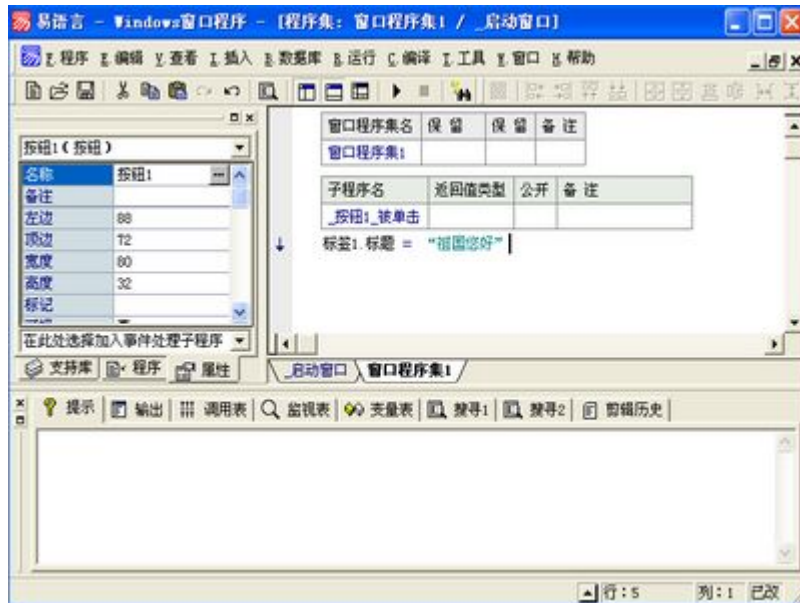
学习中文编程——易语言（四）：第一个易语言程序

运行易语言，建立一个 Windows 窗口程序，程序自动命名为“_启动窗口”，在工作夹属性面板中将标题改为：第一个易语言程序。点击组件箱中标签组件，在窗口中按鼠标左键画一个标签，再点击组件箱中按钮组件，在窗口中按鼠标左键画一个按钮，程序窗口设计完毕。



第一个易语言程序的_启动窗口

双击按钮，进入“窗口程序集 1”界面，在“_按钮 1_被单击”子程序下编写代码：标签 1. 标题 = “祖国您好”，程序代码编写完毕。



第一个易语言程序的窗口程序集

点击易语言工具条中“运行”按钮，或按下 F5 键来运行程序，在出现的窗口中点击按钮，标签显示出“祖国您好”。



第一个易语言程序“祖国您好”

最后点击菜单“编译”，制作成 EXE 可执行文件或安装文件发布。

分析第一个易语言程序，当运行时弹出一个窗口，这个窗口就是新建易语言程序时的“_启动窗口”，窗口中的标签和按钮是窗口程序设计时添加的。当用鼠标点击按钮时，就运行了“_按钮 1_被单击”子程序，而这个子程序运行了程序里的代码“标签 1.标题= ‘祖国您好’”，改变了标签的标题，显示出“祖国您好”，程序运行完毕。

据说“祖国您好”这个程序，是易语言创始人吴涛用易语言编写的第一个程序，后来有关易语言的各种教程，大多把“祖国您好”这个程序，作为第一个易语言程序实例讲解。

第一个易语言程“祖国您好”，诠释着中文汉语编程的深远意义。

学习中文编程——易语言（五）：窗口、集程序和子程序

从第一个易语言程序“祖国您好”中，可以看到这个程序由窗口和程序集两部分组成。

1. 窗口

窗口是一个程序的载体，当用易语言建立一个 Windows 窗口程序时，会自动生成一个“_启动窗口”，这个窗口的名称是不能改变的，否则，运行中会出现错误。通过组件箱，在窗口中添加不同的组件，根据需要，调整布局，对窗口进行设计，这个窗口就是所写程序的界面。



易语言窗口

2. 程序集

每个窗口对应一个程序集，默认的名称是“窗口程序集 X”，“X”是从 1 开始的序号，程序集的名称可以更改。程序集中的内容有表格和文字代码据两种，简单说：表格是用来定义程序数据的，而文字

代码是用来描述程序中命令、属性、事件的。

一个程序可以由多个窗口和程序集组成，还可以插入类模块、自定义数据类型、全局变量、DLL 命令、常量、资源。

3. 子程序

每个程序集中有若干个子程序，子程序用来封装一系列命令，实现模块化，重复调用，使程序更清晰、合理。

(1) 子程序有“返回值类型”、“公开”和“备注”三个属性：

子程序的返回值类型：是指子程序执行后结果值的类型，用于子程序之间的联系。如果程序代码不需要子程序的返回值，可以不定义返回值。

子程序的公开属性：子程序公开，可以全局调用，在任何子程序中都能用；子程序不公开，只能局部调用，只能在本程序集中用。

子程序的备注属性：是对子程序的注释，不参与运行，只在编程时参考。

(2) 子程序分“事件子程序”和“用户自定义子程序”。

事件子程序：是指对应组件所发生事件生成的子程序。可以在这些事件子程序中写入执行代码，运行时，一旦这些事件产生，就会执行相应的子。事件子程序的名称、参数、返回值都是系统定义的，不允许修改。

用户自定义子程序：是指由用户创建，其参数和返回值由用户自行定义的子程序。用户自定义子程序的名称、参数、返回值，可以根据需要在程序设计时任意修改。

(3) 子程序的参数：参数是指供命令进行判断、选择或再加工的因素，多个参数用逗号分开。子程序每个参数都存在“参考”、“可空”、“数组”、“备注”四个属性，决定子程序参数的几种特性。

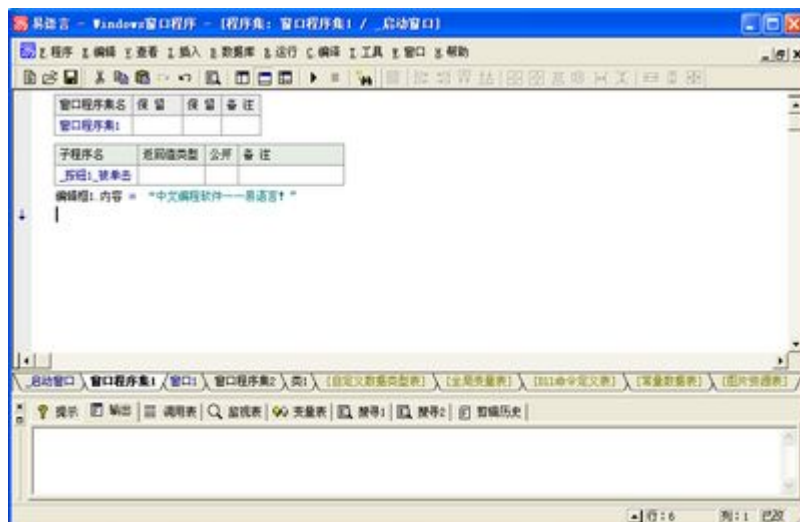
参数参考属性是指：参数是指向变量数据的指针，既参数实际保存的是变量地址，这个地址的内容就是变量的值。编程时只要记住如果提供给子程序参数的变量具有参考属性，那么在子程序内部对该参数的操作，相当于直接对变量的操作。

参数可空属性是指：子程序的参数在调用时是否可以省略该参数，具有可空属性，表示调用该子程序可以省略该参数；不具有可空属性，表示调用该子程序必须提供该参数。当参数可空时，查找帮助，可省略的参数用“[]”括起来来。

参数数组属性是指：子程序的参数是一个数组变量。

参数备注属性是指：对参数的注释，只在编程时参考。

(4) 子程序的调用：是指重复执行某一子程序时，直接用命令调用已存在的子程序。子程序调用的命令是：“XXXXXX ()”，XXXXXX表示子程序的名称。子程序的调用，能减少重复代码的编写，优化程序代码。



易语言集程序及子程序

4. 易语言程序语句

易语言程序集和子程序中的代码描述称为易语言程序语句，常用的程序语句分三种：

(1) 赋值型语句，基本形式是：“被赋值的对象 = 用于赋值的数值”，即将等号后面的值赋给等号前面的值。如：标签 1. 标题=“祖国您好”。

(2) 命令语句，基本形式是：“命令名 ()”、“组件名. 命令名 ()”，有组件名的命令又称为方法。如：编辑框 1. 加入文本 (“中文编程软件——易语言!”)。

(3) 非运行语句，在运行中不被运行的语句，以单引号为前缀，例如注释型语句。

学习中文编程——易语言（六）：数据类型

数据：是指能够输入致计算机，并被计算机识别和加工处理的符号的集合。数据按一定的标准分为不同的类型，易语言称其数据类型。易语言的数据类型从数据结构区分，可分为基本数据类型、特殊数据类型。

1. 基本数据类型

基本数据类型有 6 种：数值型、逻辑型、日期时间型、文本型、字节集型、子程序指针型。

数值型又包括：字节型、短整数型、整数型、长整数型、小数型、双精度小数型。

易语言数据类型的长度和存储的值域

名称	长度 bit	占用字节	取值范围
字节型	8	1	0——255
短整数型	16	2	-32,768——32,767
整数型	32	4	-2,147,483,648——2,147,483,647
长整数型	64	8	-9,223,372,036,854,775,808—— 9,223,372,036,854,775,807
小数型	32	4	-3.4E38——3.4E38（7 位小数）
双精度小数型	64	8	-1.7E308——1.7E308（15 位小数）
逻辑型	32	4	“真”或“假”
日期时间型	64	8	100 年 1 月 1 日——9999 年 12 月 31 日
子程序指针型	32	4	与外部程序或操作系统 API 进行交互，是一个子程序在内存中的地址
文本型			由以字节 0 结束的一系列字符组成
字节集			一段字节集型数据串

各种类型的数据都在内存中占用一定的存储空间，字节是计算机数据处理的基本单位，一个字节由 8 个二进制位构成，即 8 个比特（1Byte=8bit）。如：字节型取值范围：0——255，是二进制 00000000——11111111 转化为十进制的数据；整数型取值范围是：-2,147,483,648——2,147,483,647，是二进制 $-1E32$ —— $1E32-1$ （ -1×10^{32} —— $1 \times 10^{32}-1$ ）转化为十进制的数据。

如果给数据类型赋值超出其取值范围，就会发生数据溢出，如：给字节型数据赋值 266，就会出现错误。

数据类型所占字节数越多，容纳数据数量越大。占用内存也越大。因此，在选择数据类型时，既要避免空间浪费，又要防止数据溢出。

2. 特殊数据

易语言特殊数据类型是指基本数据类型之外的其它数据类型。包括通用型数据类型、库定义数据类型、内部组件数据类型、自定义数据类型。

通用型数据类型：仅在系统内部使用，能够匹配所有的基本数据类型、库定义数据类型、自定义数据类型。

库定义数据类型：由易语言支持库提供，可直接使用，同基本数据类型一样。

内部组件数据类型：易语言的每一种内部组件，都可以作为一种数据类型使用。

自定义数据类型：用户可以随时在程序中自行定义新的数据类

型，自定义数据类型时需要设置数据类型的名称及其成员，其中数据类型成员各属性设置方法与变量设置方法相同。



工作夹程序面板中自定义数据

3. 数据类型的转换

在编写程序时，有时需要对数据类型比较，相同数据类型之间可以直接比较，不同数据类型之间需要先转换再比较。

易语言中常用的数据类型间转换的命令有：

“到数值（）”命令，用来将一个通用型数据转换到双精度小数型。

“到小数（）”命令，用来将一个通用型数据转换到小数型。

“到整数（）”命令，用来将一个通用型数据转换到整数型。

“到文本（）”命令，用来将一个通用型数据转换到文本型。

“到字节集（）”命令，用来将一个通用型数据转换到字节集型。

“到时间（）”命令，用来将一个文本型的数据转换成日期时间型。

学习中文编程——易语言（七）：变量

计算机一般以两种方式将程序中用到数据存储在内存中，其中一种是变量。可以取不同数值的量称为变量，变是绝对的，没有固定的值，是可以改变的数。

可以把变量理解成一个容纳物品的容器，只是这个变量容纳的是各种可变的数据。变量容纳的数据可以提取，也可以改变。

变量的名称可以任意定义，不过在编程中，应该根据需要给变量定义一个有实际意义的名称，方便程序的开发、维护和互相交流。

1. 变量分类

变量从作用范围分三类：

全局变量，程序运行后，所有程序集都可以使用的变量。全局变量程序运行后即占用内存空间，程序结束后才从内存中清除，会较长时间占用系统资源。

程序集变量，仅在本程序集中被调用，若在其它程序集中调用，则需要在变量名前加程序集对应的窗口名称前缀。如：“信息框（_启动窗口.程序集变量，0，）”。

局部变量，只在所在子程序中被调用的变量，只有子程序被调用才占用系统资源，子程序结束，变量所占内存就被系统收回，非常节省系统资源。

2. 静态变量

静态变量，是指静止存在的局部变量。当子程序退出时，静态局部变量能够保留现行内容，以供下次继续使用；非静态局部变量，下

次进入子程序时将被初始化。

局部变量如果不设置“静态”属性，子程序执行完毕后将清空所有非静态局部变量；局部变量如果设置了“静态”属性，子程序执行完毕后将不会被清空，当子程序再次调用时，静态变量仍然保持上次被调用时的状态。静态变量在子程序首次被调用时分配内存，在程序结束时销毁。

3. 数组变量

数组变量。是指可以存放一组数据的变量。数组变量中的每个成员拥有独立的存储单元，可以单独调用和赋值。数组变量可以看作是多个非数组变量的组合，每个成员由方括号“[]”括住从“1”开始的自然数表示，称为“数组成员下标”。

数组变量又分为“单维数组变量”和“多维数组变量”：

单维数组变量的表示形式是：“数组变量名[数组成员下标]”，如：“变量[2]”，表示一个数组中的第 2 个成员。

多维数组变量可以看作是多个单维变量组合，表示形式是：“数组变量名[数组成员下标][数组成员下标]”如：“变量[1][2]”，表示一个二维数组中第 2 个成员。

4. 变量定义

(1) 定义全局变量：用“Ctrl+G”，新建一个全局变量；还可以在易语言“插入”菜单中选择“全局变量”来插入全局变量。

(2) 定义程序集变量：将光标停留在编辑区“窗口程序集”中任意位置，按回车键，建立一个程序集变量。

(3) 定义局部变量：在子程序处使用“Ctrl+L”，新建一个局部变量；还可以在易语言“插入”菜单中选择“局部变量”来插入局部变量。

(4) 定义静态变量：在子程序定义变量的表格中，勾选“静态”下的空格，出现“√”，定义成功，再次单击，“√”消失，取消静态属性。

(5) 定义数组：定义一个变量，将变量名称命名为“数组”，然后将光标指向这一行的“数组”，输入要定义的成员数，多维数组成员数，要用逗号隔开。如图：

窗口程序集名	保 留	保 留	备 注
窗口程序集1			

子程序名	返回值类型	公开	备 注
__启动窗口_创建完毕			

变量名	类 型	静态	数组	备 注
数组	整数型		3	

定义一个单维数组

窗口程序集名	保 留	保 留	备 注
窗口程序集1			

子程序名	返回值类型	公开	备 注
__启动窗口_创建完毕			

变量名	类 型	静态	数组	备 注
数组	整数型		3,3,3	

定义一个多维（三维）数组

变量赋值

给变量赋值可以使用“赋值（）”命令，给数组变量赋值可以使用“连续赋值（）”命令；也可以使用等号（=）给变量和数组变量赋值。

给变量或数组变量赋值时，要注意赋给相应的数据类型，否则会出现错误。

学习中文编程——易语言（八）：常量

计算机一般以两种方式将程序中用到数据存储在内存中，其中一种是常量。固定不变的量称为常量，常量是相对的，不变是相对的，没有绝对意义上的常量。

常量是一个固定的值。可用于定义在程序中多个位置使用的值，此值在正常情况下不能更改。

常量与变量不同，在易语言程序使用时有专门的处理方法。

1. 自定义常量

常量只能在程序设计时定义，定义格式由常量名称和常量值组成。

定义常量，可以双击工作夹程序面板中的“常量表”一项，在设计区会出现“常量数据表”；也可以在设计区通过鼠标右键选择“新常量”或使用快捷键“Ctrl+N”方式建立。点击“常量数据表”任意地方即可出现空白常量行，在空白常量行分别输入常量名称、常量值、备注（可省略）等信息。

常量的类型有数值型、文本型、逻辑型、日期时间型，从易语言 4.03 版本开始 新增了长文本型。

定义长文本常量，可以在设计区通过鼠标右键选择“新长文本常量”建立，点击常量值下的“<文本长度：0>”，弹出“请输入文本”对话框，输入或导入文本，点出“确认”按钮保存，可看到文本长度。。长文本常量最大支持 30000 字节。

2. 常量的调用方式

常量在代码中调用方式为：#常量名。

3. ASCLL 码

ASCLL 码是计算机与因特网中最普遍的文字档案格式，是一种通用的常量，由控制字符和 ASCII 值组成。ASCLL 码与计算机键盘对应，如：字符“A”的 ASCLL 值是 89、字符“a”的 ASCLL 值是 97。

ASCII 码表

ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符
0	NUL	32	(space)	64	@	96	,
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	X	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	:	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

4. 支持库常量

易语言核心支持库、扩展支持库也定义了许多常量。可在易语言工作夹支持库面板中对各支持库常量进行查询：展开一个支持库，如

果有“常量”选项，点击“常量”，展开常量列表，按下“F1”键，在状态夹提示框中，可看到该常量的帮助信息，并可查到常量的值。

系统核心支持库提供的常量是使用频率最高的常量，主要有：

(1) 颜色常量：易语言提供了 33 个颜色常量，可以用命令调用，调入时输入“#颜色名”即可。有颜色属性的组件，有一个颜色选择器，用来直接改变颜色，其值都可以作为常量。

(2) “#换行符”：如果想让编辑框显示的一段文本自动换行，就需要换行符。“#换行符”常量等同于文本<回车>+<换行>。

(3) “#引号”、“#左引号”、“#右引号”：为了不和代码中表示文本数据的引号冲突，程序代码中将文本中的引号作为一个文本常量。如果让编辑框显示出一个有引号的文本，就要使用常量“#左引号”、“#右引号”，如：编辑框 1. 内空 = #左引号 + “易语言” + #右引号。

(4) 键代码常量：易语言将键盘上常用按键的键代码都作为核心支持库定义的常量，在程序使用时，“#”+要调用的键名。如：键盘上的 F8 的键代码，用常量表示为“#F8”。

(5) 用常量填写参数：常量除了直接调用外，还可以作为命令的参数。很多命令的参数可以直接使用常量，这样使程序看起来更为直观。

5. 枚举常量

枚举常量是一个常量的集合，将多个常量以成员的形式，存放在一个常量中，使用格式是：“#枚举常量名. 成员名”。

枚举常量是一种常量的表现形式，是由易语言支持库定义的常量集合，不能自定义。易语言很多支持库中使用了枚举常量，如核心支持库中定义的“变体类型”。“变体类型”提供变体型中所能容纳数据类型的枚举值。

变体类型常量成员及常量值

成员	常量值描述	成员	常量值描述
未知	-1	数值型数组	7
空	0	文本型数组	8
数值型	1	逻辑型数组	9
文本型	2	日期型数组	10
逻辑型	3	对象型数组	11
日期型	4	错误值型数组	12
对象型	5	变体型数组	13
错误值型	6		

学习中文编程——易语言（九）：资源表

在编程的时候，经常会用到图片和声音文件，在程序发布的时候，需要将使用的图片、声音文件一起提供软件才能正常使用。易语言提供资源表，用来存储、打包这些图片和声音文件。

资源表存放程序开发中的图片、声音等二进制数据资源。添加后的资源是字节集型数据，这些文件保存在程序内部，保护这些文件不被复制和修改。

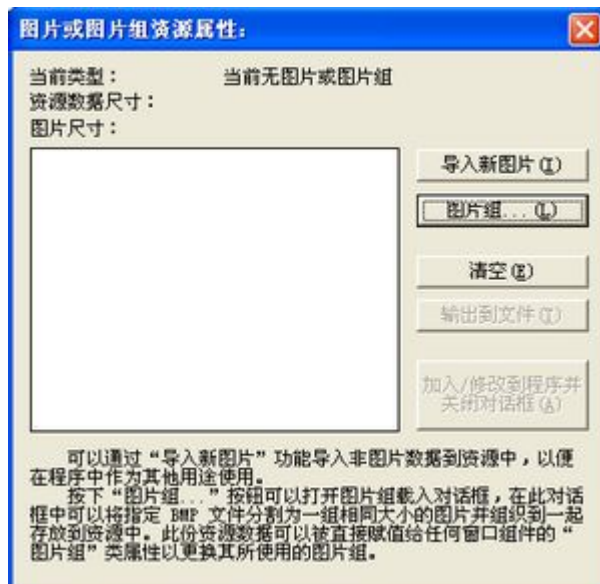
1. 添加图片或图片组资源

添加图片或图片组资源的方法是：双击工作夹程序面板资源表中的“图片或图片组”选项，切换到“图片资源表”界面，直接回车，在“图片或图片组名称”下的表格中输入名称，然后在“内容”下的表格双击，弹出一个“图片或图片组资源属性”对话框，点击“导入新图片”按钮，按提示导入图片文件；点击“图片组”按钮，进入“图片组设置”对话框，进行图片组设置操作。

重复以上操作，可以添加多张或多组图片。



图片资源表界面



图片或图片组资源属性对话框



图片组设置对话框

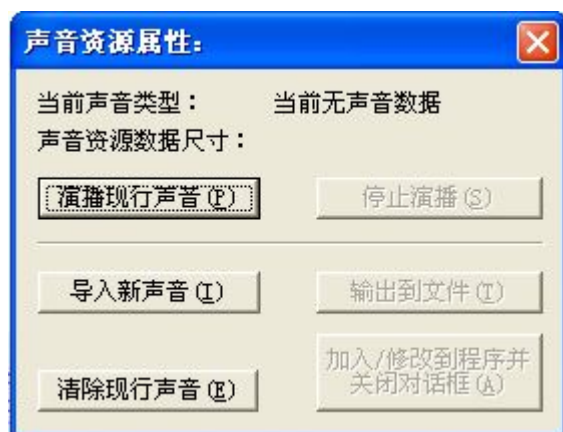
2. 添加声音资源

添加声音资源的方法是：双击工作夹程序面板资源表中的“声音”选项，切换到“声音资源表”界面，直接回车，在“声音名称”下的表格中输入名称，然后在“内容”下的表格双击，弹出一个“声音资源属性”对话框，点击“导入新声音”按钮，按提示导入声音文件。

重复以上操作，可以添加多。声音资源只支持“MID”和“WAV”格式的声音文件。



声音资源表界面



声音资源属性对话框

3. 资源表中导入可执行文件

资源表中还可以导入任意文件，易语言不会判断导入的文件是否是图片或声音。添加任意文件的方法，在我的电脑上，以图片资源表导入“ere”文件为例：双击工作夹程序面板资源表中的“图片或图片组”选项，切换到“图片资源表”界面，直接回车，弹出一个“图片或图片组资源属性”对话框，在“图片或图片组名称”下的表格中输入“

“文件”，然后在“内容”下的表格双击，点击“图片或图片组资源属性”对话框中“导入新图片”按钮，弹出“请输入图片文件名”对话框，在“文件类型”下拉列表框中选择“所有文件 (*.*)”，双击“易语言学习”文件打开，将该文件导入到资源表。点击“打开/修改到程序并关闭对话框”按钮，图片资源表界面内容表格显示“20”，表示该文件大小是 20 个字节。

用“写到文件（）”命令写出，并用“运行（）”命令运行。

用“声音资源表”导入任意文件的方法相同。



请

输入图片文件名对话框

图片或图片组名称	内容	公开	备注
文件	20		1

启动窗口 图片资源表

加入文本的图片资源表

学习中文编程——易语言（十）：运算符和表达式

1. 易语言运算符

易语言程序进行算术运算或关系比较操作时，需要使用运算符。易语言运算符如下：

易语言运算符

运算符分类	运算符	运算符含义	代码显示
算术运算符	+	加法运算，将加号两边的数相加	+
	-	减法运算，将减号两边的数相减：负	-
	*	乘法运算，将乘号两边的数相乘	×
	/	除法运算，将除号两边的数相除	÷
	\	除法运算，将除号两边的数相除	\
	%	整除运算，将整除号两边的数整除	%
关系运算符	>	判断是否大于	>
	<	判断是否小于	<
	=	判断是否等于	=
	>=	判断是否大于等于	≥
	<=	判断是否小于等于	≤
	<>、!=	判断是否不等于	≠
	?=	判断是否约等于	≈
逻辑运算符	&&、qie	逻辑与运算符，同时连接几个	且
	、huo	逻辑或运算符，可选连接几个	或
	取反（）	逻辑非运算符	取反
赋值运算符	=	将等号后面的值赋给等号前面的对象	=

程序中的运算符有其优先级别，在程序运行时按照符号的优先级别从高到低依次运算。易语言运算符优先级别如下：

易语言运算符优先级

运算符	优先级
() 小括号	1
* 乘、/ 除	2
\ 整除	3
% 求余	4
+ 加、- 减	5
> 大于、≥ 大于等于、< 小于、≤ 小于等于、 = 等于、!= 不等于、?= 约等于	6
&& 逻辑与	7
逻辑或	8
= 赋值	9

2. 易语言表达式

用运算符和括号将作为对象的各类数据连接起来，符合易语言语法规则的式子，称易语言表达式。

下面是一个表达式：变量 1=((3×9-15÷3)-2)\8。

表达式中运算先后，是按照运算符的优先级别来的，计算步骤和结果是：

变量 1=((3×9-15÷3)-2)\8 第 1 步：内层小括号内
乘除运算

变量 1=((27-5)-2)\8 第 2 步：内层小
括号内减法运算

变量 1=(22-2)\8 第 3 步：外
层小括号内减法运算

变量 1=20\8 第 4 步：
20 被 8 整除

变量 1=2

结果

等于 2

运算级可以通过括号改变，括号中的表达式先计算；当出现括号嵌套时，最内层括号最先计算；括号内的运算符优先级不变。

3. 赋值运算符和赋值表达式

等号（=）是赋值运算符，在程序中给变量赋值或用代码改变组件属性，使用“=”进行赋值，将等号后面的值赋值给等号前面的赋值对象。如：“变量 1=10”、“编辑框 1. 高度=20”。

赋值运算符“=”和比较运算符“=”区别是：一般在易语言中的一条语句中，不被小括号包括的最左边个“=”代表赋值，且一条语句只有一个赋值，其它的“=”都是比较命令“等于”。

以上几篇博文，介绍易语言编程软件的基础知识。当我们粗略地学习完这些基础知识，好像对易语言有个模糊的认识了。



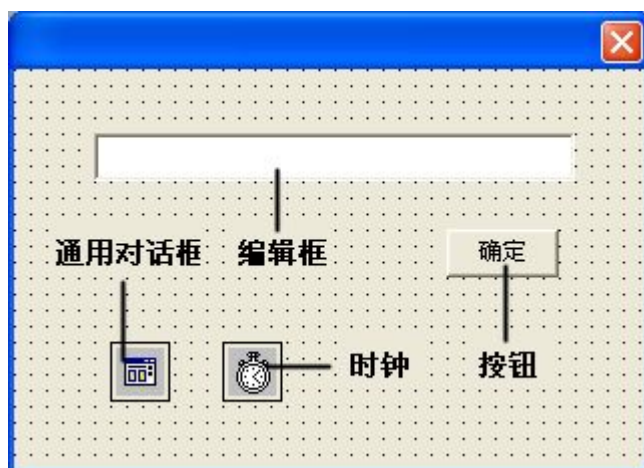
好像对易语言有个模糊的认识了

学习中文编程——易语言（十一）：易语言组件.1

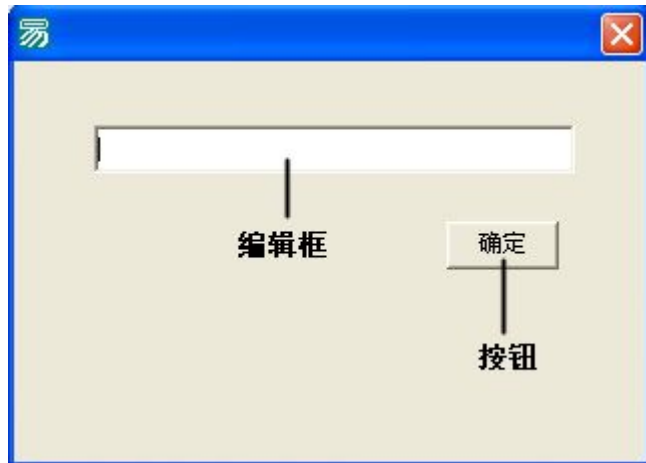
1. 易语言组件的概念

易语言组件是程序界面可视化操作的基本单元，也是程序中主要的输入。输出部件。所有的组件都是一个独立的对象，拥有自己的属性、事件和方法。

一部分组件是可见的，如：编辑框、按钮等；还有一部分组件是不可见的，如：通用对话框、时钟等。



在窗口中添加编辑框、按钮两个可见组件和通用对话框、时钟两个不可见组件



程序运行后只能见看到编辑框、按钮两个可见组件

易语言在程序开发中，要用到不同的组件，每个组件的实质是一种特殊的窗口，所以又叫窗口组件。不同的组件，互相配合，分别实现不同的功能。

2. 易语言组件的分类

易语言的组件分基本组件、扩展组件、外部组件三类。基本组件是最主要的、使用频率最高的组件，有 40 个；扩展组件数量众多，有 81 个，当基本组件不能实现一些功能时，可试一下扩展组件；外部组件目前只有 Windows 播放器 1 个组件。

易语言组件属核心支持库。核心库内置数据类型，包含成员和命令，与基本数据类型定义方法相同，有字体和打印设置信息。

3. 易语言组件的属性、事件、命令

(1) 组件的属性

属性：用于表示窗口组件的功能性质及与其它对象之间的关系。每个组件都有共有属性和独有属性。

共有属性是所有组件共同具有的基本属性，主要表示组件的名称、备注、相对位置、大小等。

独有属性是不同组件特有的属性，显示不同组件的特有功能。

（2）组件的事件

事件：类似一个子程序，当程序运行接受到组件某些信息时，系统会自动调用这些子程序。只有可见组件有共有事件，大部分组件有独有事件，还有一些组件没有事件。

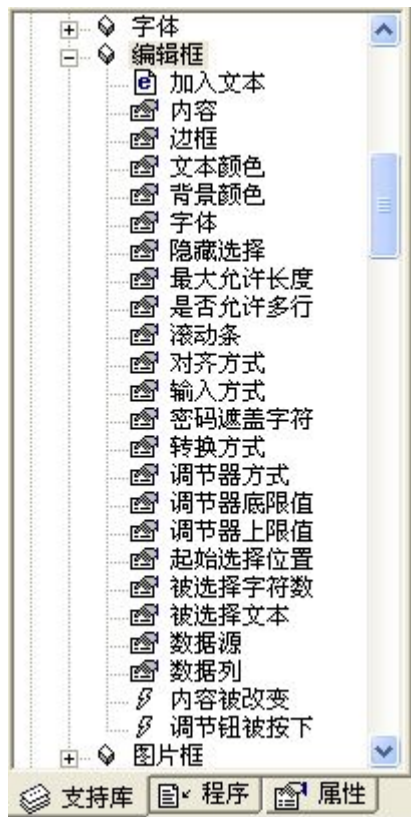
共有事件，可见组件运行时都有一些相同的事件，和鼠标、键盘相关，即共有事件。

独有事件，不同的组件有不同的事件。

（3）组件的命令

命令：可以简单理解为一个功能调用的代码，对应窗口组件的命令又称方法。窗口与组件是同源产物，窗口拥有的命令所有组件都可以使用，因此是共有命令。有的组件有自己的独有命令。

在易语言主界面工作夹支持库面板中，可以查到所有组件的独有属性、事件、命令。右击窗口中的任一组件，再点击“查看数据类型定义”，选择工作夹支持库面板，有“e”字的图标是组件的独有命令，有“手指”的图标是组件的独有属性，有“闪电”的图标是组件的独有事件。



易语言主界面工作夹支持库面板

学习中文编程——易语言（十二）：易语言组件.2

基本组件简介

基本组件按其功能可作以下分类：

1. 窗口：窗口是 Windows 程序中最基础的显示组件，是其它组件存放的容器。易语言组件箱中没有“窗口”这个组件，在默认设置下，每新建一个易语言程序，就会自动生成一个“_启动窗口”，通过菜单还可以插入一个新窗口。由于窗口是其它组件的载体，窗口的属性、事件和命令，直接影响其它组件。

2. 菜单：菜单是窗口实现不同功能的列表组合，能让程序的操作简单、方便。易语言用“菜单编辑器”创建菜单，选择易语言菜单“工具”——“菜单编辑器”；或在窗体单击鼠标右键，选择右键菜单“菜单编辑器”；还可以使用快捷键“Ctrl + E”调出“菜单编辑器”。

3. 按钮类组件：用于鼠标点击时执行各种操作。有按钮、图形按钮。

4. 列表类组件：显示多条信息和数据，供选择、处理。有组合框、列表框、选择列表框。

5. 系统类组件：对计算机硬盘中的文件进行管理。有通用对话框、文件框、目录框、驱动器框。

6. 图形、影像类组件：对图形、影像文件显示和处理。有图片框、画板、颜色选择器、影像框。

7. 分组类组件：对窗口组件进行分组管理，或对操作对象进行选择。有分组框、外形框、选择框、单选框、选择夹。

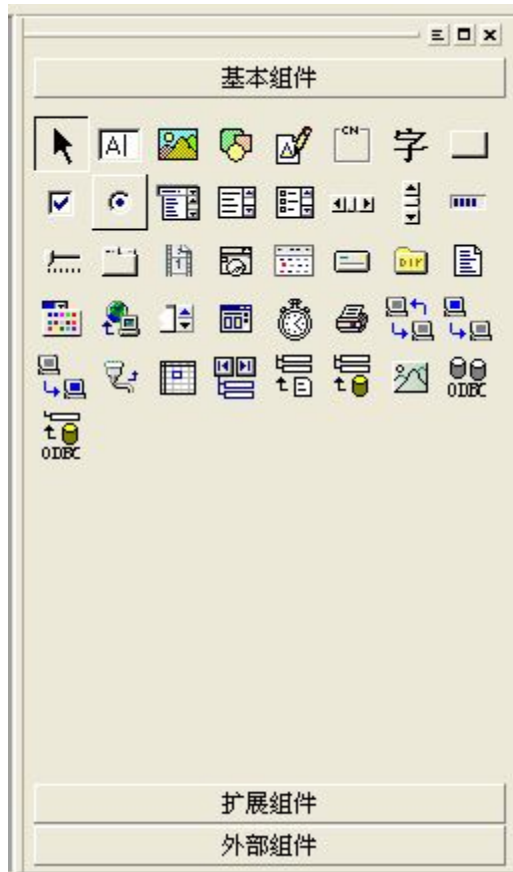
8. 位置控制类组件：对程序进度、内容进行控制和显示。有进度条、滑块条、横向滚动条、纵向滚动条、调节器。

9. 网络、通信组件：主要用于网络之间的通信、链接。有客户、服务器、端口、数据报、超级链接框。

10. 时间类组件：用于显示、处理时间、日期。有时钟、月历、日期框。

11. 显示类组件：主要用于显示、编辑文本数据，打印机用于输出各类数据。有编辑框、标签、表格、打印机。

12. 数据库类组件：用于处理内部和外部数据库数据。有数据库提供者、数据源、通用提供者。



易语言基本组件

常用扩展组件简介

易语言的扩展组件有 80 多个，其中有一些类型的组件是经常用到的：

1. 文字显示类组件：用于输入、显示和编辑单项文本数据。有透明标签、超级编辑框。
2. 列表类组件：显示、编辑金条记录和数据。有树型框、超级列表框。
3. 按钮类组件：主要用于鼠标点击时执行各种操作。有工具条。
4. 办公类组件：用于编辑文字、表格、演示文稿。有办公组件、Word 程序、Word 图形、Word 文档集、Excel 程序、Excel 工作簿、Excel 图表、PPT 程序、PPT 播放、PPT 文稿。
5. 多媒体类组件：用于多媒体文件的显示、播放、处理。有录音、录音音波、媒体播放、CD 播放、高级影像框。
6. 网络类组件：主要用于网络之间的通信。有超文本浏览器、IP 编辑框。
7. 农历日期类组件：用于显示农历日期。有农历日期框、农历月历。
8. 数据库类组件：用于处理内部和外部数据库数据。有数据库连接、记录集。
9. 其它类组件：有分隔条、状态条、气球提示框、每日一帖等等。



易语言扩展组件

扩展组件的功能强大，使用方法稍微复杂一些，但编写程序时确实不可或缺。如：我们使用扩展组件“工具条”，可能创建工具栏，这些图形快捷按钮，比使用菜单栏更直观、方便、快捷。又如：我们可用树形框，创建有父子关系又有平行关系的树形图表。

易语言组件，内容较多，是学习易语言的重点，要学习每一个组件，熟练应用。这时，我们才刚刚踏进易语言编程的大门。



这时，我们才刚刚踏进易语言编程的大门

学习中文编程——易语言（十四）：易语言命令.1

1. 易语言命令的概念

易语言命令是处理问题一段程序封装的计算机指令。为了使用计算机计算、处理一些问题，需要把解决问题的方案用计算机指令表达出来，这些指令就是一段程序，将这些程序进行封装即为命令。一个命令可以由一个或多个简单的命令封装成一个功能更加强大的命令。

当需要相同作用的功能时，只要调用这个命令，就会得到相同的结果。这会简化编程的复杂程度，缩短软件开发周期。

一个软件由各种命令组合而成，不同的命令完成不同的工作。

2. 易语言命令的格式、参数、返回值

（1）易语言命令的格式

易语言命令的完整格式是：“[返回值][所属对象.]命令名称（[参数 1]，[参数 2]，…）”，其中“[]”中表示可以省略的部分。

易语言中大多数命令格式是：“命令名称（）”，其中“（）”中可能有参数，也可能没有参数。

（2）易语言命令的参数

易语言命令的参数，是指调用一个功能时附加的数据、条件或结果。命令的参数可以是一个数据类型的具体数据值，也可以是一个变量，还可以是一个数组变量。但在调用时传递给命令的参数，必须满足该命令对参数的定义，否则会出现错误。

参数写在命令名称后的小括号中，多个参数用逗号隔开。

（3）易语言命令的返回值

大多数易语言命令执行完毕后都有返回值，这些返回值是运算结果、反馈信息、是否成功状态等。大部分时候，当前命令的返回值对后续命令非常重要。一个命令运行成功或失败，会弹出信息框提示。

各命令的语法规则了其返回值的数据类型，在实际应用中，应当根据需要对返回值的数据类型进行转换。

3. 易语言命令的嵌套调用

易语言的命令可以嵌套使用，即一个命令的参数是另一个命令的返回值。如：

“编辑框 1. 内容 = 到文本（到数值（编辑框 1. 内容）+ 1）”，此代码在“到文本（）”命令中嵌套了“到数值（）”命令。

命令的嵌套调用

窗口程序集名	保 留	保 留	备 注
窗口程序集1			

子程序名	返回值类型	公开	备 注
__启动窗口_创建完毕			

编辑框1.内容 = 到文本 (到数值 (编辑框1.内容) + 1)

↑

※被赋值的变量或变量数组： 编辑框1.内容

↑

※用作赋予的值或资源： 到文本 (到数值 (编辑框1.内容) + 1)

易语言的命令大约有 9400 多条，可初步分为 17 类简述。

学习中文编程——易语言（十五）：易语言命令.2

易语言命令简述-1

1. 流程控制命令

可以控制程序运行的路线，如果满足一定条件运行一些代码；不满足条件时，运行另一些代码。此类命令有流程控制线，运行路线一目了然。流程控制命令在易语言中是非常重要的一类命令，多数程序编写离不开这类命令。

流程控制命令分为分支类流程控制命令、循环类流程控制命令、跳转类流程控制命令：

（1）分支类流程控制命令

分支类流程控制命令原型是：“〈无返回值〉 命令名称（逻辑型条件）”，当条件为真时，程序顺序执行后续代码；当条件为假时，程序则沿虚线箭头处停止，或跳到下一行代码执行。

分支类流程控制命令判断的条件可以使用“且”、“或”连接多个条件，最终返回一个逻辑结果来实现多条件联合判断。

分支类流程控制命令包括：如果真（）、如果（）、判断（）3个命令。

●如果真（）命令，条件为真时，运行程序；条件为假，停止运行。

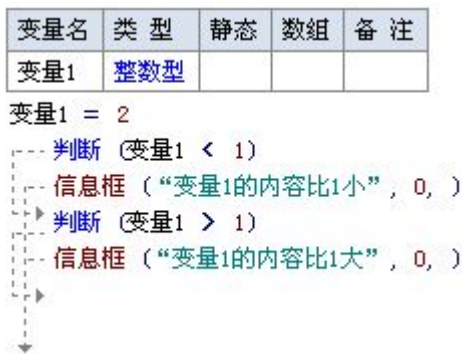
●如果（）命令，条件为真时，运行后续程序；条件为假，沿虚线箭头跳到下一行代码执行。

●判断（）命令，和如果（）命令相同，但在嵌套使用时，判断（）命令结构更加清晰，简略。如下图：

如果（）命令



判断（）命令



(2) 循环类流程控制命令

循环类流程控制命令，可以在一定条件下多次执行一段代码，由循环首和循环尾两部分组成，循环首和循环尾中间是循环块，是循环命令重复执行的代码。

循环类流程控制命令包括：判断循环首（）、循环判断首（）、

计次循环首（）、变量循环首（）4 个命令。

●判断循环首（）命令的原型是：“<无返回值> 判断循环首（逻辑型 条件）- 循环块 - 判断循环尾（）”。

●循环判断首（）命令的原型是：“<无返回值> 循环判断首（）- 循环块 - 循环判断尾（逻辑型 条件）”。

判断循环命令和循环判断命令的区别是：判断循环命令先判断后循环，循环判断命令先循环后判断，由于两个命令判断顺序不同，可能对循环体内的运行造成影响。

●计次循环首（）命令的原型是：“<无返回值> 计次循环首（整数型 循环次数，[整数型变量 已循环次数记录]）- 循环块 - 计次循环尾（）”。

编写一个输出 1——100 的所有整数和的程序，调试输出后的值是 5050。代码如下：

计次循环首（）命令

变量名	类 型	静态	数组	备 注
累加变量	整数型			
计次变量	整数型			

→ 计次循环首 (100, 计次变量)

累加变量 = 累加变量 + 计次变量

--- 计次循环尾 ()

↗ 调试输出 (累加变量)

●变量循环首（）命令的原型是：“<无返回值> 变量循环

首（整数型变量起始值，整数型变量 目标值， 整数型变量 递增值，
[整数型变量 循环变量]）- 循环块 - 变量循环尾（）”。

（3）跳转类流程控制命令

有了跳转类流程控制命令，可以方便的控制程序的流程，节省资源。

跳转类流程控制命令包括：到循环尾（）、跳出循环（）、返回（）、结束（）4 个命令。

●到循环尾（）命令的原型是：“<无返回值> 到循环尾（）”，
当一个循环中运行了到循环尾（）命令，就会直接跳到循环尾代码处。

●跳出循环（）命令的原型是：“<无返回值> 跳出循环（）”，
当一个循环中运行了跳出循环（）命令，当前循环就会结束，然后运行循环体后的程序。

●返回（）命令的原型是：“<无返回值> 返回（[通用型 返回到调运方的值]）”，返回（）命令被执行后，就会退出当前子程序，自动执行子程序后的代码。

●结束（）命令的原型是：“<无返回值> 结束（）”，结束（）命令结束当前程序的运行，实现程序关闭功能。

易语言命令简述-2

2. 算术运算命令

算术运算命令分两类，一是基本算术运算命令，二是扩展算术运算命令。

（1）基本算术运算命令

基本算术运算命令，是程序中常用的命令之一，几乎每个程序都离不开。包括：相加（+）、相减（-）、相乘（*）、相除（/）、整除（\）、求余数（%）6个命令。

●相加（+）命令的原型是：“<通用型> 相加（通用型 被加数或文本或字节集，通用型 加数或文本或字节集，...）”。

相加（+）命令是基本算术运算中唯一一个适用任何基础类型的运算符，当数字运算时用于数字相加，当文本或字节集运算时用于数据相连。如数字运算：123+456=579；文本相连：“ABC ” + “DEF” = “ABCDEF”。

●相减（-）命令的原型是：“<双精度小数型> 相减（双精度小数型 被减数，双精度小数型 减数，...）”。

“-”号还是“负（）”命令的简化运算符，命令的原型是：“<双精度小数型> 负（双精度小数型 数值）”。“相减”与“负”只是使用在不同位置，它们的概念可以互相转换。

●相乘（*）命令的原型是：“<双精度小数型> 相乘（双精度小数型 被乘数，双精度小数型 乘数，...）”。

相乘 (*) 命令会将两个数进行求积运算。

●相除 (/) 命令的原型是：“<双精度小数型> 相除 (双精度小数型 被除数, 双精度小数型 除数, ...)”。

相除 (/) 命令会将两个数进行求商运算。应该注意的是：运算中除数不能为 0，因为数学中除数为 0 没有意义，因此，除数为 0 会提示错误。

●整除 (\) 命令的原型是：“<双精度小数型> 整除 (双精度小数型 被除数, 双精度小数型 除数, ...)”。

整除 (\) 命令会将两个数进行求商运算并舍弃小数部分。应该注意的是：舍弃的小数并非四舍五入，直接舍弃。

●求余数 (%) 命令的原型是：“<双精度小数型> 求余数 (双精度小数型 被除数, 双精度小数型 除数, ...)”。

求余数 (%) 命令会计算出被除数与除数求商运算的余数。

(2) 扩展算术运算命令

扩展算术运算命令是为方便编程提供的，是一些与数学运算相关的命令。包括：取符号 ()、取绝对值 ()、取整 ()、绝对取整 ()、四舍五入 ()、求次方 ()、求平方根 ()、求正弦 ()、求余弦 ()、求正切 ()、求反正切 ()、求自然对数 ()、求反对数 ()、是否运算正确 ()、置随机数种子 ()、取随机数 () 16 个命令。其中置随机数种子 ()、取随机数 () 编程时经常用到：

●置随机数种子 () 命令的原型是：“<无返回值> 置随机数种子 ([整数型 欲置入的种子数值])”

本命令设置产生随机数的基数，如果省略参数，默认是使用当前系统启动时间的毫秒作为基数。

●取随机数（）命令的原型是：：“<整数型> 取随机数（[整数型 欲取随机数的最小值]，[整数型 欲取随机数的最大值]）”。

本命令用于在指定的范围中获取一个随机数。

建立一个易语言程序，在窗口中添加一个编辑框和一个按钮，写如下代码：

取随机数代码

子程序名	返回值类型	公开	备 注
_启动窗口_创建完毕			

置随机数种子 0

子程序名	返回值类型	公开	备 注
_按钮1_被单击			

编辑框1.内容 = 到文本 (取随机数 (1, 100))

运行程序，点击按钮，编辑框中就会随机出现一个大于 1 小于 100 的整数，每按一次按钮，数字随机变化一次。

取随机数



易语言命令简述-3

3. 逻辑比较命令

在易语言中逻辑比较命令只适用于基本数据类型，比较的数据类型必须一致，不一致的需要转换。

● 等于（）命令，运算符号为“=”或“==”，命令的原型是：“<逻辑型> 等于（通用型 被比较值，通用型 比较值）”。

本命令比较两个数据是否一致，如果一致返回“真”，否则返回“假”。

● 不等于（）命令，运算符号为“<>”或“!=”或“≠”，命令的原型是：“<逻辑型> 不等于（通用型 被比较值，通用型 比较值）”。

本命令比较两个数据是否不一致，如果不一致返回“真”，否则返回“假”。

● 小于（）命令，运算符号为“<”，命令的原型是：“<逻辑型> 小于（通用型 被比较值，通用型 比较值）”。

本命令比较一个数据是否小于另一个数，如果是返回“真”，否则返回“假”。

● 大于（）命令，运算符号为“>”，命令的原型是：“<逻辑型> 大于（通用型 被比较值，通用型 比较值）”。

本命令比较一个数据是否大于另一个数，如果是返回“真”，否则返回“假”。

● 小于或等于 () 命令，运算符号为 “<=” 或 “≤”，命令的原型是：“<逻辑型> 小于或等于 (通用型 被比较值, 通用型 比较值)”。

本命令比较一个数据是否小于或等于另一个数，如果是返回“真”，否则返回“假”。

● 大于或等于 () 命令，运算符号为 “>=” 或 “≥”，命令的原型是：“<逻辑型> 大于或等于 (通用型 被比较值, 通用型 比较值)”。

本命令比较一个数据是否大于或等于另一个数，如果是返回“真”，否则返回“假”。

● 近似等于 () 命令，运算符号为 “?=” 或 “≈”，命令的原型是：“<逻辑型> 近似等于 (文本型 被比较文本, 文本型 比较文本)”。

本命令判断一个文本内是否包含另一个文本，如果是返回“真”，否则返回“假”。

● 并且 () 命令，运算符号为 “&&” 或 “And” 或 “且”，命令的原型是：“<逻辑型> 并且 (逻辑型 逻辑值一, 逻辑型 逻辑值二, …)”。

本命令判断两个或几个逻辑值是否都为“真”，如果是返回“真”，否则返回“假”。

● 或者 () 命令，运算符号为 “||” 或 “Or” 或 “或”，命令的原型是：“<逻辑型> 或者 (逻辑型 逻辑值一, 逻辑型 逻辑值

二，…）”。

本命令判断两个或几个逻辑值是否有一个为“真”，如果是返回“真”，否则返回“假”。

● 取反（）命令，命令的原型是：“〈逻辑型〉取反（逻辑型 被反转的逻辑值）”。

本命令用于获取一个逻辑值或逻辑型变量的值的相反值，如果被取反的值为“真”，返回“假”，如果被取反的值为“假”，返回“真”。

如下图：“逻辑变量”为逻辑型，先将“取反（子菜单.选中）”赋值给“逻辑变量”，真的反值为假；再将为“逻辑变量”赋值给“子菜单.选中”，值为真。然后用如果（）命令判断逻辑变量保存的值，以显示不同的信息提示用户的操作。

取反命令（）

变量名	类 型	静态	数组	备 注
逻辑变量	逻辑型			

逻辑变量 = 取反（子菜单.选中）

子菜单.选中 = 逻辑变量

如果（逻辑变量 = 真）

信息框（“子菜单被选中”，#信息图标，“逻辑比较”）

信息框（“子菜单未被选中”，#信息图标，“逻辑比较”）

4. 位运算命令

位运算是指对数据进行二进制的逐位运算。现代计算机都是采用二进制方式存储和处理数据，一个二进制用“比特”（bit）表示，其状态只有 0 或 1，8 个比特=1 个字节。

在计算机内部运算中常用的进位制有 4 种：

二进制，逢 2 进 1，由数字 0 和 1 组成，以下标 2 或后缀 B 表示。

八进制，逢 8 进 1，由数字 0、1、2、3、4、5、6、7 组成，以下标 8 或后缀 Q 表示。

十进制，逢 10 进 1，由数字 0、1、2、3、4、5、6、7、8、9 组成，以下标 10 或后缀 D 表示，该后缀可以省略。

十六进制，逢 16 进 1，由数字 0、1、2、3、4、5、6、7、8、9 和字母 A、B、C、D、E、F 组成，以下标 16 或后缀 H 表示。

要进行位运算，就得将八进制、十进制、十六进制数据转换成二进制。

在易语言中，位运算命令都是针对整数型数据进行操作的，整数型长度为 4 个字节，也就是一组 32 位长度的二进制数。

位运算命令主要包括：位取反（）、位与（）、位或（）、位异或（）、左移（）、右移（）、合并整数（）、合并短整数（）8 个命令。

易语言命令简述-4

5. 数组操作命令

相同数据类型变量的集合即数组。在程序中使用数组能方便管理相关的一组数据，并能使用循环控制一组数据。数组操作命令包括：重定义数组（）、取数组成员数（）、取数组下标（）、复制数组（）、加入成员（）、插入成员（）、删除成员（）、清除数组（）、数组排序（）、数组清零（）10个命令。

6. 环境存取命令

这里的环境指程序运行所在的系统、服务等一系列的设置。环境存取命令包括：取命令行（）、取运行目录（）、取执行文件名（）、读环境变量（）、写环境变量（）5个命令。

7. 拼音处理命令

拼音处理命令是可以方便的处理汉语拼音相关的命令。拼音处理命令包括：取所有发音（）、取发音数目（）、取拼音（）、取生母（）|取韵母（）、发音比较（）、输入字比较（）7个命令。

8. 文本操作命令

对文本型数据进行编辑处理的命令。文本操作命令比较多，包括：取文本长度（）、取文本左边（）、取文本右边（）、取文本中间（）、字符（）、取代码（）、寻找文本（）、倒找文本（）、到大写（）、到小写（）、到全角（）、到半角（）、到文本（）、删首空（）、删尾空（）、删首尾空（）、删全部空（）、文本替换（）、子文本替换

（）、取空白文本（）、取重复文本（）、文本比较（）、分文本（）、指针到文本（）24 个命令。

9. 时间操作命令

用于时间日期编辑处理的命令。包括：到时间（）、增减时间（）、取时间间隔（）、取某月天数（）、时间到文本（）、取时间部分（）、取年份（）、取月份（）、取日（）、取星期几（）、取小时（）、取分钟（）、取秒（）、指定时间（）、取现行时间（）、置现行时间（）、取日期（）、取时间（）18 个命令。

10. 数值转换命令

只适用于数值型数据类型，将现有数值型数据类型转换成指定的数值。包括：到数值（）、数值到大写（）、数值到金额（）、数值到格式文本（）、取十六进制文本（）、取八进制文本（）、到字节（）、到短整数（）、到整数（）、到长整数（）、到小数（）11 个命令。

11. 字节集操作命令

用于对字节集数据编辑处理的命令。字节集即字节的组合，相当于一个字节型数组，字节集和字节型数组可以互换。包括：取字节集长度（）、到字节集（）、取字节集数据（）、取字节集左边（）、取字节集右边（）、取字节集中间（）、寻找字节集（）、倒找字节集（）、字节集替换（）、子字节集替换（）、取空白字节集（）、取重复字节集（）、分割字节集（）、指针到字节集（）14 个命令。

12. 磁盘操作命令

对计算机磁盘上的文件进行管理的命令。磁盘操作命令比较

多，包括：取磁盘总空间（）、取磁盘剩余空间（）、取磁盘卷标（）、置磁盘卷标（）、改变驱动器（）、改变目录（）、取当前目录（）、创建目录（）、删除目录（）、复制文件（）、移动文件（）、删除文件（）、文件更名（）、文件是否存在（）、寻找文件（）、取文件时间（）、取文件尺寸（）、取文件属性（）、置文件属性（）、取临时文件名（）、读入文件（）、写到文件（）22个命令。

13. 文件读写命令

对文件进行编辑处理的命令。文件读写命令比较多，包括：打开文件（）、打开内存文件（）、关闭文件

（）、关闭所有文件（）、锁住文件（）、解锁文件（）、移动读写位置（）、移到文件首（）、移到文件尾（）、读入字节集（）、写出字节集（）、读入文本（）、写出文本（）、读入一行（）、写文本行（）、读入数据（）、写出数据（）、是否在文件尾（）、取读写位置（）、取文件长度（）、插入字节集（）、插入文本（）、插入文本行（）、删除数据（）、打开加密文件（）25个命令。

14. 系统处理命令

对操作系统的剪辑板、窗口、注册表等进行编辑处理的命令。系统处理命令是命令中最多的，包括：运行（）、取剪辑板文本（）、置剪辑板文本（）、剪辑板中可有文本（）、清除剪辑板（）、取屏幕宽度（）、取屏幕高度（）、取鼠标水平位置（）、取鼠标垂直位置（）、取颜色数（）、输入框（）、信息框（）、鸣叫（）、取启动时间（）、置等待鼠标（）、恢复鼠标（）、延时（）、取文本注册项（）、取数

值注册项（）、取字节集注册项（）、写注册项（）、删除注册项（）、注册项是否存在（）、取默认底色（）、快照（）、读配置项（）、写配置项（）、取配置节点名（）、取操作系统类别（）、多文件对话框（）30 个命令。

15. 媒体播放命令

用于播放音频文件的命令。包括：播放音乐（）、停止播放（）、播放 MID（）、播放 MP3（）、同步播放 MP3（）、暂停播放 MP3（）、继续播放 MP3（）、取 MP3 播放状态（）8 个命令。

16. 网络通信命令

用于编辑网络通信的相关命令，包括：取主机名（）、通信测试（）、转换主机名（）、转换为 IP 地址（）4 个命令。

17. 其它命令

其它命令包括：标准输出（）、标准输入（）、载入（）、选择（）、多项选择（）、是否为空（）、是否已创建（）、取数据类型尺寸（）、取颜色值（）、取事件组件（）、事件转移（）、复制窗口组件（）、处理事件（）、载入图片（）、卸载图片（）、取硬盘特征字（）、取系统语言（）、写到内存（）、DLL 命令调用转向（）、置错误提示管理（）、置 DLL 装载目录（）、取组件名（）、取对象类型（）、寻找组件（）、取找到组件数目（）、取所找到组件（）、置入代码（）27 个命令。

以上命令只是易语言常用命令的一部分。易语言命令是学习易

语言又一个重点，当我们通过刻苦学习，能够灵活应用易语言命令，那么我们在踏进易语言编程大门后，在易语言编程的道路上蹒跚漫步了。



在易语言编程的道路上蹒跚漫步了

学习中文编程——易语言（十九）：易语言数据库的构成

要编写比较复杂的程序，就需要用到数据库。

数据库(Database)是按照一定的数据结构来组织、存储和管理数据的仓库。数据库有很多类型，从最简单的存储各种数据的表格，到能够进行海量数据存储的大型数据库系统，在各个方面得到广泛应用。

易语言作为一款中文编程软件，拥有完全自主知识产权和核心技术的数据库。易语言数据库，简称“易数据库”，是一个小型的数据库管理系统，它以一种简单的类似表格的形式组织信息，并形成持久化存储。

1. 易数据库的文件

易数据库的组成包括以下文件：

扩展名 “.edb” 文件是易数据库主体文件。

扩展名 “.edt” 文件是易数据库辅助文件，仅在数据库中存在备注型和字节集型字段时才存在，文件名除了扩展名，和数据库主体文件名相同，而且必须在同一目录下。

扩展名 “.enx” 文件是易数据库索引文件，索引文件由用户根据需要自行创建。

2. 易数据库的记录和字段

易数据库是关系型数据库。关系数据库，是建立在关系模型基础上的数据库，以行和列的形式存储数据，这一系列的行和列被称为表，这张表就是易数据库。

易数据库表每一行称为一条“记录”。记录一个成员各字段的信息。每一个记录包含这行中的所有字段信息，一般记录在数据库中并没有专门的记录名，常常用它所在的行数表示这是第几个记录。

易数据库表每一列称为一个“字段”。字段在数据库中的属性称为“字段信息”，由名称、类型、最大文本三方面属性构成：

(1) 名称：文本型，长度在 16 个字符以内。

(2) 类型：整数型，可以是以下常量之一：#字节型；#短整型；#整数型；#长整数型；#小数型；双精度小数型；#逻辑型；#日期时间型；#文本型；#字节集型；#备注型。

(3) 最大文本长度：整数型，初始值为 20 个字符，仅当字段为文本型时才有效，其值必须在 1 到 512 之间。

下面是一个易数据库“员工工资表”，该表有“张三”、“李四”、“王五”三条记录，每条记录又有“姓名”、“工资”、“扣除”、“实发”四个字段。

姓名	工资	扣除	实发
张三	1500	500	1000
李四	2000	100	1900
王五	1800	200	1600

员工工资表

学习中文编程——易语言（二十）：易语言数据库的建立

易语言数据库操作有两种方法，一种是使用易语言菜单，另一种是在程序中使用代码。我们重点学习重点是使用易语言菜单编辑易数据库，使用代码编辑易数据库不展开细说。

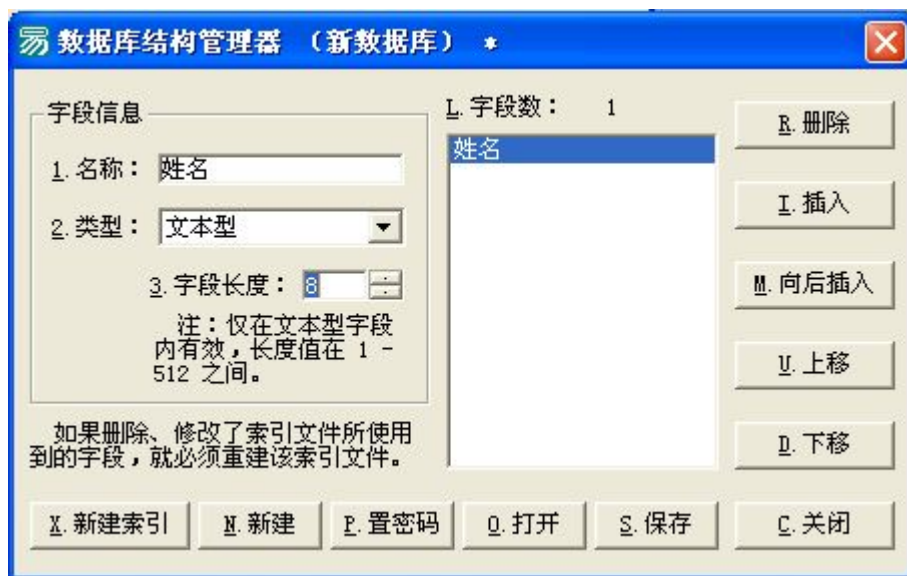
1. 创建易数据库

（1）使用菜单创建易数据库

使用菜单创建易数据库也称手工创建易数据库，选择点击菜单“数据库”——“结构编辑器”，会弹出数据库结构管理器界面，然后插入字段信息，定义名称、类型、字段长度。插入若干需要的字段，并定义名称、类型、字段长度，保存并为数据库命名。

利用数据库结构管理器，可以对字段进行修改：在“字段数”列表框中选择需要修改的字段，在“字段信息”分组框中修改。还可以上、下移动调整字段位置。

数据库结构管理器还可以新建索引、新建数据库表、给数据库设置密码、打开已有数据库、保存当前数据库、关闭数据库结构管理器等功能。



数据库结构管理器界面

(2) 使用代码创建易数据库

一般将数据库中所使用的名称及字段名设置为常量，有利创建和维护。以前面“员工工资表”为例，建立一个常量数据表，如下：

常量名称	常量值	公开	备注
员工工资表	“员工工资表.edb”		
姓名	“姓名”		
工资	“工资”		
扣除	“扣除”		
实发	“实发”		

数据库名称和字段名常量表

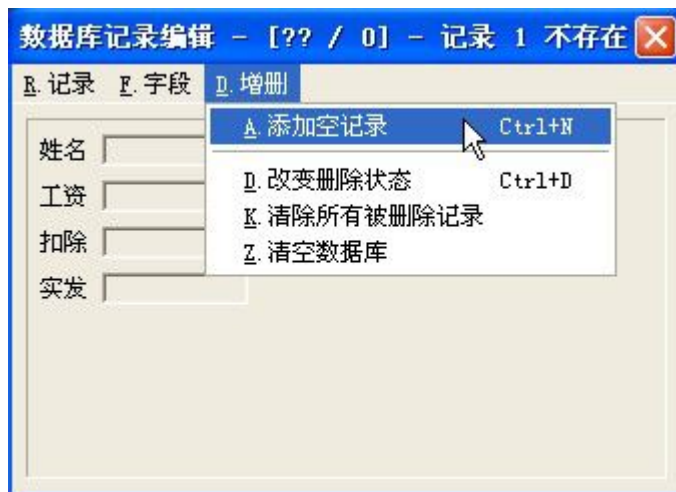
使用代码创建易数据库，需要使用“创建（）”命令，该命令的原型是：“<逻辑型>创建（文本型欲创建数据库的文件名，字段信息数组字段信息表）”。创建（）命令的返回值是逻辑型，表明是否执行成功；第一个参数是要创建数据库的文件名，如：“员工工资表.edb”，也可加上要创建的路径；第二个参数是一个字段信息数组。

2. 为易数据库添加记录

（1）使用菜单添加记录

依次点击菜单“数据库”——“记录编辑器”，打开“员工工资表”数据库，弹出“数据库记录编辑”界面，选择菜单“增删”——“添加空记录”，填写字段信息。再点击“添加空记录”子菜单，每点击一次，就会添加一条空记录。

还可用“记录”菜单下子菜单，实现“到首记录”、“到尾记录”、“上一记录”、“下一记录”、“上一页”、“下一页”、“到指定记录”操作；用“字段”菜单下子菜单，实现“到首字段”、“到尾字段”、“上一字段”、“下一字段”、“上一行”“下一行”操作；用“删除”菜单下子菜单，实现“改变删除状态”、“清除所有被删除记录”、“清空数据库”操作。



数据库记录编辑界面

查看添加好的记录，依次点击易语言菜单“工具”——“报表编辑器”，弹出“易之表”界面，点击“易之表”菜单“文件”——“打开数据库”，找到并打开“员工工资表.edb”文件，如下图：

易之表				
F. 文件 E. 编辑 C. 单元格 A. 关于				
	姓名	工资	扣除	实发
1	张三	1500	500	1000
2	李四	2000	100	1900
3	王五	1800	200	1600

员工工资表

(2) 使用代码添加记录

使用代码添加和修改记录，主要有以下几个命令：

“加空记录（）”，可以在当前数据库记录的尾部添加一条空记录。

“加记录（）”，在添加空记录时，可以同时提供欲添加的数据。

“添加（）”，可以将其它数据库内的记录添加到本数据库。

“写（）”，可以将数据写入当前数据库中的当前记录处的指定字段内。

“写字段（）”，可以将数据写入当前数据库内的记录字段。

“修改（）”，可以一次性修改当前记录的多个字段。

3. 易数据库密码设置

（1）使用菜单设置密码

选择菜单“数据库”——“设置数据库密码”，弹出“设置数据库密码”界面，两次输入密码，点击“确认”按钮。



设置数据库密码

（2）使用代码设置密码

使用代码设置密码，需要使用“设数据库密码（）”命令，命

令原型是：“<逻辑型>设数据库密码（[文本型新密码文本]）”。

易数据库设置密码，可以提高安全性，避免信息外泄。设置了密码的数据库打开时，要输入正确的密码。

学习中文编程——易语言（二十一）：易语言数据库相关组件

为了方便将数据库中的数据显示到程序界面，易语言提出了“数据应用框架”的概念。

“数据应用框架”最大的特点就是把数据、数据操作、数据显示分为三个不同的层次，每个层次由各自的组件完成相对独立的工作，各层次之间的联系，由易语言在内部实现。这三个层次由低到高分别是：数据提供者、数据源、数据处理者。其中数据提供者用于存储、提供数据；数据源用于操作数据库；数据处理者用于显示数据。

1. 易数据库相关组件简介

（1）通用提供者

“通用提供者”可作为数据源提供对数据的存取操作，使用内存作为数据的存储仓库，全面支持所有数据操作接口。

通用提供者的重要属性有“初始行数”和“初始列数”，用于指定初始数据的行数和列数。

通用提供者功能强大，可对数据进行各种操作，但没有直接数据来源。

（2）数据库提供者

“数据库提供者”可作为数据源提供对数据库的存取操作，使用数据库作为数据的存储仓库，不支持以下 19 种接口：置行高、置类别、置文本色、置背景色、置字体名、置字体尺寸、置字体属性、置边距、置文本输入格式、置对齐方式、置密码方式、合并、分解、加线条、删线条、初始尺寸时同时改变列数、在中间插入行、插入列、删除

列。如果对数据进行以上操作，应将数据通过数据源导入到通用提供者中。

数据库提供者的重要属性有“数据库文件名”、“字节集字段处理”、“数据库密码”。

数据库提供者功能较少，只能完成数据的基本操作，但可以直接连接到数据库。

（3）数据源

数据源用于配合各种数据提供者窗口组件提供数据，组件内置了一系列对数据库的操作，简化窗口组件与数据库的关联。

数据源重要属性“数据提供者”，指定本数据源所基于的“数据提供者”组件名，如果此时窗体上已经放置多个“数据提供者”组件，则从下拉列表框中选择其中一个。

数据源组件的命令非常多，大约有 70 多条。

2. 易数据库相关组件关联方法

数据提供者、数据源、数据处理者三者之间必须事先设置关联，才能共同完成对数据的处理。

设置方法：首先一次性添加数据库提供者、数据源、数据处理者三种组件，然后设置数据提供者的“数据库文件名”属性，连接数据库文件，再将数据源的“数据提供者”属性设置为某个数据提供者组件，最后将数据处理者组件的“数据源”属性设置为某个数据源组件。

例如：新建一个易语言程序，添加数据提供者、数据源、表格（数据处理者）三个组件，选择“数据提供者 1”的数据库文件名属性，

找到并选取“员工工资表.edb”易数据库文件；设置“数据源 1”的数据提供者属性为 “数据提供者 1” 组件；设置“表格 1”的数据源为 “数据源 1” 组件。易数据库相关组件关联完毕。



数据提供者 1 与员工工资表关联



数据源 1 与数据提供者 1 关联



表格 1 与数据源 1 关联

	姓名	工资	扣除	实发
1	张三	1500	500	1000
2	李四	2000	100	1900
3	王五	1800	200	1600

数据库相关组件关联后的窗口

如果是通用提供者、数据源、数据处理者三种组件关联，一次性添加通用提供者、数据源、数据处理者三种组件，以“员工工资表”为例，选择“通用提供者 1”，设置初始行数为“1”，设置初始列数为“4”；设置“数据源 1”的数据提供者属性为“通用提供者 1”；设置“表格 1”的数据源属性为“数据源 1”。

3. 易数据库程序的界面设计

我们编写一个易语言数据库程序，探讨易数据库程序的界面设计。

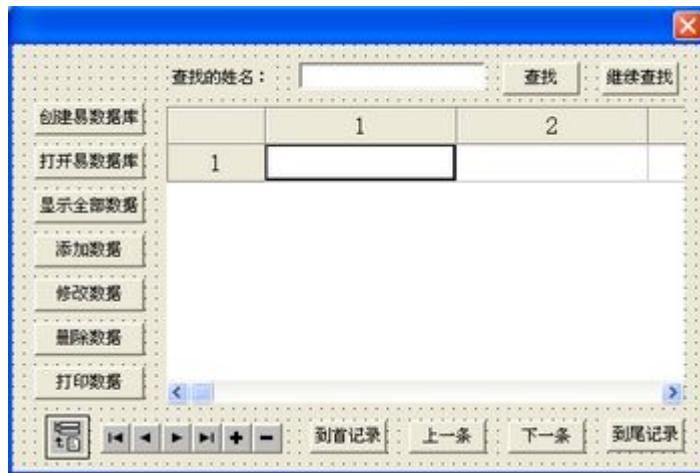
(1) 运行易语言程序，在窗口添加一个“通用提供者”组件，将初始行数和初始列数分别定义为“1”和“4”；添加一个“数据源”组件，将其数据提供者属性设置为刚添加的“通用提供者 1”组件；添加一个“表格”组件，将数据源属性设置为刚添加的“数据源 1”组件。

(2) 左侧添加 7 个按钮，标题分别设置为：“创建易数据库”、“打开易数据库”、“显示全部数据”、“添加数据”、“修改数据”、“删除数据”、“打印数据”。

(3) 底部添加 4 个按钮，标题分别设置为：“到首记录”、“上一条”、“下一条”、“到尾记录”。

(4) 顶部添加 1 个标签，标题设置为：“查找的姓名：”；添加 1 个编辑框，用来输入查找的姓名；添加 2 个按钮，标题分别设置为：“查找”、“继续查找”。

易数据库程序的界面设计完毕，可以为这些组件编写代码了。



易数据库程序的界面

学习中文编程——易语言（二十二）：易语言数据库的操作命令

易数据库程序界面设计好后，通过易语言数据库的操作命令，实现其功能。

1. 易数据库的打开与关闭

（1）打开（）

数据库使用前必须打开，不打开则无法使用，已经打开的数据库不能重复打开。

打开（）命令的原型是：“<逻辑型>打开（文本型数据库文件名，[文本型在程序中使用的别名]，[逻辑型 是否只读]，[整数型 共享方式]，[文本型 保留参数 1]，[文本型数据库密码]，[文本型数组/非数组索引文件表]，…）”。

此命令用于打开指定的数据库文件。成功返回真，并将“当前数据库”状态指向此数据库；失败返回假。

（2）置当前库（）

系统内部有一个“当前数据库”状态值，用来指向被打开的数据库。

置当前库（）命令原型是：“<逻辑型>置当前库（文本型数据库别名或名称）”。

此命令可以改变系统中“当前数据库”状态值的指向。

（3）关闭（）

在程序结束或不再使用某个数据库时，要关闭数据库。

关闭（）命令的原型是：“<无返回值>关闭（[文本型数据库

别名或名称]) ” 。

此命令用来关闭已经打开的指定数据库, 如果一次性关闭已经打开的所有数据库, 可以使用 “全部关闭 () ” 命令。

2. 易数据库指针跳转命令

当打开一个数据库, 有一个 “当前记录指针” 状态值, 提供位置指示, 跳转就是移动指针位置。

(1) 首记录前 ()

当前记录指针移动到数据库第一条记录前面。

(2) 尾记录后 ()

前记录指针移动到数据库最后一条记录后面。

(3) 到首记录 ()

当前记录指针移动到数据库首记录上。

(4) 到尾记录 ()

当前记录指针移动到数据库尾记录上。

(5) 跳过 ()

移动当前记录指针, 可以遍历数据库中所有记录。

3. 记录读取命令

用于数据库记录的读取。

(1) 读 ()

读取并返回当前数据库中当前记录处指定字段的数据内容。

(2) 读字段 ()

读取非当前数据库内的记录字段。

4. 数据源常用命令

(1) 取行数 ()

返回数据源中现行数据行数。

(2) 删除行 ()

在数据源中指定位置处删除数据行。

(3) 插入行 ()

在数据源中指定位置处插入新数据行。

(4) 置文本 ()

设置数据源中指定单元格的文本内容。

(5) 置对齐方式 ()

在数据源中指定单元格所使用的对齐方式。

(6) 加线条 ()

为数据源中指定范围内单元格添加线条。

另外，还有打印相关命令。

5. 记录添加与修改命令

(1) 加空记录 ()

在当前数据库的尾部添加一条新的空记录。添加空记录后，可以用写 () 或写字段 () 命令对新加的空记录进行写操作。

(2) 加记录 ()

在当前数据库的尾部添加一条新的记录，同时提供欲添加的数据。

(3) 添加 ()

将其它数据库内的记录添加到本数据库。

(4) 写 ()

将数据写入到当前数据库中当前记录处的指定字段内。

(5) 写字段 ()

将数据写入到非当前数据库内的记录字段。

(6) 修改 ()

一次性修改当前记录的多个记录字段。

6. 记录的删除命令

(1) 删除 ()

将当前数据库中的当前记录或指定范围内的记录打删除标记，当前记录指针不变。

(2) 彻底删除 ()

将当前数据库中所有被打删除标记的记录从数据库彻底清除。

(3) 清空 ()

彻底删除当前数据库中的所有记录。

(4) 是否已删除 ()

判断本条记录是否已经打删除标记。

(5) 恢复删除 ()

去掉当前数据库中当前记录或指定范围内记录的删除标记，当前记录指针不变。

7. 复制记录与复制结构命令

(1) 复制记录 ()

复制当前数据库的记录到另一个数据库。

(2) 复制结构 ()

复制当前数据库的结构到另一个数据库，所谓“结构”是指数据库字段的定义和顺序。

8. 数值统计类命令

(1) 排序 ()

根据指定字段排序复制当前数据库的记录到另一个数据库。

(2) 计算排序 ()

根据指定数值表达式的计算值排序复制当前数据库的记录到另一个数据库。

(3) 求和 ()

根据当前数据库计算并返回某数值型表达式的和，命令执行后当前记录指针不变。

(4) 取平均值 ()

根据当前数据库计算并返回某数值型表达式的平均值，命令执行后当前记录指针不变。

(5) 取最大值 ()

根据当前数据库计算并返回某数值型表达式的最大值，当前记录指针移动到具有最大值的记录。

(5) 取最小值 ()

根据当前数据库计算并返回某数值型表达式的最上值，当前记录指针移动到具有最小值的记录。

9. 记录的查找

查找（）

命令的原型是：“<逻辑型>查找（条件语句型 查找条件）”。

此命令从当前数据库中当前记录位置处开始寻找符合给定条件的记录。如果成功找到返回真，当前记录指针移至所找到的记录；如果未找到则返回假，当前记录指针不变。

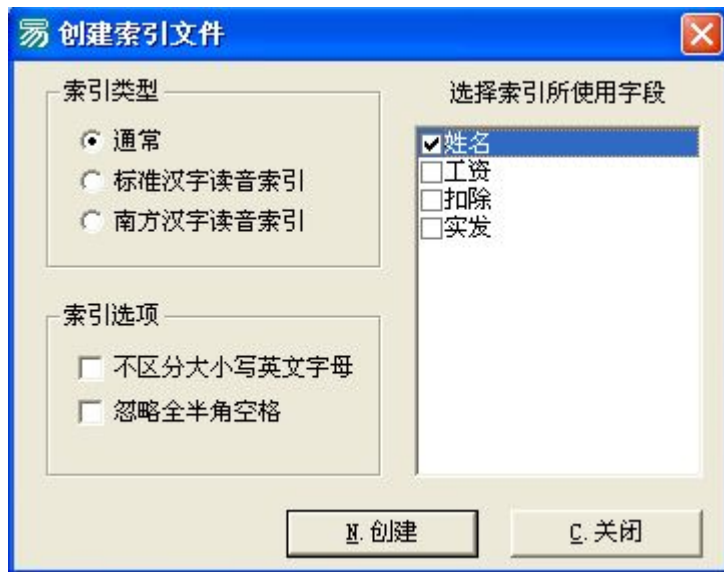
10. 索引的创建与使用

索引是一个单独的物理的数据库结构，是某个表中一系列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。表的存储由两部分组成，一部分用来存放数据页面，另一部分存放索引页面。索引页面相对数据页面小得多，建立合理的索引，就能加速数据的检索过程。

（1）索引的创建

索引的创建方法有菜单创建和代码创建。

菜单创建的方法是：选择菜单“数据库”——“结构编辑器”，点击“新建索引”按钮，在弹出的“创建索引文件”界面选择作为索引的字段，点击“创建”按钮，一个索引就创建好了。



创建索引文件界面

在程序中使用代码创建索引，需要用到“新建索引（）”命令。

命令的原型是：“<逻辑型>新建索引（文本型欲创建索引文件的名称，[整数型索引类型]，[整数型 索引选项]，[整数型 索引块尺寸]，文本型数组/非数组被索引字段的名称，…）”。

（2）索引查找（）

索引查找（）命令的原型是：“<逻辑型>索引查找（通用型欲搜寻值，…）”。|

此命令要当前数据库中使用其当前索引来快速寻找某一记录。寻找从当前记录位置处开始，命令所提供的欲搜寻值参数的数目必须与被索引字段的数目一致。如果成功找到返回真，并且当前记录指针移至所找到的记录；如果未找到则返回假，当前记录指针位置保持不变。

索引查找支持近似音索引文件，如“#南方读音索引”，这样，就可以进行近似音查询了。

学习中文编程——易语言（二十三）：外部数据库应用

语言虽然有自己的数据库，但易数据库只是以简单的、类似表格的形式组织数据信息，是一个小型数据库管理系统。有时编程中必须使用其它数据库。易语言支持以 ODBC 方式和 ADO 方式操作外部数据库。

1. ODBC 与 ADO

（1）ODBC 简介

ODBC（Open Database Connectivity，开放式数据库互联），是目前国际上通用的数据库访问标准，是微软公司开放服务结构中的组成部分，提供了一个对数据库访问的标准 API（应用程序编程接口），这些 API 利用 SQL 语言完成大部分任务。

ODBC 的最大优点是能以统一的方式处理所有的数据库。

ODBC 不能直接访问数据库，必须通过驱动程序管理器与数据库交换信息。客户应用程序连接 ODBC 数据源，ODBC 数据源通过 ODBC 驱动管理器与特定的 ODBG 驱动程序联系起来，然后通过此 ODBC 驱动程序访问本地或远程数据库。

ODBC 数据源又叫 DSN，它把客户应用程序、数据库、用户名、密码等信息组合起来，供客户端程序使用。

在 WindowsXP 系统，可通过“控制面板”——“管理工具”——“数据源（ODBC）”配置 ODBC 数据源。

（2）ADO 简介

ADO（ActiveX Data Objects，数据对象），是微软公司的一个用于存取数据源的 COM 组件，用以实现访问关系或非关系数据库中的

数据。

ADO 是面向对象的编程接口，是编程语言和统一数据访问方式 OLE DB 的一个中间层。可以为任何数据源提供高性能的访问，并允许开发人员编写访问数据的代码，而不用关心数据库是如何实现的，只需要关心数据库的连接。

ADO 是对当前微软所支持的数据库进行操作的最有效和最简单直接的命令，是功能强大的数据访问编程模式，从而使大部分数据源可编程的属性得以扩展。

ADO 方式操作数据库比 ODBC 方式操作数据库效率更高。

2. 外部数据库组件

(1) “外部数据提供者”组件和“外部数据库”组件

易语言核心支持库中的“外部数据提供者”组件和“外部数据库”组件，封装了以 ODBC 方式访问数据库的标准接口，可以直接将外部数据库绑定到 ODBC 数据源，并对外部数据库进行操作。

(2) “数据库连接”组件和“记录集”组件

数据库操作支持库中的“数据库连接”组件和“记录集”组件，封装了以 ADO 方式访问数据库的标准接口，使用 ADO 方式对外部数据库进行操作。

(3) MySQL 支持库

MySQL 支持库可以对 MySQL 数据库系统进行操作。MySQL 数据库是一个大型关系型数据库管理系统，性能高，运行速度快，操作简单，并且支持 Linux 操作系统，因此成为跨平台的数据库首选工具。

(4) Sqlite 数据库支持库

Sqlite 数据库支持库可能对 Sqlite 数据库进行操作。

Sqlite 数据库是一个小型关系型数据库，跨平台，支持 SQL 语句、事务、触发器、视图，速度快，小巧且不依赖任何驱动程序。Sqlite 数据库是本地数据库，不支持网络传输。

3. SQL Server 数据库和 Access 数据库

可供易语言使用的数据库很多，大型的数据库有：SQL Server、Oracle、Sybase、DB2、MySQL 等，小型的数据库有：Access、FoxPro、Sqlite 等。下面简单了解兼容性好、使用频率高的 SQL Server 数据库和 Access 数据库。

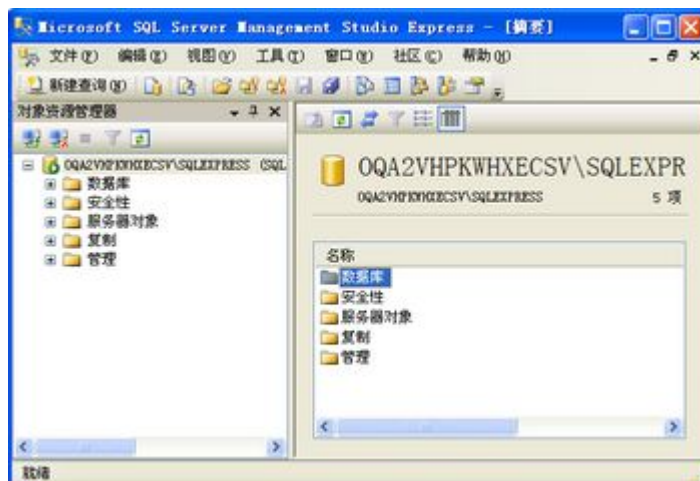
(1) SQL Server 数据库

SQL Server 是一个大型关系数据库管理系统，最初是由 Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发的，于 1988 年推出了第一个 OS/2 版本。后来 Microsoft 独自开发了 SQL Server 几个版本，如：SQL Server 2000，SQL Server 2005，SQL Server 2008，还有免费版本 SQL Server Express。

SQL Server 是一个全面的数据库平台，使用集成的商业智能 (BI) 工具提供了企业级的数据管理，数据库引擎为关系型数据和结构化数据提供了更安全可靠存储功能，可以构建和管理用于业务的高可用和高性能的数据应用程序。

SQL Server 具有使用方便、可伸缩性好、与相关软件集成程

度高等优点，结合了分析、报表、集成和通知功能，有着灵活的实现方式，允许选择最适合的解决方案。



SQL Server Express 2005

(2) Access 数据库

Access 是一个小型关系数据库，是 Microsoft 的 Office 软件包中的一个组件。现在流行的版本有：Access 2000, Access 2003, Access 2007 等。

Access 数据库具有界面友好、易学易用、开发简单、接口灵活的特点。

Access 数据库由一系列表组成，表又由一系列行和列组成，每一行是一个记录，每一列是一个字段，每个字段有一个字段名。

Access 2003 数据库由七种对象组成，它们是表、查询、窗体、报表、宏、页和模块。



Access 2003

学习中文编程——易语言（二十四）：常用 SQL 语句

SQL (Structured Query Language, 结构化查询语言) 语言, 是一种数据库查询和程序设计语言, 用于存取数据以及查询、更新、和管理关系数据库系统。

SQL 语言具有功能丰富、使用方便灵活、语言简洁等优点。SQL 语句以记录集合作为操作对象, 以接受集合作为输入, 以返回集合作为输出, 允许一条 SQL 语句的输出作为另一条 SQL 语句的输入, 所以 SQL 语句可以嵌套, 在多数情况下, 其它语言需要一大段程序实现的功能, 只需要一个 SQL 语句就可达到目的。

SQL 语言包括三种主要程序设计语言类别: 数据定义语言 (DDL)、数据操作语言 (DML)、数据控制语言 (DCL)。SQL 语言的每个功能都由若干条指令组成, 每条指令表示对数据库的一种操作。

下面例举一些常用 SQL 语句和几个 SQL 语句使用方法:

1. 常用 SQL 语句

(1) DDL (Data Definition Language, 数据定义语言): 用于定义数据结构。

- CREATE TABLE: 创建数据库表格;
- ALTER TABLE: 修改数据库表格;
- DROP TABLE: 删除数据库表格;
- CREATE INDEX: 创建索引;
- DROP INDEX: 删除索引。

(2) DML (Data Manipulation Language, 数据操纵语言): 用

于检索或修改数据。

- SELECT: 检索数据;
- INSERT: 增加数据到数据库;
- UPDATE: 从数据库中修改现在数据;
- DELETE: 从数据库中删除数据。

(3) DCL (Data Control Language, 数据库控制语言): 用于定义数据库用户权限。

- GRANT: 赋予一个用户、一个组或所有用户访问权限;
- REVOKE: 废除一个用户、一个组或所有用户访问权限。

2. 定义表的结构

(1) 创建表

SQL 语言中的 CREATE TABLE 语句用来建立新的数据表。以“员工工资表”为例, 格式如下:

CREATE TABLE 员工工资表 (姓名 text(8), 工资 int, 扣除 int, 实发 int)。

易语言中, 使用“外部数据库”的代码:

外部数据库. 执行 (“CREATE TABLE 员工工资表 (姓名 text(8), 工资 int, 扣除 int, 实发 int)”,)

其中: 姓名、工资、扣除、实发是字段信息名称, text、int 为文本型、整数型数据类型, (8) 为文本型数据的字段长度。

(2) 删除表

SQL 语言中的 DROP TABLE 语句用来删除数据表。格式如下:

DROP TABLE 员工工资表

易语言中，使用“外部数据库”的代码：

外部数据库.执行（"DROP TABLE 员工工资表",）

3. 数据检索

SQL 语言中的 SELECT 语句用来检索查询数据，通过变化查询条件和查询方式，可以完成多种查询任务。格式如下：

SELECT*FROM 员工工资表 WHERE 姓名='张三'

易语言中，使用“外部数据库”的代码：

外部数据库.查询（" SELECT*FROM 员工工资表 WHERE 姓名='张三'"）

其中：SELECT 是目标字段，FROM 是基本表，WHERE 是条件表达式。

4. 设置访问权限

SQL 语言中的 GRANT 语句用来设置用户对数据库的访问权限。GRANT 是权限，on 是数据库对象，to 是用户名；权限：all 表示所有权限,select 表示单独查询，update 表示更新；用户名：pubic 代表所有用户，单独用户。格式如下：

GRANT select on 员工工资表 to pubic

易语言中，使用“外部数据库”的代码：

外部数据库.执行（"GRANT select on 员工工资表 to pubic",）

SQL 语句在 GRANT 后面加入所需要的权限，然后再依次设定数

数据库对象和用户等。以上设置允许所有用户查询员工工资表的内容。



学习常用 SQL 语句

学习中文编程——易语言（二十五）：外部数据库相关组件. 1

易语言中 ODBC 连接的外部数据库组件包括“外部数据提供者”和“外部数据库”，以 Access 数据库为例介绍以 ODBC 方式操作数据库。

1. 外部数据提供者组件

外部数据提供者有两个重要属性：“连接文本”和“查询 SQL”。

连接文本属性：文本型，用于设置外部数据库的 ODBC 连接文本。

查询 SQL 属性：文本型，用作指定数据库中的数据表名或用作查询记录集的 SELECT 类 SQL 语句。

（1）新建一个易程序，在组件箱中选择“外部数据提供者”组件，将其放到“_启动窗口”上。在属性面板中选中“连接文本”属性，点击出现的小按钮。

（2）在弹出的“选择数据源”对话框中选择“文件数据源”，点击“新建”按钮。

③ 在弹出“创建新数据源”对话框，选中“Driver do Microsoft Access (*.mdb)”，然后单击“下一步”按钮。

（4）单击“浏览”按钮，弹出“另存为”对话框，选择保存位置，输入“员工工资.dsn”，点击“保存”按钮，返回“创建新数据源”对话框。

（5）点击“下一步”按钮，出现“创建新数据源”确认对话框，确认无误后，点击“完成”按钮，创建数据源工作结束。

（6）在弹出的“ODBC do Microsoft Access 安装”对话框中，

点击“选择”按钮，在：“选择数据库”对话框中选中 Access 数据库“员工工资.mdb”，点出“确定”按钮 这时“ODBC do Microsoft Access 安装”对话框中间多了 Access 数据库路径名。

(7) 点出“确定”按钮，又出现“选择数据源”对话框，在“搜索范围”下的列表框中选择“员工工资”数据源文件，单击“确定”按钮，完成了数据源的连接。

(8) 进入“ODBC do Microsoft Access 安装”对话框，单击“确定”按钮，完成全部操作。

当下次再连接同一数据源时，直接在“选择数据源”对话框中选择“员工工资”就行了。

2. 外部数据库组件

外部数据库组件有“名称”、“备注”、“左边”、“顶边”、“宽度”、“高度”、“标记”等基本属性。

外部数据库组件有 19 条命令，分别是：

(1) 打开（）命令

命令的原型是：<逻辑型>对象. 打开（[文本型 ODBC 数据源连接文本]，[逻辑型 是否只读]，[逻辑型 不显示 ODBC 连接对话框]）。

(2) 关闭（）命令

命令的原型是：<无返回值>对象. 关闭（）。

(3) 取连接文本（）命令

命令的原型是：<文本型>对象. 取连接文本（）。

(4) 启动事务（）命令

命令的原型是：〈逻辑型〉对象. 启动事务（）。

(5) 回滚事务（）命令

命令的原型是：〈无返回值〉对象. 回滚事务（）。

(6) 提交事务（）命令

命令的原型是：〈无返回值〉对象. 提交事务（）。

(7) 查询（）命令

命令的原型是：〈整数型〉对象. 查询(文本型查询类 SQL 语句)。

(8) 重新查询（）命令

命令的原型是：〈逻辑型〉对象. 重新查询（整数型 记录集句柄）。

(9) 关闭记录集（）命令

命令的原型是：〈无返回值〉对象. 关闭记录集（整数型 记录集句柄）。

(10) 首记录前（）命令

命令的原型是：〈逻辑型〉对象. 首记录前（整数型 记录集句柄）。

(11) 尾记录后（）命令

命令的原型是：〈逻辑型〉对象. 尾记录后（整数型 记录集句柄）。

(12) 到首记录（）命令

命令的原型是：〈无返回值〉对象. 到首记录（整数型 记录集句柄）。

(13) 到尾记录 () 命令

命令的原型是: <无返回值>对象. 到尾记录 (整数型记录集句柄)。

(14) 到前一记录 () 命令

命令的原型是: <无返回值>对象. 到前一记录 (整数型记录集句柄)。

(15) 到后一记录 () 命令

命令的原型是: <无返回值>对象. 到后一记录 (整数型记录集句柄)。

(16) 读 () 命令

命令的原型是: <通用型>对象. 读 (整数型 记录集句柄, 通用型字段名称或位置)。

(17) 执行 () 命令

命令的原型是: <逻辑型>对象. 执行 (文本型 非查询类 SQL 语句, [通用型文本或字节集], ...)。

(18) 打开 MDB 数据库 () 命令

命令的原型是: <逻辑型>对象. 打开 MDB 数据库 (文本型 MDB 数据库文件名, [文本型 用户名], [文本型 密码], [逻辑型 是否只读], [逻辑型 不显示 ODBC 连接对话框])。

(19) 打开 SQL 数据库 () 命令

命令的原型是: <逻辑型>对象. 打开 SQL 数据库 (文本型服务器名称, [文本型 用户名], [文本型 密码], [文本型 数据库名称], [逻辑型 是否只读], [逻辑型 不显示 ODBC 连接对话框])。

辑型 是否只读], [逻辑型不显示 ODBC 连接对话框])。

3. 应用实例

新建一个易语言程序, 加入组件: 通用提供者、数据源、表格、外部数据提供者、外部数据库和“简单事物”、“简单查询”、“添加”、“删除”、“刷新”五个按钮。编写代码, 完成数据库的基本功能。

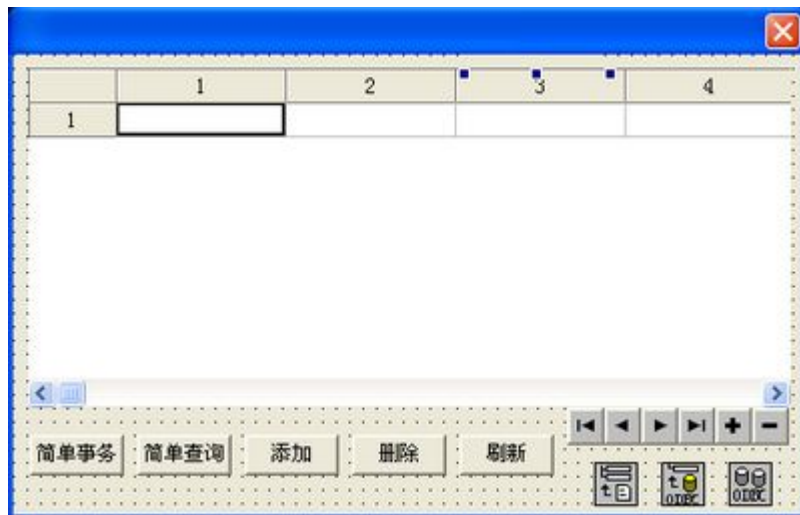
(1) 应用“外部数据库. 打开 MDB 数据库 (取运行目录 () + “\ 员工工资.mdb” …)”命令打开 Access 数据库。

(2) 应用“外部数据库. 执行 (“delete from 员工工资表 where 编号=1”)”命令删除编号是 1 的记录。

(3) 应用“外部数据库. 执行 (“update 员工工资表 set 工资=1800 where 姓名= '张三' ,)”命令修改姓名是张三的记录的工资为 1800。

(4) 应用“记录集句柄=外部数据库. 查询 (“select*from 员工工资表 order by 编号”)”命令得到包含数据表中全部记录 of 记录集句柄。

还可以应用“记录集句柄=外部数据库. 查询 (“select*from 员工工资表 where 实发>¥1500.00 and 姓名='王五' ”)”命令得到表中都符合查询条件的记录。



应用外部数据库组件的界面设计

学习中文编程——易语言（二十六）：外部数据库相关组件. 2

易语言中 ADO 连接的外部数据库组件包括“数据库连接”和“记录集”，以 Access 数据库为例介绍以 ODBC 方式操作数据库。

1. 数据提连接组件

(1) 数据提连接组件的属性

“最后错误”属性：只读，文本型，获取最后错误文本。

“是否已连接”属性：只读，逻辑型，判断数据库是否连接。

“对象提供者”属性：只读，文本型，对象提供者名称。

“引擎版本”属性：只读，文本型，数据引擎版本。

数据库连接组件的这些属性都是只读属性，只能读取它的内容而不能更改。

(2) 数据提连接组件的命令

● 连接（）命令

命令的原型是：<逻辑型>对象. 连接（文本型 连接文本）。

● 连接 Access（）命令

命令的原型是：<逻辑型>对象. 连接 Access（文本型 文件名，文本型密码）。

● 连接 SQL Servet（）命令

命令的原型是：<逻辑型>对象. 连接 SQL Servet（文本型 服务器名，文本型 数据库名，文本型 用户名，文本型 密码）。

● 关闭（）命令

命令的原型是：<逻辑型>对象. 关闭（）。

●执行 SQL () 命令

命令的原型是：<逻辑型>对象. 执行 SQL (文本型 SQL 语句)。

●取得权限 () 命令

命令的原型是：<整数型>对象. 取得权限 ()。

取得访问及相关权限，并返回一个整数代表用户取得当前权限。该整数为以下常数之一或之和：0 (#未知权限)、1 (#只读权限)、2 (#只写权限)、3 (#读写权限)、4 (#独占读权限)、8 (#独占写权限)、12 (#独占权限)、16 (#共享权限)。

●设置权限 ()

命令的原型是：<逻辑型>对象. 设置权限 (整数型要设置的模式)。

参数要设置的模式为以下常数之一或之和：0 (#未知权限)、1 (#只读权限)、2 (#只写权限)、3 (#读写权限)、4 (#独占读权限)、8 (#独占写权限)、12 (#独占权限)、16 (#共享权限)。

●设超时时间 () 命令

命令的原型是：<整数型>对象. 设超时时间 (整数型要设置的时间)。

●取超时时间 () 命令

命令的原型是：<整数型>对象. 取超时时间 ()。

●开始事务 () 命令

命令的原型是：<整数型>对象. 开始事务 ()。

●保存事务 () 命令

命令的原型是：〈整数型〉对象. 保存事务（）。

●回滚事务（）命令

命令的原型是：〈整数型〉对象. 回滚事务（）。

2. 记录集组件

（1）记录集组件的属性

“是否已打开”属性：只读，逻辑型，判断记录集是否打开。

“操作状态”属性：只读，整数型，取得当前操作状态。

“记录数量”属性：只读，整数型，取得记录的数量。如果为-1，表示 ADO 当前无法判断记录的数量，或者数据提供者不支持这个属性。

“字段数量”属性：只读，整数型，取得字段的数量。

“首记录前”属性：只读，逻辑型，判断当前记录指针是否已在首记录前。

“尾记录后”属性：只读，逻辑型，判断当前记录指针是否已在尾记录后。

“当前位置”属性：只读，整数型，当前记录在记录集中的位置，0 为第一条记录。如果为-2 表示位置未知，如果为-3 表示处于文件开头，如果为-4 表示文件结尾。

（2）记录集组件的命令

●置连接（）命令

命令的原型是：〈逻辑型〉对象. 置连接（数据库连接 连接）。

●取连接（）命令

命令的原型是：〈数据库连接〉对象. 取连接（）。

●打开（）命令

命令的原型是：〈逻辑型〉对象. 打开（文本型命令文本，整数型命令类型）。

●打开并排序（）命令

命令的原型是：〈逻辑型〉对象. 打开并排序（文本型表名，文本型排序条件）。

●打开并过滤（）命令

命令的原型是：〈逻辑型〉对象. 打开并过滤（文本型表名，文本型过滤条件）。

●关闭（）命令

命令的原型是：〈无返回值〉对象. 关闭（）。

●添加（）命令

命令的原型是：〈逻辑型〉对象. 添加（）。

●更新（）命令

命令的原型是：〈逻辑型〉对象. 更新（）。

●删除（）命令

命令的原型是：〈逻辑型〉对象. 删除（整数型 删除选项）。

●写文本（）命令

命令的原型是：〈逻辑型〉对象. 写文本（通用型 序号或字段名，文本型文本）。

●写单精度（）命令

命令的原型是：<逻辑型>对象. 写单精度（通用型 序号或字段名，小数型单精度）。

●写整数（）命令

命令的原型是：<逻辑型>对象. 写整数（通用型 序号或字段名，整数型整型）。

●写逻辑值（）命令

命令的原型是：<逻辑型>对象. 写逻辑值（通用型 序号或字段名，逻辑型逻辑）。

●写日期时间（）命令

命令的原型是：<逻辑型>对象. 写日期时间（通用型 序号或字段名，日期时间型日期时间）。

●写双精度（）命令

命令的原型是：<逻辑型>对象. 写双精度（通用型 序号或字段名，双精度小数型双精度小数）。

●写字节集（）命令

命令的原型是：<逻辑型>对象. 写字节集（通用型 序号或字段名，字节集型数据）。

●读文本（）命令

命令的原型是：<逻辑型>对象. 读文本（通用型 序号或字段名，文本型变量文本变量）。

●读双精度（）命令

命令的原型是：<逻辑型>对象. 读双精度（通用型序号或字段

名，双精度小数型变量双精度变量）。

●读单精度（）命令

命令的原型是：〈逻辑型〉对象. 读单精度（通用型 序号或字段名，小数型变量单精度变量）。

●读整数（）命令

命令的原型是：〈逻辑型〉对象. 读整数（通用型 序号或字段名，整数型变量整数变量）。

●读逻辑值（）命令

命令的原型是：〈逻辑型〉对象. 读逻辑值（通用型 序号或字段名，逻辑型变量逻辑变量）。

●读日期时间（）命令

命令的原型是：〈逻辑型〉对象. 读日期时间（通用型序号或字段名，日期时间型变量 日期变量）。

●读字节集（）命令

命令的原型是：〈逻辑型〉对象. 读字节集（通用型 序号或字段名，字节集型变量字节集变量）

●保存到 XML（）命令

命令的原型是：〈逻辑型〉对象. 保存到 XML（文本型 XML 文件名）。

●保存到 ASTG（）命令

命令的原型是：〈逻辑型〉对象. 保存到 ADTG（文本型 ADTG 文件名）。

●到首记录（）命令

命令的原型是：〈逻辑型〉对象. 到首记录（）。

●到尾记录（）命令

命令的原型是：〈逻辑型〉对象. 到尾记录（）。

●到前一条（）命令

命令的原型是：〈逻辑型〉对象. 到前一条（）。

●到下一条（）命令

命令的原型是：〈逻辑型〉对象. 到下一条（）。

●移到（）命令

命令的原型是：〈逻辑型〉对象. 移到（整数型 目标记录号）。

●取字段名（）命令

命令的原型是：〈文本型〉对象. 取字段名（整数型 字段序号）。

●取字段属性（）命令

命令的原型是：〈整数型〉对象. 取字段属性（通用型序号或字段名）。

●取字段定义长度（）命令

命令的原型是：〈整数型〉对象. 取字段定义长度（通用型序号或字段名）。

●取字段实际长度（）命令

命令的原型是：〈整数型〉对象. 取字段实际长度（通用型序号或字段名）。

●取字段类型（）命令

命令的原型是：<整数型>对象. 取字段类型（通用型序号或字段名）。通过序号或字段名取得字段类型，返回以下常量之一：0（#空白型字段）、2（#短整数字段）、3（#整数型字段）、4（#单精度字段）、5（#双精度字段）、6（#货币型字段）、7（#日期型字段）、8（#宽文本字段）、9（#IDispatch 字段）、10（#错误码字段）、11（#逻辑型字段）、12（#变体型字段）、13（#IUnkown 字段）、14（#数值型字段）、16（#有符号字节字段）、17（#字节集型字段）、18（#无符号短整数字段）、19（#无符号整数字段）、20（#超长整数字段）、21（#无符号超长整数字段）、72（#标识型字段）、128（#二进制字段）、129（#字符型字段）、130（#宽文本字段）、131（#数字型字段）、132（#自定义型字段）、133（#数据库日期型字段）、134（#时间型字段）、135（#时间戳字段）、200（#变长文本字段）、201（#超变长文本字段）、202（#变长宽文本字段）、203（#超变长宽文本字段）、204（#变长二进制字段）、205（#超变长二进制字段）。

● 字段类是否空（）命令

命令的原型是：<逻辑型>对象. 字段是否空（通用型序号或字段名）。

● 查找（）命令

命令的原型是：<逻辑型>对象. 查找（文本型 查找条件，整数型查找方向）。

● 查找下一个（）命令

命令的原型是：<逻辑型>对象. 查找下一个（）。

●过滤记录（）命令

命令的原型是：<逻辑型>对象.过滤记录（文本型 过滤条件）。

3. 应用实例

新建一个易语言程序，加入组件：通用提供者、数据源、表格、数据库连接、记录集和“简单查询”、“添加”、“删除”、“刷新”四个按钮。编写代码，完成数据库的基本功能。

（1）应用“数据库.连接 Access（）”命令打开数据库；打开数据库后，应用“记录集.置连接（）”命令连接到数据库连接组件；再用“记录集.打开（）”命令打开表。

（2）先使用“记录集.添加（）”命令添加一条空记录；再向空记录中写入具体内容，根据写入内容的类型，分别应用：“记录集.写文本（）”、“记录集.写整数（）”、“记录集.写逻辑值（）”等命令；最后更新数据库，完成记录添加。

（3）使用“数据库连接.执行 SQL（）”命令直接数据库中插入记录。

（4）使用“数据库连接.执行 SQL（）”命令直接数据库中删除记录。



应用外部数据库组件的界面设计

数据库是存储、管理数据的仓库，编写程序离不开数据库。我们耐心地学习了易语言数据库和可应用的外部数据库后，多了一些编程思路，仿佛对易语言中文编程有了些兴趣。当坚持学到今天，我们开始在浩瀚的编程世界中遨游了。

学习中文编程——易语言（二十七）：API 应用

1. API 简介

API（Application Programming Interface，应用程序编程接口），是软件系统不同组成部分衔接的约定，是软件厂商为二次开发提供的命令，使用 API 可以方便地调用其它程序所提供的功能。

Windows 是一个多任务操作系统，可以看成是一个很大的服务中心，通过调动这个系统的各种服务，帮助各种应用程序实现自己的功能。WIN32 API 命令就是 Windows32 位平台应用程序接口，它的下面是 Windows 操作系统核心，而它的上面是五彩缤纷的应用程序。

当 Windows 操作系统开始占据主导地位的时候，程序员使用的编程工具只有 API 命令。随着软件技术的不断发展，出现了许多可视化编程工具，包括中文编程软件易语言，它们在 WIN32 API 命令基础上，把常用 API 命令组合成一些类库和各种组件，并赋予方便的使用命令，简化和加快了编程过程。

虽然类库和组件使应用程序开发方便了许多，但它们只能提供 Windows 的一般功能操作，对于比较复杂和特殊的功能来说，使用现有的类库和组件可能无法实现，这时必然涉及到直接调用 API 命令。

2. API 的定义

在 Windows 中，API 命令位于位于 DLL（动态链接库）文件中，在易语言中把调用 API 命令称之为“DLL 命令”。

新建一个易语言程序，在工作夹“程序”面板中双击“DLL 命令”选项，然后在编辑区单击鼠标右键，在弹出的右键菜单中单击“新

DLL 命令”，在编辑区出现“DLL 命令定义表”，根据需要，填写表格内容，就完成了 DLL 命令定义。

表格的内容是：（参考下图“定义 API”）

（1）DLL 命令名：默认为“DLL 命令”，可用中文修改，以后在程序代码中使用此名称，就相当于使用了对应的“在 DLL 库中对应命令名”。

（2）返回数值类型：不能为非系统基本数据类型或字节集型。

（3）备注：与当前 DLL 相关的备注信息。

（4）DLL 库文件名：当前 DLL 命令所在动态链接库的文件名，如不指定库文件名，系统默认在 Windows 系统的基本 WIN32 API 库中搜寻指定命令。

（5）在 DLL 库中对应命令：当前 DLL 命令在其所在动态链接库中的名称，如不指定，系统默认等同于 DLL 命令名。

（6）参数名：可自行命名。

（7）类型：所定义的数据类型应该与实际 DLL 库命令一至。不能为复和类型数组、窗口、窗口组件、菜单。

（8）传址：某些 DLL 库命令的参数需要接收数据地址，勾选后设置本属性为真，可以实现参数数据地址的传递。

（9）数组：设置当前 DLL 命令参数是否为数组。

Windows 系统中的 WIN32 API 命令有很多，网络上也有很多 WIN32 API 命令的中文帮助文件，里面列举一常用 WIN32 API 命令的相关资料。

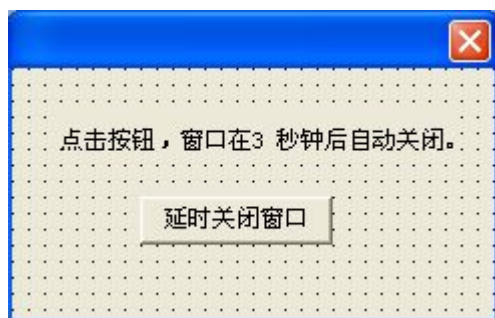
一般 API 命令是以 MSDN（微软技术资源库）为基准的，MSDN 全部是英文资料，就是英语比较好的人也不一定完全看懂。而 VB 对 MSDN 的内容做了简化，并按一定格式做出相关解释，这样能直观看到 API 命令的结构和使用方法，因此在易语言中采用了 VB 声明 API 命令的方式加以对照定义。

3. API 的调用

Windows 系统自带的 API 命令称为系统 API，也叫内部 API；非系统自带、由公司或个人提供的 API 称为非系统 API，也叫外部 API，非系统 API 需要另行安装。

现在调用一个系统 API 命令，来实现延迟三秒钟关闭窗口例程。

（1）新建一个易语言程序，在“_启动窗口”中添加一个标签，名称改为“标签_延时”，标题改为“点击按钮，窗口在 3 秒钟后自动关闭。”；添加一个按钮，名称改为“ 按钮_延时”，标题改为“延时关闭窗口”。



延时关闭窗口

(2)新建 DLL 命令, 定义 API, 本例程用到一个 API 命令“Sleep”:

延迟指定时间, 参数为时间数值, 单位为毫秒。定义 API 如下:

Dll命令名	返回值类型	公开	备 注	
延迟时间	整数值		Sleep	
库文件名：				
kernel32.dll				
在库中对应命令名：				
Sleep				
参数名	类 型	传址	数组	备 注
延迟毫秒数	整数值			dwMilliseconds

定义 API

双击“延时关闭窗口”按钮, 进入代码编辑区, 写如下代码:

子程序名	返回值类型	公开	备 注
_按钮_延时_被单击			

标签_延时.标题 = “还有3秒窗口关闭”

延迟时间 (1000) ‘ 延时第1秒

标签_延时.标题 = “还有2秒窗口关闭”

延迟时间 (1000) ‘ 延时第2秒

标签_延时.标题 = “还有1秒窗口关闭”

延迟时间 (1000) ‘ 延时第3秒

_启动窗口.销毁 ()

延时关闭窗口代码

运行程序, 窗口延迟 3 秒钟后关闭。

学习中文编程——易语言（二十八）：DLL 应用

1. DLL 简介

DLL (Dynamic Link Library, 动态链接库), 是一个包含可由多个程序同时使用的代码和数据。在 Windows 操作系统中, DLL 对于程序执行是非常重要的, 程序在执行时必须链接到 DLL 文件才能运行。而 DLL 文件可以被许多程序共用。但 DLL 不是独立运行的程序, 它是某个程序的一部分, 只能由所属的程序调用。

API 命令都保存在 DLL 文件中。

DLL 动态链接库有以下特征:

- (1) 多个程序共享代码和数据。
- (2) 某些程序过滤系统消息时必须使用 DLL 动态链接库。
- (3) DLL 动态链接库以一种自然的方式, 将一个大的应用程序划分为几个小的模块, 有利于组内成员的分工与合作。
- (4) 为了应用程序国际化, 往往需要使用 DLL 动态链接库。

2. 编写 DLL

很多编程语言可以编写 DLL 文件, 其中包括易语言。用一个例题, 通过调用自定义 DLL 文件中的命令来计算任意三角形的面积:

- (1) 启动易语言主, 新建一个易语言动态链接库程序, 在易语言代码编辑区自动生成: “_启动子程序”和“_临时子程序”两个子程序。



新建 DLL 文件

(2) 新建一个子程序，更改子程序名为“求任意三角形面积”，返回值类型定为“小数型”，勾选“公开”属性，然后创建三个参数：“边长一”“边长二”、“边长三”。

(3) 定义三个局部变量：“半周长，小数型”；“乘积，小数型”；“面积，小数型”。编写如下代码：

子程序名	返回值类型	公开	备 注		
求任意三角形面积	小数型	✓			
参数名	类 型	参考	可空	数组	备 注
边长一	小数型				
边长二	小数型				
边长三	小数型				

变量名	类 型	静态	数组	备 注
半周长	小数型			
乘积	小数型			
面积	小数型			

半周长 = (边长一 + 边长二 + 边长三) ÷ 2

乘积 = 半周长 × (半周长 - 边长一) × (半周长 - 边长二) × (半周长 - 边长三)

面积 = 求平方根 (乘积)

返回 (四舍五入 (面积, 2)) ' 保留2位小数

求任意三角形面积代码

3. 编译 DLL

将编写好的 DLL 动态链接库文件编译输出：

选择菜单“编译”——“编译”，弹出的“请输入欲编译到的可执行文件名：”对话框，修改文件名为：“求任意三角形面积”，单击“保存”按钮。编译后的 DLL 文件名后缀是“dll”。

4. 调用 DLL

用易语言编写的 DLL，不仅可以被易语言调用，也可以被其它编程语言调用。以“求任意三角形面积”为例，易语言调用 DLL 的方法是：

(1) 新建一个易语言程序，在窗体中添加“a 边长”、“b 边长”、“c 边长”、“三角形面积”四个标签，四个编辑框，一个“计算”按钮。

调用 DLL 命令的窗口程序

(2) 再建一个 DLL 命令，定义 DLL 命令。如下图：

Dll命令名	返回值类型	公开	备 注	
求任意三角形面积	小数值			
库文件名：				
求任意三角形面积.dll				
在库中对应命令名：				
求任意三角形面积				
参数名	类 型	传址	数组	备 注
边长一	小数值			
边长二	小数值			
边长三	小数值			

定义 DLL

(3) 双击“计算”按钮，在“_按钮 1_被单击”子程序中，定义三个局部变量：“a 边长”、“b 边长”、“c 边长”，类型都是“小数型”，然后输入如下代码：

窗口程序集名	保 留	保 留	备 注
窗口程序集1			

子程序名	返回值类型	公开	备 注
_按钮1_被单击			

变量名	类 型	静态	数组	备 注
a边长	小数型			
b边长	小数型			
c边长	小数型			

a边长 = 到小数 (编辑框1.内容)

b边长 = 到小数 (编辑框2.内容)

c边长 = 到小数 (编辑框3.内容)

编辑框4.内容 = 到文本 (求任意三角形面积 (a边长, b边长, c边长))

调用 DLL 的代码

把此程序保存在“求任意三角形面积.d11”文件同一目录下。

(4) 运行程序，任意填写“a 边长”、“b 边长”、“c 边长”数据，单击“计算”按钮，通过调用 DLL 中的接口命令，计算任意三角形的面积。

学习中文编程——易语言（二十九）：OCX 组件与类型库

1. OCX 组件简介

OCX 组件是对象类别扩充组件，它的名称来源于它的文件的扩展名 “.ocx”，是目前通用的 Windows 组件格式，组件的本质是微软对象和嵌入（OLE）标准。

一般的开发工具都会提供一些标准控件，但不管标准控件多么强大、多么完善，都不能完全满足用户的需要。所以易语言就提供一个接口，允许用户编写自己的控件插入到易语言中使用。用户编写自己的控件在使用方法上同标准控件是相同的。按照易语言的接口标准编写的控件就是 OCX。

由于使用 OCX 可以节省开发软件的时间，所以现在几乎所有开发工具都允许用户使用 OCX 来增强其功能。后来随着 Internet 的流行，微软又让网页中可以嵌入 OCX，并称其为 ActiveX 控件。

OCX 充分利用面向对象的优点，使得程序效率得到很大提高，从而得到广泛应用。

2. OCX 组件的安装

通过 “XPForm.ocx” 组件，学习易语言中 OCX 组件的安装和汉化：

（1）选择易语言菜单 “工具” —— “类型库或 OCX 组件->支持库”，弹出 “请选择欲包装的 OCX 组件或类型库” 对话框，从列表中选择 “XPForm.ocx” 组件，点击 “下一步” 按钮，在弹出的对话框中勾选 “输出”，再点击 “保存” 按钮，保存为 “npk” 文件，这时组件箱外

部组件中有了“XP 窗口”组件。

(2) 如果系统中没有“XPForm.ocx”组件，可先到易语言官网“资源中心”下载或从书籍《中文编程从入门到精通》附带的光盘中安装，将“XPForm.ocx”文件保存到本地磁盘，点击“注册组件”按钮，选中“XPForm.ocx”文件，点击“打开”按钮，注册组件，然后可以安装了。

(3) 可以使用“XPForm.ocx”组件的汉化文件“XPForm.npk”对组件进行汉化，在“请选择欲包装的 OCX 组件或类型库”对话框中点击“安装”按钮，选择“XPForm.npk”，按提示完成安装，对“XPForm.ocx”组件进行汉化。或将“XPForm.npk”复制到易语言安装目录的“lib”文件夹中，直接使用“XPForm.ocx”组件，不需汉化了。

3. OCX 组件的使用

新建一个易语言窗口程序，在窗口中添加一个“XP 窗口”组件，在工作夹的支持库面板“XPForm”中，可以看到“XP 窗口”组件两条命令：“打印”、“隐藏最小化按钮”；两个属性：“标题”、“背景颜色”；三个事件：“鼠标被按下”、“最小化被单击”、“被关闭”。

双击窗口，在“_启动窗口创建完毕”下输入代码：“XP 窗口 1. 移动（0，0，_启动窗口. 宽度，，_启动窗口. 高度）”。

运行程序，显示出一个仿 XP 风格的窗口。

4. 类型库的安装

类型库是许多类型信息的集合，涉及类、接口、接口命令、接口命令的参数及相关常量和事件等。类型库通过“ITYPELIB”接口来访问

问，通常位于：*.tlb、*.olb、*.dll、*.ocx、*.sys 等文件中。

安装类型库与安装 OCX 相似：

(1) 选择易语言菜单“工具”——“类型库或 OCX 组件->支持库”，弹出“请选择欲包装的 OCX 组件或类型库”对话框，其列表分为 OCX 组件和类型库两类，选择一个类型库安装。

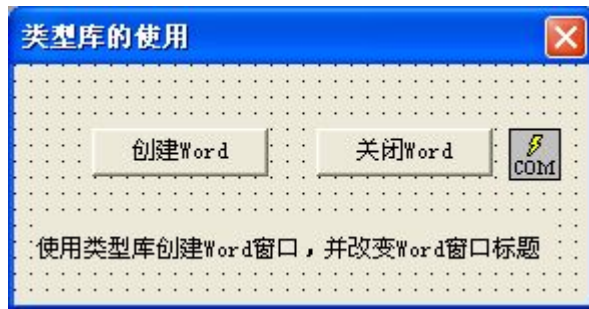
(2) 如果系统中没有需要的类型库，可先到达易语言官网“资源中心”下载或从书籍《中文编程从入门到精通》附带的光盘中安装，注册后可安装类型库了。

(3) 在“请选择欲包装的 OCX 组件或类型库：”中找到相关汉化文件，按提示对类型库进行汉化。

5. 类型库的使用

以“Word 类型库”例程为例：

(1) 新建一个易语言程序，窗口标题为“类型库的使用”；在窗口中添加“创建 Word”、“关闭 Word”两个按钮；一个标签，标题内容为“使用类型创建 Word 窗口，并改变 Word 窗口标题”；再添加一个外部事件组件“Application 事件”。



类型库的使用窗口

(2) 分别双击“创建 W “创建 Word” 按钮和“关闭 Word” 两个按钮，进入代码编辑区，建立一个程序集变量“Word 类型库”，类型为“Application”，其中：Application = 挂接事件；Vsiible = 组件隐藏与显示；Caption = 标题；Quit = 关闭程序。编写如下代码：

窗口程序集名	保 留	保 留	备 注
窗口程序集1			
变量名	类 型	数 组	备 注
Word类型库	Application		

子程序名	返回值类型	公开	备 注
_按钮_创建Word_被单击			

Word类型库.Visible = 真
Word类型库.Caption = “易语言Word”

子程序名	返回值类型	公开	备 注
_按钮_关闭Word_被单击			

Word类型库.Quit(, ,)

子程序名	返回值类型	公开	备 注
_Application事件1_Quit			

信息框(“即将关闭Word” , 0 ,)

子程序名	返回值类型	公开	备 注
_启动窗口_创建完毕			

Word类型库.挂接事件 (Application事件1)

类型库的使用代码

选择菜单“运行”——“运行”，或按下“F5”键，运行程序。

学习中文编程——易语言（三十）：对象和 COM 对象

1. 对象的概念

对象是属性、命令和事件的集合。对象的属性记录了对对象的特征，对象的命令提供了对对象进行某种操作的途径，对象的事件用作通知外部它的状态发生了改变。

易语言是一种基于对象的编程语言，能否熟练地使用易语言，理解对象的概念及使用方法非常重要。

2. 对象型变量的定义

在易语言中，对象可以通过定义数据类型的实例来创建。将一个组件拖放到窗口上，或者在程序中定义一个具有该数据类型的变量。被拖放到窗口上的组件或秘定论的变量就是对象。

通过在对象和属性或命令之间加上“.”，即可引用对象的属性或命令。

在易语言中，对象也可以通过一种抽象的定义方式来创建，该对象变量定义后并不知道具体对象是什么，需要通过对象数据类型提供的命令来实现对象的具体实例。

3. “对象”和“变体型”数据类型

易语言从 3.7 版本开始，在核心支持库数据类型增加了“对象”和“变体型”，通过它们可以方便地引用 COM 对象。

“对象”常用方法有：

- 创建清除类：创建（）、获取（）、清除（）；
- 属性读写类：写属性（）、读属性（）、读**属性（）、读

对象型属性（）：

- 方法调用类：方法（）、通用方法（）、**方法（）、对象型方法（）；

- 其它类：查看（）、取错误（）、是否为空（）、是否相等（）。

“变体型”常用方法有：

- 清除创建类：清除（）、赋值（）、创建数组（）、置类型（）；

- 调取类：取类型（）、取数组成员数（）、取**（）、取对象（）、取变形体（）。

以上**代表基本数据类型，如：文本、数值、逻辑值等。

4.COM 对象

COM (Component Object Model, 组件对象模型)，是一些小型二进制可执行程序，可以给操作系统、应用软件以及其它组件提供服务。

COM是微软为了计算机软件的生产更加符合人类的行为方式开发的一种新型软件开发技术，在COM构架下，人们可以开发出各种各样的功能专一的组件，然后将它们按照需要组合起来，构成复杂的应用系统。

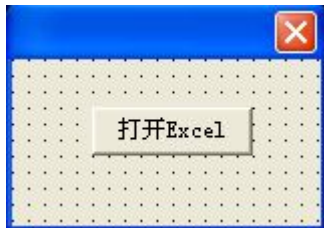
COM对象是遵循COM规范编写，以Win32动态链接库（DLL）或可执行文件（EXE）形式发布的可执行二进制代码，能够满足对组件框架的所有需求。

5. COM 对象的应用

COM 对象有自己的属性和方法，但这些都被 COM 封装起来了，客户只有通过接口才能对 COM 的方法进行调用。在 Windows 系统下，有大量的 COM 对象存在，是一个庞大的 COM 对象库，充足的组件给开发人员提供了便利。

下面，针对 COM 中的 Excel 对象做一个实例，打开一个 EExcelxcel 文件：

(1) 运行易语言程序，在窗口添加一个“打开 Excel”按钮。



打开 Excel 窗口

(2) 双击“打开 Excel”按钮，进入代码编辑区，定义“Excel 对象”和“工作表对象”两个变量，类型为“对象”。输入如下代码：

窗口程序集名	保 留	保 留	备 注
窗口程序集1			
变量名	类 型	数 组	备 注
Excel对象	对象		
工作表对象	对象		

子程序名	返回值类型	公开	备 注
_按钮1_被单击			

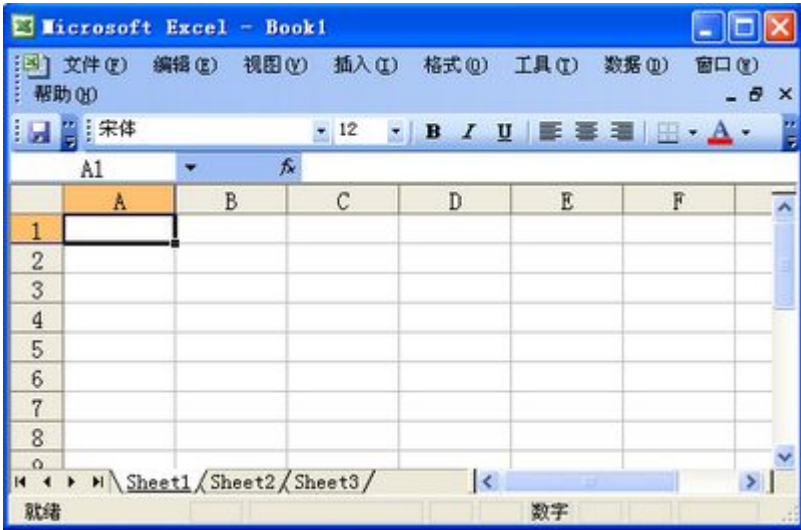
Excel对象.创建 (“Excel.application”,)
Excel对象.写属性 (“Visible”, 真)
工作表对象 = Excel对象.读对象型属性 (“workbooks”,)
工作表对象.方法 (“open”, 取运行目录 () + “\Book1.xls”)

子程序名	返回值类型	公开	备 注
_启动窗口_将被销毁			

Excel对象.方法 (“Quit”,)
Excel对象.清除 ()

打开 Excel 实例代码

运行程序，点击“打开 Excel”按钮，程序打开了一张 Excel 表格。



程序打开了一张 Excel 表格

学习中文编程——易语言（三十一）：面向对象

1. 面向对象简述

本文的面向对象和上篇博文中 CON 对象中的“对象”的概念是不同的。

面向对象(Object Oriented, 简称 OO)是当前最流行的编程方式。面向对象的概念和应用已超越了程序设计和软件开发,扩展到很宽的范围,如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

早期的计算机编程是基于面向过程的方法,例如实现算术运算 $1+1+2 = 4$, 通过设计一个算法就可以解决当前的问题。随着计算机技术的不断提高,计算机被用于解决越来越复杂的问题。通过面向对象的方式,将现实世界的物,抽象成对象;现实世界中的关系抽象成类。通过面向对象的方法,以理解的方式对于复杂系统进行分析、设计与编程。

面向对象的思想涉及到软件开发的各个方面。如:面向对象的分析 (OOA, Object Oriented Analysis)、面向对象的设计 (OOD, Object Oriented Design)以及我们经常说的面向对象的编程实现 (OOP Object Oriented Programming) 等。

2. 类的概念与创建

要了解面向对象首先了解类的概念。

对象是人們要研究的任何事物,具有相同或相似性质的对象的抽象化就是类。类可以简单的看成封装了各种数据和方法的模块。类据有的特有个性,是面向对象编程的重要组成部分。把对象比作房子,类

就有设计图纸的特性，所以面向对象编程的重点是类的设计。

类的创建可以通过菜单“插入”——“类模块”严实现，也可在工作夹程序面板中“程序数据”上单击右键，选择“新类模块”项。



从程序面板创建类模块

插入类后，会自动生成“_初始化（）”和“_销毁（）”两个方法，相当于其它编程语言中的“构造函数”和“析构函数”，基于该类的对象创建和销毁时自动调用。

类名	基 类	公开	备 注
类5			

方法名	返回值类型	公开	备 注
_初始化			当基于本类的对象被创建后，此方法会被自动调用

方法名	返回值类型	公开	备 注
_销毁			当基于本类的对象被销毁前，此方法会被自动调用

类的健

在易语言中类的创建和释放过程是自动的，创建顺序为：先创建基类对象，再创建继承类对象；释放顺序为：先释放继承类对象，再释放基类对象。

3. 类的特性

(1) 类的封装性

类可以看作一个类型，这种类型是由编程者定义的。该类类型的内部结构包括数据和行为，行为称为该类的方法；用来存放该类数据的是变量，是该类的成员。

易语言中类的成员是私有的，只能被类自身的方法调用，不能被其它类所调用。

类的封装是一种信息隐蔽技术，使数据和操作数据的方法成为一个整体，目的是把对象的设计与对象的使用分离，使用者只能看到对象的外部特性，而对象的特性对使用者是隐蔽的。

(2) 类的继承性

一个类的基类叫父类，一个类的父类可派生出许多子类，这些子类是父类的派生类，也叫其父类的继承类。

类的继承性是子类自动共享数据和方法的机制，类的对象是各自封闭的，如果没有继承性机制，则类对象中的数据、方法出现大量重复。继承不仅支持系统的可重用性，而且还促进系统的可扩充性。

（3）类的多态性

易语言中，类的多态性是指父类和子类中，都拥有定义相同（相同名称、相同参数）的方法，这些方法执行不同的操作，从而实现“一接口，多种实现”。

学习中文编程——易语言（三十二）：易语言模块

1. 易模块简述

模块，是为完成某一功能所需的一段程序。将一些通用的功能集成在一起，供其它程序调用，可简化软件系统的复杂性，具有独立功能，可以复制、拆分，能重复使用。

易语言也提供了模块化的开发支持——“易模块”。易模块是一段特殊格式的易代码，所有代码都封装在模块中，模块的使用者只能看到公开的接口结构，无法看到实现功能的代码。通过使用易模块，编程人员可以将常用的易代码封闭起来，并重复使用到其它易语言程序中，也可提供给第三方用于易语言程序的开发。

2. 易模块的开发

（1）选择菜单“程序”——“新建”，在弹出的“新建”对话框中双击“Windows 易语言模块”，新建一个易模块程序。



新建一个易模块程序

(2) 在代码编辑区，易语言自动创建了“_启动子程序”和“_临时子程序”两个子程序，其中“_临时子程序”用作调试易模块，不会被编译到易模块文件中。

(3) 新建一个子程序，改名为“信息框子程序”，并在公开属性上打勾，输入以下代码：

“信息框（“第一个易语言模块”，0）”

(4) 在“_临时子程序”中，输入：

“信息框子程序（）”

程序集名	保 留	保 留	备 注
程序集1			

子程序名	返回值类型	公开	备 注
_启动子程序	整数型		请在本子程序中放置易模块初始化代码

_临时子程序 () ’ 在初始化代码执行完毕后调用测试代码
 返回 (0) ’ 可以根据您的需要返回任意数值

子程序名	返回值类型	公开	备 注
_临时子程序			

’ 本名称子程序用作测试程序用，仅在开发及调试环境中有效，编译发布程序前将被系统自动清空，请将所有用作测试的临时代码放在本子程序中。
 ***注意不要修改本子程序的名称、参数及返回值类型。

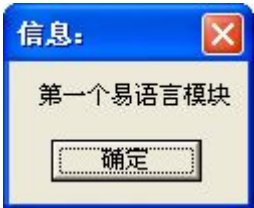
信息框子程序 ()

子程序名	返回值类型	公开	备 注
信息框子程序		✓	

信息框 (“第一个易语言模块”, 0,)

易模块代码

(5) 按 F5 键运行，可以看到一个显示“第一个易语言模块”文本信息框。



“第一个易语言模块”文本信息框

3. 易模块的编译

(1) 选择菜单“程序”——“配置”，打开“程序配置对话框”，

设置通用、作者信息、其它三个选项卡中信息。

(2) 选择菜单“编译”——“编译”选项，在弹出的保存对话框文件名中输入“第一个易语言模块”，保存后缀名为“ec”的文件。一个简单的易语言模块编写和编译完毕。

4. 易模块的调用

易模块只有正确引用到易语言程序中才可以被调用：

(1) 新建一个易语言程序，在工作夹程序面板中双击“易模块引用表”，在弹出的“易模块选择对话框”中选择刚编写的“第一个易语言模块”。

(2) 打开“第一个易语言模块”，“易模块引用表”中出现该模块。双击该模块，弹出“易模块公开信息”对话框，显示出所选模块的说明和模块接口的调用说明。

(3) 新建一个易语言程序，在易语言“_启动窗口”添加一个按钮，双击该按钮，在“_按钮 1_被单击”子程序下输入如下代码：

“信息框子程序 ()”

按 F5 运行易语言程序，出现下面左侧窗口，点击“确定”按钮，出现右侧“第一个易语言模块”信息框。比较右侧信息框和上面“第一个易语言模块”文本信息框，其实是同一个。



左侧易程序窗口，右侧调用模块显示的信息框

学习中文编程——易语言（三十三）：易语言向导

1. 易语言向导简介

向导是一种用来简化编写、开发、操作的程序，按照提示引导用户一步一步操作，最后得到用户需要的结果。

易语言向导是引导用户在新建程序或当前已经打开的程序中添加或插入自己需要的一些代码。易语言向导的功能就是提供一段完整的源代码，然后使用一个向导程序来引导用户在自己的程序中导入或插入已经写好的源程序代码。

2. 易语言向导支持库命令

易语言向导的制作，需要使用易语言向导支持库提供的命令，通过这些命令，可以对程序集、子程序、变量、常量、数据类型、DLL命令、窗口、组件等进行操作。

（1）复制程序（）

命令的原型是：“<无返回值>复制程序（程序项类型 欲复制项类型，文本型欲复制项名称，文本型 新项目名称）”。

（2）复制程序段（）

命令的原型是：“<无返回值>复制程序段（文本型 子程序名称，文本型标记名，文本型 复制到标记名）”。

（3）删除程序（）

命令的原型是：“<无返回值>删除程序（程序项类型 欲删除项类型，文本型欲删除项名称）”。

（4）删除程序段（）

命令的原型是：“<无返回值>删除程序段（文本型 子程序名称，文本型标记名）”。

（5）删除标记程序（）

命令的原型是：“<无返回值>删除标记程序（文本型 标记名）”。

（6）修改程序（）

命令的原型是：“<无返回值>修改程序（程序项类型 欲修改项类型，文本型欲修改项名称，程序项属性 欲修改属性，通用型 修改值 ）”。

（7）置组件属性（）

命令的原型是：“<无返回值>置组件属性（文本型 组件属性名，通用型属性值）”。

（8）置语句备注（）

命令的原型是：“<无返回值>置语句备注（文本型 子程序名称，文本型标记名，[文本型 备注]）”。

（9）置程序信息（）

命令的原型是：“<无返回值>置程序信息（[文本型程序名称]，[整数型 程序类型]，[文本型 备注]，[文本型版本]，[字节集图标]）”。

（10）置作者信息（）

命令的原型是：“<无返回值>置作者信息（[文本型 作者名称]，[文本型邮政编码]，[文本型 联系地址]，[文本型 电话]，[文本型 传真]，[文本型 电子邮箱]，[文本型主 页地址]，[文本型其他信息]）”。

(11) 定义模板变量 ()

命令的原型是：“<无返回值>定义模板变量（文本型 模板变量名称，[通用型模板变量值]）”。

(12) 删除模板变量 ()

命令的原型是：“<无返回值>删除模板变量（文本型模板变量名称）”。

(13) 清除修改记录 ()

命令的原型是：“<无返回值>清除修改记录 ()”。

(14) 写出程序 ()

命令的原型是：“<无返回值>写出程序（字节集 模板程序数据，[逻辑型是否关闭源程序]）”。

(15) 添加模块引用 ()

命令的原型是：“<无返回值>添加模块引用（文本型模块文件路径名）”。

(16) 删除模块引用 ()

命令的原型是：“<无返回值>删除模块引用 ()”。

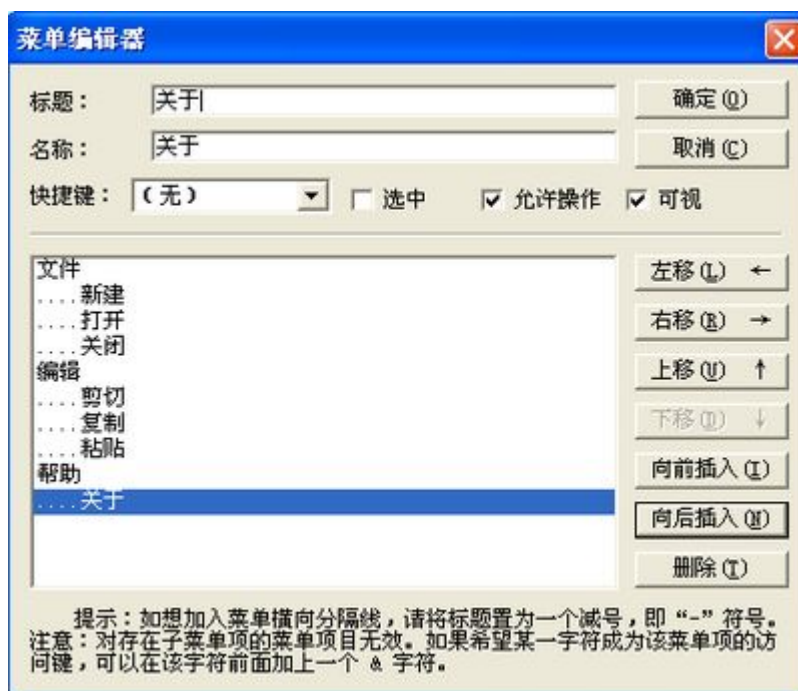
3. 易语言向导的编写

现编写一个自动生成菜单的易语言向导。分两个步骤：首先根据软件需要，设置模板程序；然后就是编辑模板程序的代码。

(1) 设置模板程序

● 建立一个易语言程序，建立模板程序菜单：单击菜单“工具”——“菜单编辑器”，弹出“菜单编辑器对话框，创建“文件”菜单，

下面再创建“新建”、“打开”、“关闭”三个子菜单；创建“编辑”菜单，下面再创建“剪切”、“复制”、“粘贴”三个子菜单；创建“帮助”菜单，下面再创建“关于”子菜单。单击“确定”按钮，菜单创建完毕，将其保存。



菜单编辑器对话框

●新建一个易语言程序，在“_启动窗口”添加“完成”、“取消”两个按钮，用作生成向导程序；添加四个选择列表框，用作显示菜单项目。“选择列表框 1”显示顶级菜单“文件”、“编辑”、“帮助”；“选择列表框 2”显示“文件”菜单的子菜单“新建”、“打开”、“关闭”；“选择列表框 3”显示“编辑”菜单的子菜单“剪切”、“复制”、“粘贴”；“选择列表框 4”显示“帮助”菜单的子菜单“关于”。

“粘贴”；”；“选择列表框 4”显示“帮助”菜单的子菜单“关于”。

●将四个选择框的大小设置一样，“选择列表框 3”和“选择列表框 4”的可视属性设为假，调整“选择列表框 2”的层次为顶层。

●用程序代码实现：点击“选择列表框 1”中的某一项目，将含有其相关子菜单的“选择列表框”的可视属性设为真，其它两个“选择列表框”的可视属性设为假，这样，可以选择“选择列表框 1”中的选中项目的子菜单。根据选择情况，决定保留还是删除菜单。



应用程序向导界面

●将“应用程序向导”保存，并导入“资源表”，如下图：

图片或图片组名称	内容	公开	备注
应用程序向导	2672		1

将应用程序向导导入资源表

(2) 编写对模板程序进行修改的代码

●选中“选择列表框 1”中的“文件”，程序窗口右侧显示包含“文件”子菜单的“选择列表框 2”。首先判断“选择列表框 1”中的“文件”项目是否被选中，如果没选中，就将“文件”及其对应的子菜单项全部删除；如果选中，再判断“选择列表框 2”中“文件”菜单对应的子菜单是否被选中，哪一项子菜单没被选中，就删除哪一项子菜单，如果都没选中，则“文件”菜单及其对应的子菜单全部删除。代码如下：

```

判断 (选择列表框1.是否被选中 (0) = 真) ' 文件菜单
    如果真 (选择列表框2.是否被选中 (0) = 假)
        删除程序 (#程序项类型.窗口组件, "_启动窗口.新建")
    如果真 (选择列表框2.是否被选中 (1) = 假)
        删除程序 (#程序项类型.窗口组件, "_启动窗口.打开")
    如果真 (选择列表框2.是否被选中 (2) = 假)
        删除程序 (#程序项类型.窗口组件, "_启动窗口.关闭")
    如果真 (选择列表框2.是否被选中 (0) = 假 且 选择列表框2.是否被选中 (1) = 假 且 选择列表框2.是否被选中 (2) = 假)
        删除程序 (#程序项类型.窗口组件, "_启动窗口.文件")
    删除程序 (#程序项类型.窗口组件, "_启动窗口.文件")
    删除程序 (#程序项类型.窗口组件, "_启动窗口.新建")
    删除程序 (#程序项类型.窗口组件, "_启动窗口.打开")
    删除程序 (#程序项类型.窗口组件, "_启动窗口.关闭")

```

编写文件菜单代码

●选中“选择列表框 1”中的“编辑”，程序窗口右侧显示包含“编辑”子菜单的“选择列表框 3”。首先判断“选择列表框 1”中的“编辑”项目是否被选中，如果没选中，就将“编辑”及其对应的子菜单项全部删除；如果选中，再判断“选择列表框 3”中“编辑”菜单对应的子菜单是否被选中，哪一项子菜单没被选中，就删除哪一项子菜单，如果都没选中，则“编辑”菜单及其对应的子菜单全部删除。代码如下：

```
判断 (选择列表框1.是否被选中 (1) = 真) ' 编辑菜单
-- 如果真 (选择列表框2.是否被选中 (0) = 假)
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.剪切")
-- 如果真 (选择列表框2.是否被选中 (1) = 假)
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.复制")
-- 如果真 (选择列表框2.是否被选中 (2) = 假)
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.粘贴")
-- 如果真 (选择列表框2.是否被选中 (0) = 假 且 选择列表框2.是否被选中 (1) = 假 且 选择列表框2.是否被选中 (2) = 假)
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.编辑")
删除程序 (#程序项类型.窗口组件, "_启动窗口.编辑")
删除程序 (#程序项类型.窗口组件, "_启动窗口.剪切")
删除程序 (#程序项类型.窗口组件, "_启动窗口.复制")
删除程序 (#程序项类型.窗口组件, "_启动窗口.粘贴")
```

编写编辑菜单代码

●选中“选择列表框 1”中的“帮助”，程序窗口右侧显示包含“帮助”子菜单的“选择列表框 4”。首先判断“选择列表框 1”中的“帮助”项目是否被选中，如果没选中，就将“帮助”及其对应的子菜单项全部删除；如果选中，再判断“选择列表框 4”中“帮助”菜单

对应的子菜单“关于”是否被选中，如果都没选中，则“帮助”菜单及子菜单“关于”全部删除。代码如下：

```
判断 (选择列表框1.是否被选中 (2) = 真) ' 帮助菜单
-- 如果真 (选择列表框2.是否被选中 (0) = 假)
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.帮助")
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.关于")
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.帮助")
-- 删除程序 (#程序项类型.窗口组件, "_启动窗口.关于")
```

编写帮助菜单代码

●当所有对模板程序的修改代码完成后，要用“写出命令（）”将修改后的程序代码写到新建的易语言程序中，写出的程序结果存放到“结果文本”中。判断“结果文本”是否为空，就知道向导程序是否创建成功。向导最终写出程序的代码是：

```
结果文本 = 写出程序 (#向导程序, )
-- 判断 (结果文本 = "")
-- 信息框 ("已创建应用程序", 0, )
-- 信息框 (结果文本, 0, )
销毁 ()
```

最终写出程序的代码

●将编译好的向导程序，复制到易语言安装目录下的 Wizard 文件夹中，就可以使用了。

4. 易语言向导的使用方法

易语言向导的使用可以说是“傻瓜式”的，根据安装向导提示，选择需要的项目，一直向下进行。

运行易语言，选择菜单“程序”——“新建”，选中“通用向导创建”——“常用向导”，就可以看到 Wizard 文件夹中的所有向导程序。根据需要，选择相应的向导程序，按提示完成向导程序操作。

学习中文编程——易语言（三十四）：易语言调试

1. 程序调试的作用

程序调试，是在开发的程序进入实际使用前，用手工或编译等方法进行测试，修正语法错误或逻辑错误的过程。

语法错误是指程序代码不符合易语言的语法规范。这种错误容易发现修正。首先在代码输入的时候，易语言系统检查并发现一些错误；其次在程序运行时，当执行到语法错误的代码行，光标停止在错误行并在状态夹面板显示错误原因。可见，易语言系统会找出绝大部分语法错误，只要按照其提示信息进行相应的修改即可。

逻辑错误是指程序流程、数据处理上的错误。含有逻辑错误的程序能够正常执行，只是执行结果与程序最终完成的结果不一致。这类错误易语言系统不会发现，只有编程者自己查找，实际应用中，通常用“断点”、“跟踪”、“查改变量”等方法结合使用，以确定错误的地方。

2. 调试步骤

程序调试的一般步骤是：

（1）编写代码时，每输入一行代码，易语言系统就会对输入的代码进行语法检查。

（2）执行所有的程序功能，从中找出并修改易语言系统找出的语法错误。

（3）如果调试运行后，程序结果与预想的不一致，通过分析判断，找到可能有逻辑错误的代码段，在前面设置“断点”。

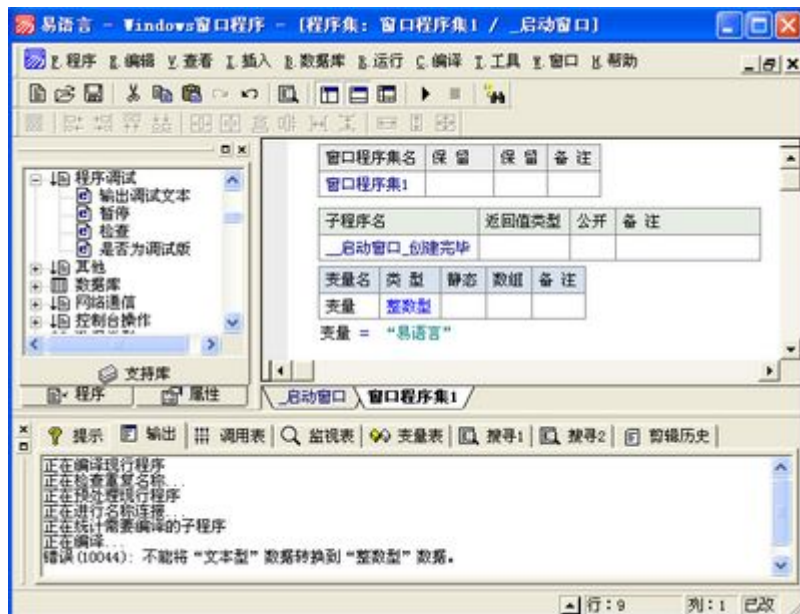
(4) 当程序在“断点”处中断后，使用“跟踪”、“查改变量”等方法，检查运行过程中每一步是否执行正确，变量的数据是否有误。

(5) 找到错误后，终止程序运行，修改代码。继续调试，直到程序所有功能全部正常为止。

3. 预编译调试

每输入一行程序代码，换行时，易语言都会对这一行代码进行预编译调试。如果发现有明显的语法错误存在，就会将光标自动移动到错误行，并在状态夹的输出面板中提示错误原因，可以根据光标停留位置和错误原因提示改正错误。

虽然易语言系统提供了预编译调试功能，但只能指出输入中的格式错误，如数据类型不匹配、指定的变量或组件未找到等。对于编程上的逻辑错误。易语言系统就不会发现，需要用其它调试方法和调试命令，找到错误代码。



预编译调试

4. 运行中的调试

(1) 运行调试菜单

易语言提供的运行调试，可在易语言菜单“运行”中查看，主要有：

- 查看表达式/变量：在调试过程中，查看指定变量值，并可加入到状态夹监视表中。

- 单步跟踪：在程序现行运行位置单步执行一程序，此程序调用了子程序，则跟踪进入子程序。

- 跟踪返回：在上级子程序调用现行子程序的语句后中断。在被调用子程序中设置断点，断点行被跟踪后，跳出当前子程序，系统跟踪至上级程序调用当前子程序语句的下一行代码。

- 执行到光标处：运行易语言，程序中断在当前光标处。

●设置或清除断点：设置或清除当前程序处的断点。如果没有断点，就为当前行设置一个断点；如果有断点，就会取消当前行的断点。

●条件断点：设置条件断点，即在断点中加入逻辑判断，当符合条件时，断点生效。

●清除所有断点：清除程序中的所有断点。

●到现行执行位置：跳到现行即将被执行语句的位置。

(2) 运行调试步骤

第 1 步：执行到光标处

将光标移动到被调试的程序代码行，选择“执行到光标处”菜单项，或按下“Ctrl+F8”键，程序会记录光标所在位置，并自动运行，当执行到光标所在位置，程序中断，并将光标移到该行代码处，在前面用黄色箭头提示。这时就可能使用跟踪调试、改变变量等功能，对变量值进行监视。

第 2 步：断点调试

将光标移动到被调试的程序代码行，选择“设置或清除断点”菜单项，或按下“F9”键，在光标所在行前面用红色箭头提示。

还可以设置“条件断点”，在断点处进行逻辑判断，只有符合了“条件断点”设置的条件，断点才会生效。在“条件断点设置对话框”中，达式可以使用“且”或“或”来连接，还可以包括四则运算。对话框中“中断前跳过次数”表示，程序会在断点条件成立后，跳过指定次数的断点，然后生效。

再按下“F9”键，清除原有的断点。

第 3 步：跟踪调试

当程序中断后，按下 “F8” 键，光标及黄色箭头会向下一句代码行移动，执行下一句代码。不断按 “F8” 键，可以使程序一步一步执行，直到所在子程序所有代码执行完毕。

如果需要跟踪到调用子程序时，可以使用 “单步跟踪”；从子程序里跳出，可以使用 “跟踪返回”。

第 4 步：查改变量

编写程序时，变量值常出现错误，而且很难查找。易语言在状态夹加入了 “监视表” 和 “变量表”，有效同时监控多个变量值在程序运行中的变化，或多个子程序中的变量值的变化。

当程序执行到被中断的代码时，选择 “查看表达式/变量” 或按下 “Shift+F9” 键，就会弹出 “查看/修改” 对话框，输入变量名称，点击 “计算” 按钮，就会看到变量的值；点击 “加入系统监视表”，就会将变量加入到 “监视表” 中。

5. 程序调试命令

为方便调试，易语言核心支持库提供了程序调试命令。命令只在调试运行中有效，编译后会被自动舍弃。

（1）输出调试文本（）

命令的原型是：“<无返回值>调试输出文本（通用型准备输出的调试文本信息，...）”。

本命令用于在输出面板中显示一个调试信息。输出时在内容前自动添加*号，在末尾自动添加换行符。

（2）暂停（）

命令的原型是：“<无返回值>暂停（）”。

本命令用于暂时终止程序运行。效果和设置中断点相同，可以查看各变量的值。

（2）检查（）

命令的原型是：“<无返回值>检查（逻辑型 被校验的条件）”。

本命令用于检查指定条件、逻辑变量或逻辑值。参数指定了欲检查的内容，如果为“假”，则程序中断调试模式；如果为“真”，则继续向下执行。

（2）是否为调试版（）

命令的原型是：“<逻辑型>是否为调试版（）”。

本命令用于检查所在程序是否在调试模式行。如果当前程序执行文件为易语言编辑环境调试运行的程序，返回“真”，否则表明为发布版本，返回“假”。

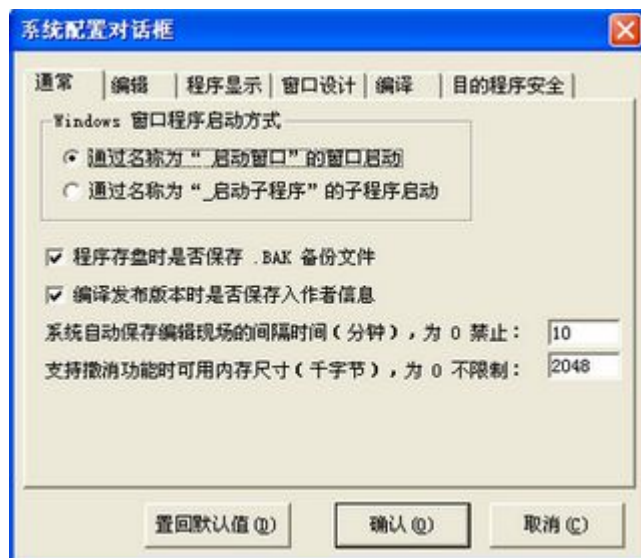
学习中文编程——易语言（三十五）：易语言编译发布

程序编写完毕后，需要对其进行编译，这样才能在其它电脑上安装运行。这是编程的最后一项工作。

1. 编译前的配置

（1）系统配置

选择菜单“工具”——“系统配置”，打开“系统配置对话框”，该对话框有“通常”、“编辑”、“程序显示”、“窗口设计”、“编译”、“目的程序安全”六个选项卡，根据编译需要设置选项。



系统配置对话框

（2）程序设置

选择菜单“程序”——“配置”，打开“程序配置对话框”，

该对话框有“通常”、“作者信息”、“其它”三个选项卡，“通常”项可设置程序名称、程序描述、程序备注、程序版本号，还可为程序设置图标；作者信息可设置作者姓名、联系方式；其它项可设置程序密码等。



程序配置对话框

2. 编译

编译时，程序源码部分进行编译，而不包含易语言的支持库。编译后必须在安装了易语言的电脑中使用。

编译方法是：选择菜单“编译”——“编译”选项，或按下F7键，在弹出的对话框中选择要保存的文件路径并输入目标文件名，保存文件。编译后，在状态夹输出面板中有编译成功的信息，以及运行

时需要的文件列表。

在发布时，需要将以上列出的文件与目标文件一同提供给用户。

3. 静态编译

静态编译时把支持库中被程序使用到的有效代码 按实际需要链接到目标文件中，打包编译。

静态编译方法是：选择菜单“编译”——“静态编译”选项，或按下 Shift+F7 键，在弹出的对话框中选择要保存的文件路径并输入目标文件名，保存文件。静态编译前，必须指定使用的外部链接器，并给出完整路径。

4. 编译生成安装软件

为方便程序的发布，易语言提供了标准安装软件制作的向导程序：

（1）选择菜单“编译”——“编译生成安装软件”选项，或按下 Ctrl+Shift+F7 键继续；

（2）确定保研路径和文件名，点击“确定”按钮继续；

（3）在弹出的对话框中添加安装软件标题、自述文件、许可协议，点击“下一步”按钮继续；

（4）设置安装窗口外观，点击“下一步”按钮继续；

（5）选择需要加入到安装的相关文件，默认全选，点击“下一步”按钮继续；

（6）添加其它文件到不同的目录中，点击“下一步”按钮继续；

- (7) 设置软件启动和卸载方式，点击“下一步”按钮继续；
- (8) 设置安装目录及相关选项，击“下一步”按钮继续；
- (9) 确定安装文件属性，添加输出安装文件名，点击“确认”按钮，完成安装软件制作完成。

发布制作的安装软件，用户可以用它将编写的软件安装在自己的电脑中。

学习中文编程——易语言（三十六）：编程之路

学习中文编程——易语言”共发了 36 篇博文，这是最后 1 篇。其实这是我整理的学习易语言的笔记，分为以下几个部分：

第 1 篇可以作为引言。

第 2 篇是易语言简述。

第 3—10 篇是易语言基础知识，主要内容包括：易语言工作界面、程序集和子程序、数据类型、变量、常量、资源表、运算符和表达式等。

第 11—13 篇是易语言组件，组件好像积木，每一个组件就是一块形状不同、功能各异的积木。积木互相配合，可以搭建漂亮的造型，组件互相结合，可以组成完美的程序窗口，实现可视化编程。

第 14—18 篇是易语言命令，上网查了一下，据说易语言命令有 9340 多条，常用命令也有几百条。要写代码，又离不开命令，这足以使人眼花缭乱，头疼不已。

第 19—22 篇是易语言数据库，易语言数据库就是一个表，由记录（行）和字段（列）组成，以类似表格的形式存储、管理数据，简单易用，但不能说功能强大。

第 23—26 篇是外部数据库应用，易语言支持以 ODBC 方式和 ADO 方式操作外部数据库，可供使用的数据库有很多，从兼容性、宜用性考虑，我选择了 SQL Server 数据库和 Access 数据库学习。应用外部数据库，就要用到 SQL 语句，这又使我陷入面对英文的尴尬境地。

第 27—31 篇是易语言的高级应用，分析了 API 应用程序接口、DLL 动态链接库、OCX 组件和类型库、对象和 COM 对象、面向对象。

第 32—35 篇是易语言的扩展功能，简述易语言模块、易语言向导、易语言调试、易语言编译输出。

第 36 篇就算结束语吧。

学习编程的难点主要有两个：一是语言障碍，编程多用英文，没有英语基础的人学编程，确实很困难；二是编程思路，编程要有较强的逻辑思维能力、描述能力；还要具备一定的数学基础。正因如此，易语言虽然是中文编程软件，但学习起来还是有一定难度。

当我们日常操作电脑时，是别人编写好的软件我们使用，而编程则是我们编写好软件供别人使用，正是这一逆向操作，把许多电脑爱好者挡在编程门槛之外。

编程好比是创作，中文编程也是如此。我们都可以用汉字写一篇文章，可长可短，这不算太难；但让我们象作家一样写一篇长篇小说，大多数人无从下笔了。

熟练的掌握一门编程技巧，不是一朝一夕的事，有人说：“学好编程，要写过十万行以上的代码”，没有什么捷径，这是编程的必经之路！



学好编程，要写过十万行以上的代码

原文地址: 

http://blog.sina.com.cn/s/articlelist_2327901543_7_1.html