



Python图形用户界面设计

Delphi For Python高级开发教程



Delphi是古希腊神话中智慧、战争、技艺、文艺女神雅典娜（Athena）的圣所，也是大地之母盖娅（Gaia）的神庙，她委派自己所生的巨蟒（Python）在Delphi守护神谕（Oracle），镇守世界的中心。

目录

简介	5
谁应该阅读这本教程	5
第1章 Python与Delphi	6
什么是Delphi4Python	6
1.1 Python语言介绍	7
1.1.1 了解Python	7
1.1.2 Python的应用领域	9
1.2 Delphi语言介绍	9
1.2.1 Delphi的历史	9
1.2.2 如何学习Delphi开发	16
1.2.3 如何安装Delphi社区版	16
1.3 Delphi与Python之道	17
1.3.1 Delphi神谕与Python禅语	17
1.3.2 Delphi神谕	17
1.3.3 Python禅语	18
1.3.4 Delphi与Python之间的区别	19
1.3.5 Delphi和Python的开发层级	20
1.3.6 Python运行Delphi VCL和Delphi FireMonkey框架	21
1.3.7 立刻开始使用DelphiVCL for Pyhton和DelphiFMX for Pyhton	25
第2章 Delphi快速入门	26
2.1 Delphi语言介绍	26
2.1.1 Delphi集成开发环境	26
1. 主表单窗口(Main Form)	27
2. Structure(对象体系浏览器)	27
3. Object Inspector(对象检阅器)	27
4. 表单窗体口(Form)	28
5. 代码编辑器	28
6. UI界面设计器	28
2.1.2 Delphi项目的主要文件	28
(1) 项目文件(.dpr)	28
(2) 表单窗体文件(.dfm)	29
(3) 单元文件(.pas)	30
(4) 资源文件(.res)	31
2.2 Delphi的主要数据类型	31
1. 整数类型	31
2. 浮点数类型	32
3. 字符和字符串类型	32
4. 布尔类型	32
5. 指针类型	32
2.3 Delphi程序结构	32

2.3.1 知识讲解	34
1. 顺序结构语句	34
2. 选择结构语句	36
3. 循环语句	38
2.3.2 if语句在程序编写中的应用	41
1. if... then语句	41
2. if... then... else语句	42
3. if嵌套语句	43
2.3.3 Case语句在程序编写中的应用	45
2.3.4 Repeat... Until语句在程序编写中的应用	47
2.3.5 While... Do语句在程序编写中的应用	49
2.3.6 For循环语句在程序编写中的应用	50
2.3.7 跳出条件语句或循环语句	65
2.4 数组在程序开发中的应用	67
2.4.1 数组类型	67
1. 一维数组	69
2. 二维数组	70
3. 动态数组	71
2.5 异常处理	72
2.5.1 知识讲解	73
1. 异常处理机制	73
2. Try... Except结构	73
3. Try... Finally... end结构	74
第3章 函数在项目开发中的应用	75
3.1 函数在项目中的应用	75
3.1.1 知识讲解	76
3.2 字符与字符串函数在项目中的应用	77
3.2.1 知识讲解	77
3.2.2 去空格函数在程序中的应用	80
3.2.3 判断用户输入的Email地址是否合法	81
3.2.4 居民身份证位数转换	87
3.3 日期时间类函数在工程开发中的应用	91
3.3.1 知识讲解	91
3.4 文件处理函数在项目开发中的应用	94
3.4.1 知识讲解	94
第4章 可视化UI软件界面设计	96
4.1 标签组件	96
4.1.1 知识讲解	96
4.2 编辑类组件	98
4.2.1 知识讲解	98
4.3 按钮类组件	100
4.3.1 知识讲解	100

4.4 修饰类组件	103
4.4.1 知识讲解	103
4.5 列表类组件	104
4.5.1 知识讲解	105
4.6 进度与刻度类组件	109
4.6.1 知识讲解	109
4.6.2 用TStatusBar组件显示系统状态	110
4.7 分组类组件	110
4.7.1 知识讲解	110
4.7.2 利用TPanel组件对窗体进行页面布局	112
4.7.3 利用TPageControl组件实现多页程序设计	113
4.8 图表类组件	113
4.8.1 知识讲解	113
第5章 搭建Delphi4Python GUI开发环境	116
5.1 Python的下载和安装	116
5.1.1 Python开发环境简介	116
5.1.2 从Python官网下载最新版	116
1. 下载软件	116
2. 安装步骤	117
3. 检查Python是否安装成功	120
4. Python安装失败的解决方法	121
5.2 DelphiVCL for Python安装步骤	123
5.2.1 安装	123
5.2.1 测试	124
5.3 Python开发最佳IDE工具PyScripter	126
5.3.1 使用PyScripter帮助创建 Python GUI	127
5.3.2 Python4Delphi简介	127
5.3.3 Python GUI 中 DelphiVCL框架	127
5.3.4 Delphi本机原生GUI的巨大优势	128
第6章 Delphi4Python GUI开发指南	129
6.1 VCL指南1: 一个空的、最简单的应用程序	129
6.2 FMX指南1: 一个空的和最简单的应用程序	131
6.3 VCL指南2: Hello, DelphiVCL for Python	131
6.4 FMX指南2: Hello, Delphi FMX for Python	134
6.5 VCL指南3: TODO任务应用程序	135
6.6 FMX指南3: TODO任务应用程序	139
6.7 界面风格皮肤样式	141
6.7.1 VCL样式	142
6.7.2 VCL样式列表	143
6.7.3 FMX样式	151
6.7.4 FMX样式列表	152
6.8 样式总结	162

第7章 Delphi4Python开发案例	162
7.1 基础Demo案例	162
7.2 Windows中Delphi For Python使用机器学习案例	167
7.2.1 在Windows的Python4Delphi中使用Matplotlib	168
7.2.2 在Windows的Python4Delphi中使用NLTK	170
7.2.3 在Windows的Python4Delphi中使用Pillow	171
7.2.4 在Windows的Python4Delphi中使用OpenCV	172
7.2.5 在Windows的Python4Delphi中使用Keras	174
7.3 案例：用Delphi设计GUI利用Fastai制作最先进的深度学习应用程序	176
7.3.1 fastai库简介	176
7.3.2 安装fastai库	178
7.3.3 为fastai库构建Delphi GUI	179
7.3.4 在Delphi用程序中使用fastai执行深度学习	181
7.3.5 使用fastai和Delphi实现图像分类的深度学习案例	182
1. 加载带有标签的图像数据集	182
2. 训练深度学习模型	183
3. 使用fastai的效果	189
7.4 案例总结	195
附录	
关于Embarcadero Technologies	195
关于PyScripter	195
关于Delphi	196

简介

欢迎使用 DelphiFMX 和 DelphiVCL 进行 Python GUI图形用户界面开发。这些强大的框架库是使用Python为Windows和其他多个平台跨平台构建本地应用程序的完美选择。这些框架库是为Python开发者设计的，因此只需要读者具备少量的Delphi知识就可以使用，为此本教程提供了一个精简的Delphi初学者快速学习教程，读者很容易就能掌握。

本教程提供了Delphi4Python架构和平台的概述，并指导读者安装和学习使用这两个框架库，通过学习这本教程，读者可以快速掌握当今业界最先进的本地原生跨平台开发工具Delphi和最流行的AI人工智能机器学习语言Python。

- DelphiFMX for Python
- DelphiVCL for Python

本教程还提供了丰富的代码示例和案例代码，帮助读者快速用于实际工作中。最后，本教程以实际案例深入探索了Delphi和Python混合开发的可能性。

谁应该阅读这本教程？

用于Python的DelphiFMX和DelphiVCL库是由Embarcadero的编程语言和工具集Delphi提供的。然而，使用这些框架库不需要对Delphi有任何熟悉或经验，而是为了方便Python开发者的使用。如果使用Python工作，并想创建优秀的用户界面，那么这本教程就是为此准备的，这本教程提供了一个精简版的Delphi快速学习教程。

同时，本教程对那些想了解Python开发的Delphi开发人员也很有用，我们希望本教程以及 DelphiVCL 和 DelphiFMX 库能够帮助你为你的Python项目创造出惊人的图形用户界面。

这本教程的重点是用Python GUI编程，但也讨论了 Delphi作为编程语言的历史。没有必要选择一个而放弃另一个。这两者都是很优秀的编程语言。而且，在程序员的工具箱中，有很多可以容纳编程语言的空间，可以选择使用合适的工具来完成工作。因此，本教程尝试为读者探讨一下解锁这两个伟大工具的可能性。

第1章 Python与Delphi

Python是一种语法简单，易于学习，功能强大的编程语言，而GUI图形用户界面开发是每一个Python程序员必须掌握的非常重要的技能之一。

Delphi作为一个快速可视化跨平台原生本机开发工具和编程语言，是使用Python

开发GUI图形用户界面最前沿、最高效、最有发展前途的技术，使用Delphi的VCL和FMX组件框架可以5倍的开发效率为Python开发跨平台支持Windows, Android, iOS, macOS, Linux的App应用。

Python for Delphi (P4D)功能：

- 对 Python API 的低级访问；
- 与 Python 的高级双向交互；
- 使用 Delphi 自定义变量 (VarPyth.pas) 访问 Python 对象；
- 使用 RTTI (WrapDelphi.pas) 包装 Delphi 对象以在 python 脚本中使用；
- 使用 Delphi 类和函数创建 python 扩展模块；

Python for Delphi (P4D) 是一套用于Python开发的免费的开源组件 (<https://github.com/pyscripter/python4delphi>)，将Python DLL包装到Delphi中。它们可以轻松地执行Python脚本，创建新的Python模块和新的Python类型。可以创建Python扩展为DLLs，以及更多。P4D 使得使用 python 作为 Delphi 应用程序的脚本语言变得非常轻松简单。

什么是Delphi4Python?

Delphi for Python的主要重点是为用户提供免费的Python模块或Delphi的GUI框架的绑定。基于Delphi的两个不同的框架两个不同的库。DelphiVCL，用于Windows 的本地库；DelphiFMX，用于跨平台开发。本教程将帮助读者学习开始使用它们中的两个。

 <h2 style="margin: 0;">Delphi for Python</h2> 	
<h3>Delphi VCL for Python</h3> <ul style="list-style-type: none"> ▪ Windows 32-bit and 64-bit only ▪ Windows 8.1 through Windows 11 <ul style="list-style-type: none"> ▪ Earlier versions may work but are not supported ▪ Based on native Windows components ▪ Includes Windows Handles, Messages, Accessibility, etc. ▪ Styling system 	<h3>Delphi FMX for Python</h3> <ul style="list-style-type: none"> ▪ Uses GPU for custom rendering ▪ Multi-platform for Windows, Linux, Android, and Mac OS ▪ Higher level of abstraction ▪ Platform services simplify behaviors ▪ Styling system
	

Delphi for Python的主要是为Python开发者提供免费的Python模块或Delphi的GUI

库的绑定，让广大的Python程序员有一个优秀的GUI框架。通过Delphi for Python让你有能力把 Delphi 的 VCL 和 FireMonkey 的强大、成熟的 GUI 框架用于 Python。它的基础是 Python for Delphi 库，与流行的PyScripter Python IDE的技术相同。所以 Python for Delphi是一座连接Python和Delphi的双向桥梁。它是同时使用两者的一个最佳框架。

有相当多的Python开发者对漂亮的GUI或对Delphi感到好奇。所以这个教程对Delphi和Python开发者都有帮助。

1.1 Python语言介绍

1.1.1 了解Python

一个程序员要成长为一名高级开发人员，在整个职业生涯的旅途中需要了解并掌握多种编程语言和技能。当然，在工作中每个人都可能有一、两种是最有效率、最拿手的主力编程语言，但有时开发人员也会在特定的事情上使用其他编程语言，无论是公司要求还是个人学习中，每一个程序员不可能生活在只有一种编程语言的世界中。



现实情况是，软件公司都不得不使用多种编程语言，最起码程序员可能需要阅读并试图理解其他编程语言写的SDK或者类库或者书籍。这意味着应该为正确的工作任务选择正确的编程工具，这样开发人员就不会试图用扳手钉钉子或用锤子转动螺栓。

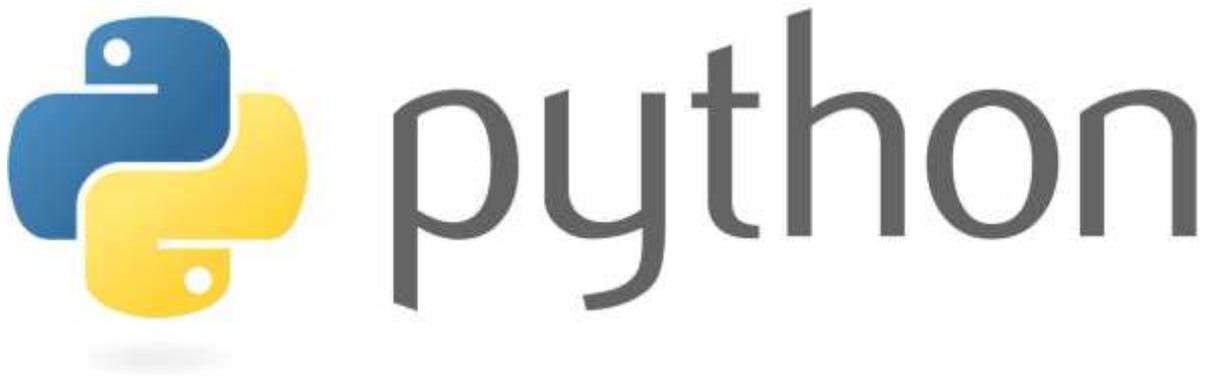


图 Python

Python是一种非常流行且适应性强的通用程序语言，深受使用大数据、云计算、人工智能(AI)和机器学习(ML)的数据科学家和开发人员的欢迎。最近几年，Python已经成为TIOBE编程语言排行榜最佳的初学者编程语言。

Python易于设置、学习曲线低和入门成本低可能有助于激发人们对写代码和编程的兴趣。Python 语言之所以自问世以来一直在程序员中越来越受欢迎，原因与学习者喜欢它的原因相同，但也因为它全面的多功能性。大量的功能加上较低的进入门槛使开发人员能够快速跟上步伐并在市场上竞争。

Python 很灵活，可以使用多种编程范式。为了真正具有竞争力，开发人员应该熟悉各种 Python 工具、库和 IDE，以便充分利用 Python 并提高他们的技能。

Python的显著特点是：

- 语法简洁；
- 代码易于阅读；
- 自动处理内存泄漏问题；
- 容易学习；
- 使用简单；

May 2022	May 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	12.74%	+0.86%
2	1	▼	 C	11.59%	-1.80%
3	3		 Java	10.99%	-0.74%
4	4		 C++	8.63%	+1.01%
5	5		 C#	6.39%	+1.98%
6	6		 Visual Basic	5.86%	+1.85%
7	7		 JavaScript	2.12%	-0.33%
8	8		 Assembly language	1.92%	-0.51%
9	10	▲	 SQL	1.87%	+0.16%
10	9	▼	 PHP	1.52%	-0.34%
11	17	▲	 Delphi/Object Pascal	1.42%	+0.22%

图 1-1 2022年6月 TIOBE编程语言排行榜

1.1.2 Python的应用领域

Python 作为一种功能强大的编程语言，因其简单易学而受到很多开发者的青睐。Python 的应用领域非常广泛，几乎所有大中型互联网企业都在使用 Python 完成各种各样的任务，例如国外的 Google、Youtube、Dropbox，国内的百度、新浪、搜狐、腾讯、阿里、网易、淘宝、知乎、豆瓣、汽车之家、美团等等，同时在人工智能领域 Python 也被广泛应用，例如最近大火的 ChatGPT。

1.2 Delphi语言介绍

1.2.1 Delphi的历史

1970年，瑞士科学家Nicklaus Wirth教授在ALGOL 68的基础上或受其影响创造了 Pascal 程序语言。它被设计用来教授良好的编程实践、结构化编程等。

1983年，Anders Hejlsberg 创造了 Turbo Pascal，它最初被称为蓝标 Pascal 操作系统。

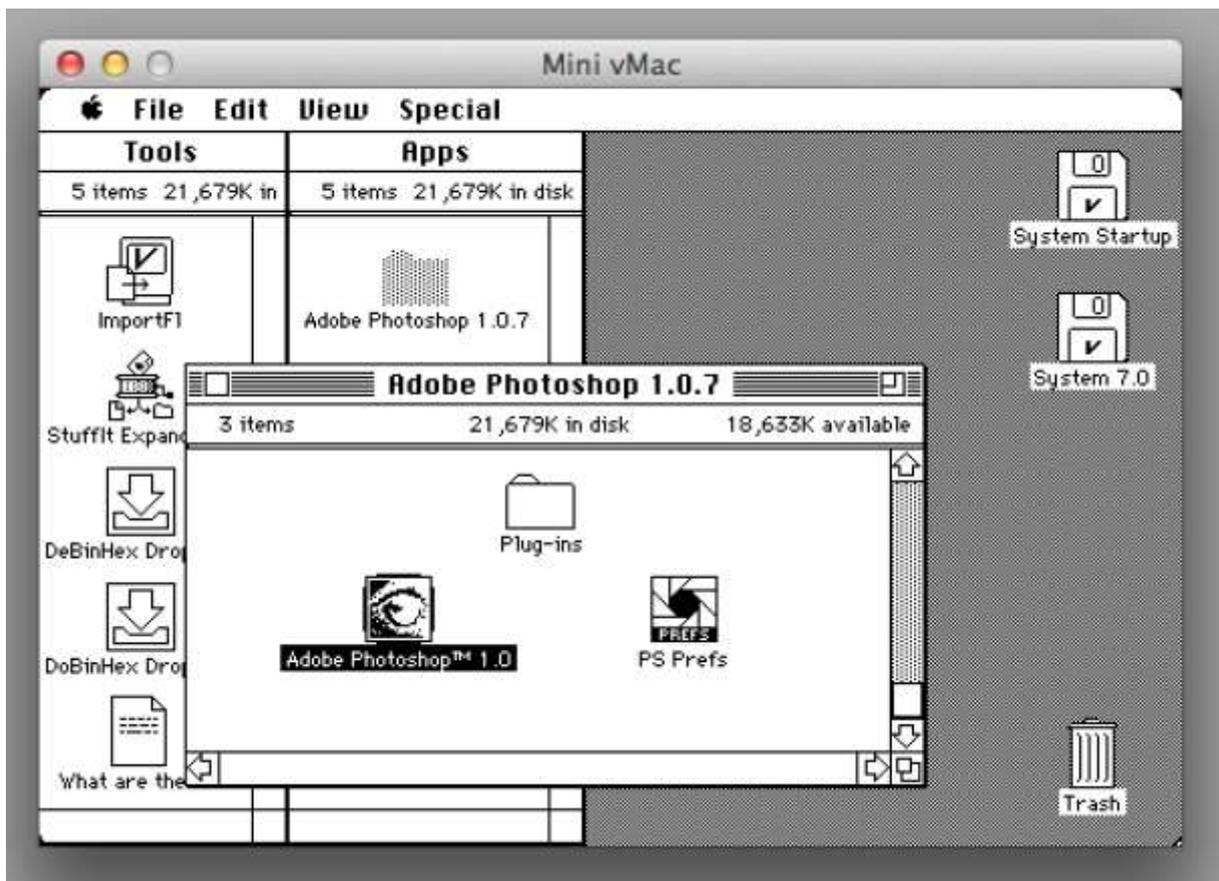


图 使用Pascal开发的PhotoShop 1.0版

Borland 软件公司收购了它，雇佣了Hejlsberg，并推广了Pascal语言。几个Pascal变体从最初的Pascal演变而来，例如：苹果公司的Think Pascal，并且使用Think Pascal开发了最早版本的苹果电脑操作系统。

后来在1987年，ABC编程语言诞生了。ABC是一种命令式的通用编程语言，Guido van Rossum使用ABC工作过一段时间。1991年，他开始创建Python，目的是使其成为ABC的继承者。1995年，Delphi，一种Object Pascal的扩展，作为Turbo Pascal的继承者被推出。Delphi的第一个版本包括一个图形用户界面（GUI）框架VCL和一个IDE开发环境。

1995年有许多的编程语言问世。事实上，Delphi和Python是差不多同一时代的编程语言。后来，Python 2.0和Python 3.0相继发布，进一步推动了Python语言的发展。而后，在2011年，Delphi引入了对Mac OS的支持，并开发了FireMonkey GUI框架，实现了跨平台开发。在2013年，Delphi发布了ARM编译器，从而增加了对iOS和Android的支持。

2017年，Embarcadero发布了Delphi Tokyo，带来了对Linux的支持。然后，2018年是Guido van Rossum退出的时候，他作为Python的仁慈的独裁者永久休假了，再后来微软收编了他，相信在微软的加持下，Python会有更大的发展空间。这本教程将学习

Delphi for Python。

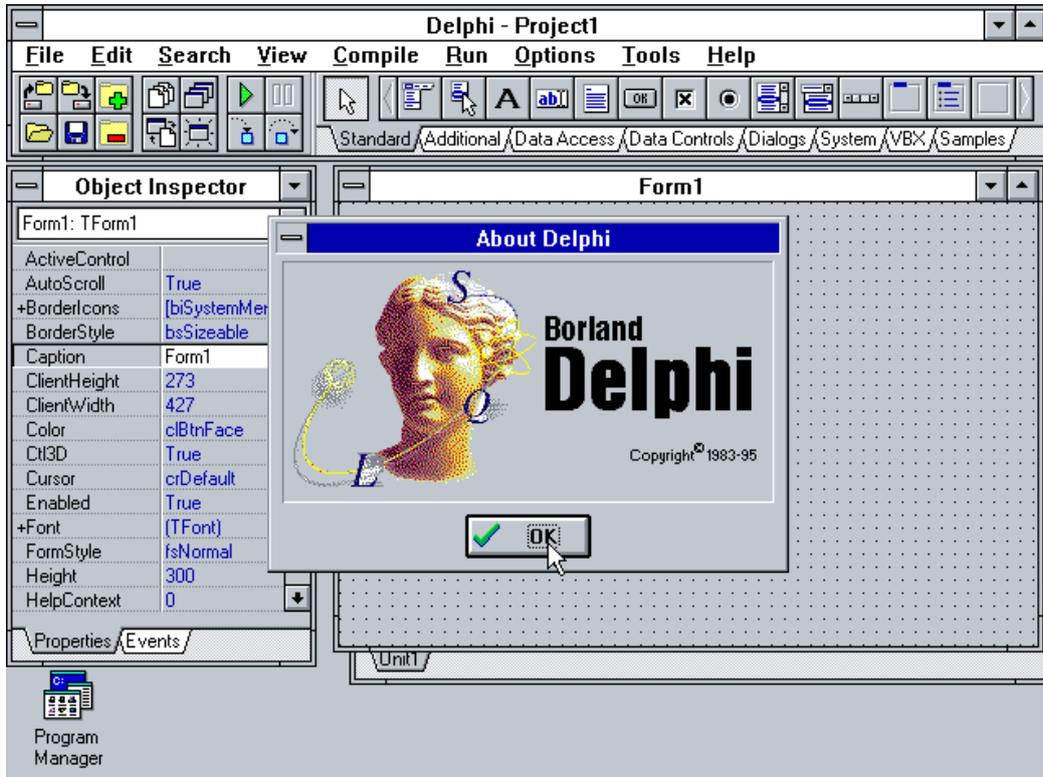
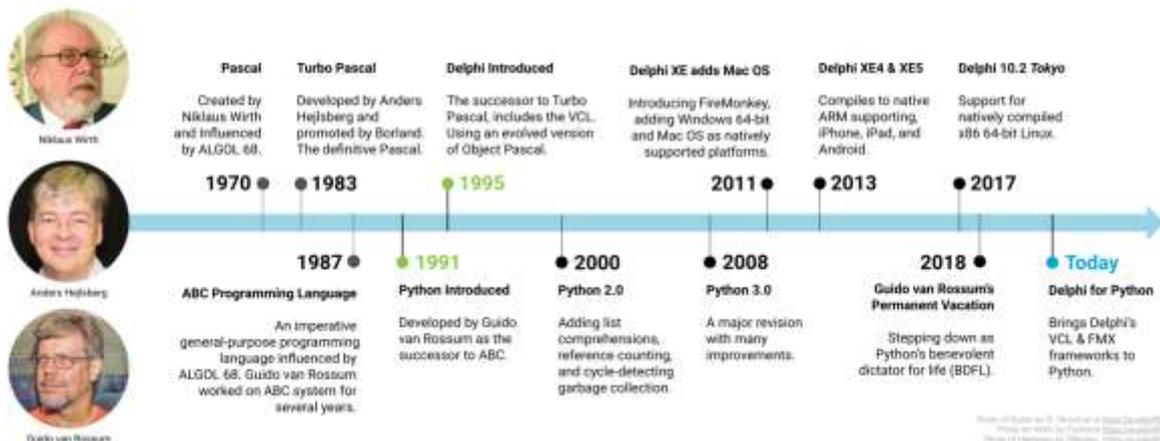


图 Delphi 1.0版

A Timeline for Delphi and Python



Delphi是美国EMBARCADERO公司的旗舰开发工具产品。作为 Object Pascal 语言的专有编程工具，Delphi 具有具有低代码“拖放”组件的图形应用程序开发、适用于大多数移动平台的所见即所得UI设计器，以及强大的Style样式选项，包括平台标准和独特的调色板，可提供完全自定义的外观和感觉。除其他功能外，包含的库还提供 GUI 控件、数据库访问管理器以及对目标平台硬件和平台操作系统的直接访问。Delphi FireMonkey (FMX) 框架将项目编译为32位和64位Windows, macOS, Android, iOS和

Linux的本机代码，允许用户开发和维护一个代码库，分发到大多数应用市场。Delphi已经被稳定支持超过28年。

Delphi是软件行业的传奇编程语言之一，该产品于1995年2月14日情人节在美国San Francisco（旧金山）正式发布，她是软件开发历史的基石。今天随着IT科技的飞速发展，各种新平台和框架不断涌现，各种新编程语言也在不断推出，知名的TIOBE编程语言排行榜总计有多达数百种编程语言，让初学者眼花缭乱，但是Delphi因其作为跨平台本机原生开发工具的可靠性和有效性而在IT界发展趋势来来去去的潮流中站稳了脚跟，在编程语言的世界中拥有自己的一席之地。

Delphi是一种功能强大且易于学习的面向对象编程语言。它具有快速应用程序开发（RAD）范式，可用于创建从数据库解决方案到移动应用程序的任何软件。Delphi以其应用程序的易于部署而闻名 - 不需要任何额外的运行时或依赖项，以及其健壮性和单一源代码的跨平台功能。因此，Delphi被广泛认为是为Windows，macOS，iOS，Android和Linux创建，编译，打包和部署跨平台本机应用程序的最有效方法。

笔者对Delphi的总结是：Delphi具有C/C++的强大能力，能够完成任何编程任务，但却比C/C++更易于学习和使用！换句话说Delphi的开发效率比Java、C/C++更高！使用Delphi开发应用软件能够帮助公司和个人节约更多的时间和金钱。

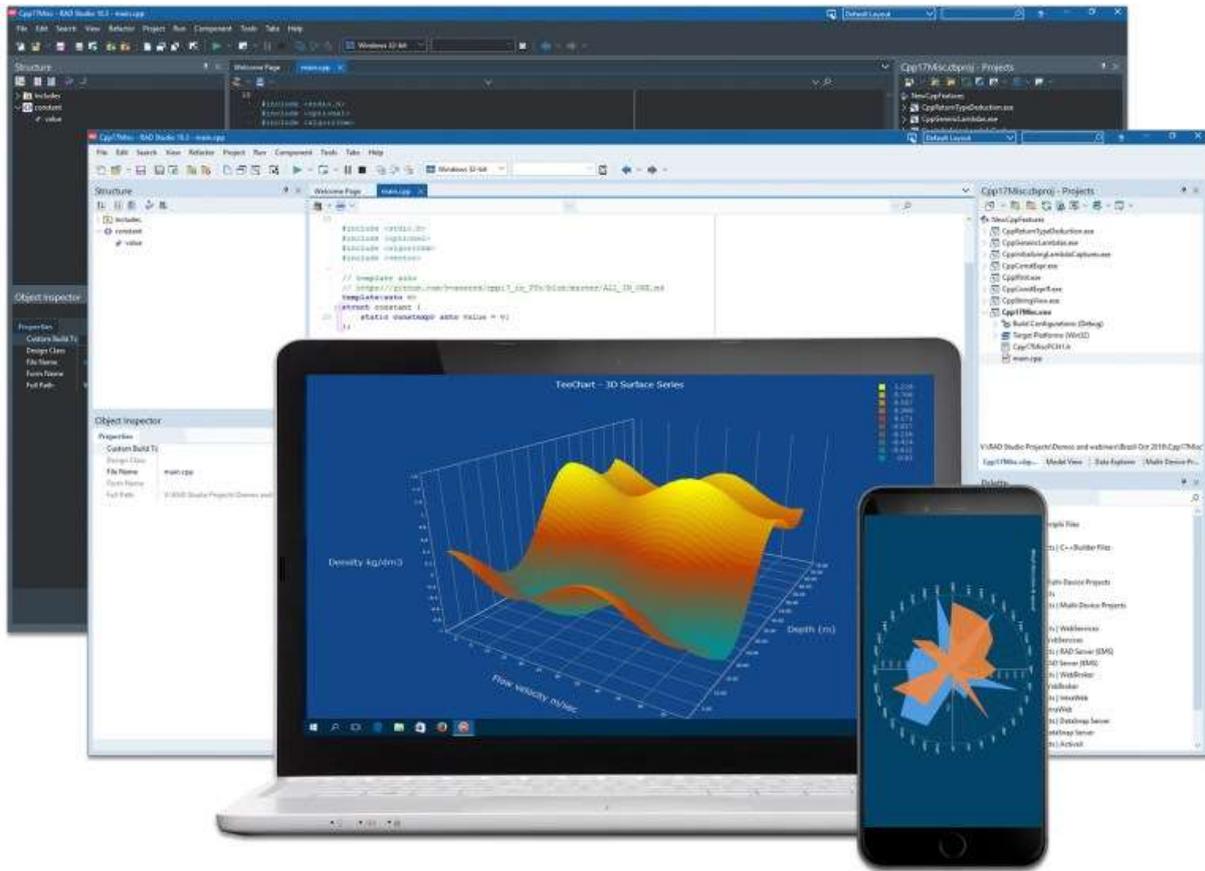


图 本机原生跨平台开发工具Delphi

从Delphi 1.0发布至今的28年中，对于整整一代软件开发者来说，Delphi给全世界数以百万计的程序员提供了编程工具和赖以养家糊口的工作技能，他（她）们使用Delphi构建了数以百万计的应用软件，以及被数以百万计的用户使用的软件。有史以来使用最多的Delphi应用程序可能是Windows桌面版Skype或者是中国著名的网络游戏《热血传奇》，世界各地的众多开发人员依靠Delphi在业界获得了广受赞誉的名声甚至实现了个人财务自由。

同时Delphi也被用来开发价值数百万美元的企业软件和在金融市场上管理千百万美元的应用软件。现在Delphi至今仍被用来开发维护这些应用软件，并在工业、自动化、银行、金融、医疗、旅游和体育领域（包括其他领域）继续创建新的应用软件.....包括为用于生日派对的移动应用软件，或者用来管理运营一个小型机场的软件系统。

2023年最新发布的Delphi 11.3可以让开发人员以5倍速度只写一套代码，即可为Windows, Android, iOS, macOS和Linux分别开发编译构建原生本机应用软件，甚至也能用于开发树莓派Raspberry Pi OS应用和WEB应用。



图 Delphi 11 Alexandria

更重要的是，随着时间的推移，Delphi不仅没有被历史抛弃，它还随着当代软件解决方案日益复杂和客户开发需求的日益增长而不断发展和扩展其功能、特性和库。

自Delphi 1发布以来的28年里，Delphi一直与快速发展的IT行业保持着高度、紧密的联系，今天Delphi 11.3更提供了一个独特且极具吸引力的主张，它结合了多平台本机功能、低代码、高生产力开发以及与其他编程语言的集成（通过C++ Builder使用C++，通过 Python4Delphi 使用Python）。

今天，从北美大陆到斯堪的纳维亚半岛、从欧洲大陆到希腊半岛爱琴海、从非洲草原到遥远的东方大国，众多世界领先的公司包括财富100强公司以及大型医疗机构、国防部门、地质勘察、船舶制造、金融证券、核工业、航天工业等关键业务依赖使用Delphi 开发的应用软件，如：索尼、福特、ABB、西门子、软银、3M、佳能、摩托罗拉、诺基亚、丰田、De11、松下、现代、UPS、美国航空航天局等，笔者曾经为中国某省的核电厂开发过上位机控制软件，该央企客户指定必须用Delphi开发。在中国台湾新竹科技园，众多高科技公司使用Delphi开发智能设备软件，相对于Java、C#、QT，Delphi开发的软件更容易部署分发，运行更稳定，有效避免了其他编程语言的DLL地狱依赖。



图 Delphi 11

VCL 和 FireMonkey (FMX) 是成熟稳定的跨平台 GUI 框架。VCL 专注于原生 Windows 开发，在 Windows 电脑中至少有一个应用软件是用 Delphi 和 Visual Component Library 构建的。例如，KMPlayer、WinRAR、MySQL 管理工具、AIMP、BurnAware、Dev-C++、EarMaster、FL Studio、Nero Burning Rom、Ultra ISO、WinRAR 以及更多应用软件。而 FireMonkey 为 Windows、Linux、macOS 甚至 Android 带来了强大灵活的 GUI 框架。通过 Embarcadero 新发布的开源免费 Python 框架可以将 Delphi 的 GUI 框架的强大功能和灵活性带入 Python 世界。

本教程提供了一个快速精简的 Delphi 初学者教程，目标是描述 Delphi 开发生态系统和该语言的基本语法。对于从另一种编程语言第一次使用 Delphi 的人来说，这个教程可能会帮助你对 Delphi 快速入门。

Delphi 其实很容易学习——编程语言的语法可以在几个小时内学会，因为这门语言开发的初衷就是帮助计算机专业的大学生学习电脑编程的。Delphi 的 Object Pascal 语言是由经典的 Pascal 语言演变进化而来的，长期以来一直是大学和教育中非常流行的编程语言，用于教授软件开发的基本概念。了解编程的基础知识是编程的关键部分。但是存在一个完整的术语和技术生态系统。对于来自不同技术背景的人来说，这总是让人不知所措。

另一个问题是，有太多不同的 Delphi 组件、示例和库可用，Delphi 拥有一个庞大

的第三方专业组件厂商生态系统，提供了各行各业各种用途的组件，使用其他编程语言的程序员可能不太相信这一点，你可以访问著名的Delphi开发库网站（<https://torry.net/>），这个网站提供的组件列表会让第一次接触Delphi的人大吃一惊，就像阿里巴巴第一次进入藏有珍宝的洞府一样，总共有大约5000个组件开发商提供了超过10000多个免费或者商业组件包，涵盖了计算机编程的各个方面，无所不包。

1.2.2 如何开始学习Delphi开发

Delphi 是美国Embarcadero（原Borland）公司开发的低代码可视化软件开发工具，是基于Windows的面向对象的跨平台可视化软件开发系统，作为企业应用开发的始创通用开发框架，具有简单、高效、功能强大的特点，是用于创建跨平台、本机编译应用的终极 IDE。Delphi是用于构建单源码多平台本机应用的创始 IDE开发工具，具有强大的低代码可视化设计功能和一流的 Windows 集成，可将工作效率提高多达 5 倍，Delphi程序员只需要开发一个代码库，即可同时支持Windows, Android, iOS, macOS, 和 Linux所有平台，借助第三方厂商组件，Delphi也可以快速开发网页WEB应用。

从1995年2月14日情人节发布Delphi 1.0版开始，多年以来Delphi一直是程序员热爱的编程开发工具，受到全球超过 300 万开发人员的信赖，今天全球数以百万计的开发人员使用Delphi来帮助构建一个更美好的世界。感谢所有Delphi研发团队开发人员、世界各地的Delphi社区和Delphi MVP以及Delphi生态链的第三方组件开发厂商。所有这些热爱Pascal语系的人们让Delphi变得伟大！

1.2.3 如何安装Delphi社区版

Delphi Community Edition是一款出色的工具，用于为Windows, macOS, iOS和Android创建视觉上吸引人的高性能本机Delphi应用程序。要安装此 IDE，请访问：<https://www.embarcadero.com/products/delphi/start-for-free>。然后，填写表格，将开始下载社区版或者试用版。

Delphi社区版是免费的，因此所需的产品密钥将通过电子邮件发送给您。因此，您所要做的就是解压缩并安装 IDE。

首先要做的是从www.embarcadero.com下载 Delphi 软件。对于学生或者创业者可以找到对所有人免费的最新 Delphi 10.4.2 社区版。请务必安装带有示例和帮助文件的 Delphi IDE，这些示例和帮助文件对初学者和中高级开发人员都有帮助。

如果嫌下载麻烦，购买本教程的开发者可以添加客服QQ：540375526 客服人员会直接将Delphi安装包发给你，并提供安装指导文档。



1.3 Delphi与Python之道

1.3.1 Delphi神谕与Python禅语

Delphi被创建的基本考虑之一是释放和增强开发人员的生产力开发效率。为了帮助程序员实现最高的开发效率，它提供了一个所见即所得的可视化UI设计器，可视化拖放组件创建图形用户、界面数量庞大的视觉和非视觉组件，以及与这些组件相关的属性和事件完美兼容旧版本的Delphi项目

丰富的第三方组件生态系统，在著名的<https://torry.net/>网站，你可以找到解决任何编程任务的组件或类库，无论是视频游戏、工控、地理信息系统、CAD图形等。开发人员可以使用这些组件快速开发出功能强大的应用程序。

Delphi神谕与Python禅语，影响了Python编程语言的设计。让我们来看看其中的几条。

1.3.2 Delphi神谕

1. Developer productivity - Really the main goal is getting things done

- quickly（开发人员生产力——真正重要的目标是快速完成工作任务）；
2. Maintainability - Code is easy to read and understand with good encapsulation（可维护性——代码应该易于阅读和理解，具有良好的封装性）；
 3. Fast compiled native apps - Compiles fast, and native applications run fast（快速编译的原生应用程序——编译速度快，运行速度快）；
 4. Database access - Always includes a rich set of database access components（数据库开发——数据库组件支持世界上的任何一种数据库）；
 5. Platform API access - You don't need to call platform APIs, but can if you want（操作系统API - 通常您不需要直接调用操作系统API，但如果您想调用，当然可以）；
 6. Property-Method-Event - General model for working with components（属性-方法-事件 - 领先的组件通用模型）；
 7. Visual designers - WYSIWYG with drag and drop interface（可视化设计 - 所见即所得的拖放式图形用户界面设计）；
 8. Reliable applications - Exception handling and component owner model（稳定可靠的应用程序——完善的异常处理和组件所有者模型）；
 9. Backwards compatibility - Even with all the updates most code is compatible（代码向后兼容性——即使版本升级，大多数代码也是兼容的）；
 10. Rich component ecosystem - There is usually a component for everything（丰富的第三方组件生态系统——无论任何开发任务都能找到一个组件可以满足开发需求）；

1.3.3 Python禅语

影响 Python 编程语言设计的 19 条“指导原则”

1. Beautiful is better than ugly（美比丑好）；
2. Explicit is better than implicit（清晰比晦涩好）；
3. Simple is better than complex（简单比复杂好）；
4. Complex is better than complicated（复杂比杂乱好）；
5. Flat is better than nested（扁平比嵌套好）；
6. Sparse is better than dense（松散比密集好）；
7. Readability counts（可读性很重要）；
8. Special cases aren't special enough to break the rules（即使特殊情况也不

- 应该违反这些规则)；
9. Although practicality beats purity (但现实往往并不总是那么完美)；
 10. Errors should never pass silently (异常不应该被静默处理)；
 11. Unless explicitly silenced (除非你希望如此)；
 12. In the face of ambiguity, refuse the temptation to guess (遇到模棱两可的地方，不要胡乱猜测)；
 13. There should be one-- and preferably only one --obvious way to do it (任何事情都有也肯定会找到一种通常也是唯一的一种最佳解决方法)；
 14. Although that way may not be obvious at first unless you're Dutch (虽然这种方法并不是显而易见的，因为你不是那个荷兰人[这里指的是Python之父 Guido]所以你需要全力去寻找)；
 15. Now is better than never (立刻行动比迟迟不动手好)；
 16. Although never is often better than *right* now (不行动比盲目行动好[极限编程中的YAGNI原则])；
 17. If the implementation is hard to explain, it's a bad idea (如果一个实现方案难于理解，它就一定不是一个好的方案)；
 18. If the implementation is easy to explain, it may be a good idea (如果一个实现方案易于理解，它很有可能是一个好的方案)；
 19. Namespaces are one honking great idea -- let's do more of those! (命名空间非常有用，我们应当多加利用)；

所以在很多方面，Delphi和Python有着相同的开发理念，Python禅语也同时适合于Delphi。

1.3.4 Delphi和Python之间的区别

一些关键的区别是这样的。Delphi虽然是商业软件（商业软件对开发者和客户不是坏事儿，实际上有商业公司支撑才能不断的更好发展，很多开源软件慢慢死掉，就是因为开发人员无法获得稳定的收入，从而导致开源项目无人维护而被废弃），但Delphi有一个免费的社区版可以用来免费开发商业或者开源项目，而Python是完全开源的（现在获得了微软的支持）。

Python有带引用计数和循环检测的垃圾收集。Delphi采用手动内存管理，但它有一个所有权模型，对于大多数情况来说，它简化了内存管理。

Delphi是一种编译语言，尽管有一些第三方的解释型方案。相比之下，Python主要是解释型的，也有一些第三方工具可以编译它。

人们为商业目的、企业对企业类型的软件或内部IT开发使用Delphi构建。虽然Python已经发展成为一种通用语言，但它在研究和机器学习原型设计中很流行。

1.3.5 Delphi和Python的开发层级

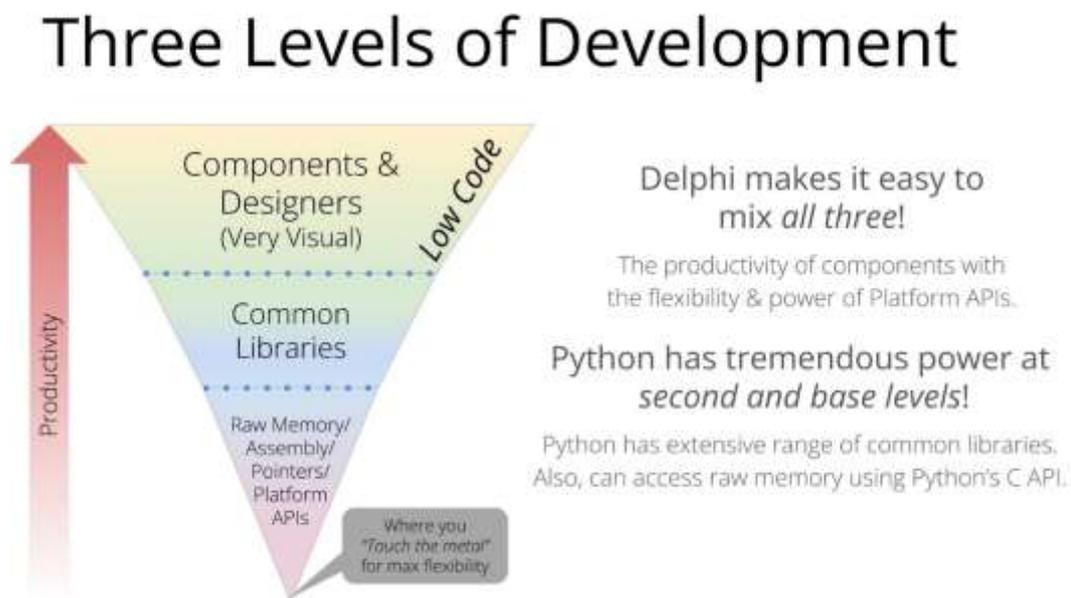


图 Delphi开发的三个层级

使用Delphi或一般的开发，都属于这三类中的一类。

- 低代码，使用视觉组件和设计师；
- 实现特定功能的标准和第三方组件库；
- 使用指针或汇编代码的原始内存访问；

Delphi还有许多低代码工具，是为那些不想学习编程，只想完成一些工作中的特定任务的人设计的。这些工具非常直观的，容易使用，只需要写很少的代码，Delphi的重点是生产力。在这种低代码水平上，你能做的事情是有限制的。在第二个层次，我们使用普通的库并编写程序。与第一层相比，我们在这里可以定制更多的具体功能。然后，你有最低级别，这是最灵活和最强大的，但也是生产力最低的。在这里，你要通过编写汇编语言代码来处理原始内存和指针。你在用古老的平台和API直接与CPU对话。

Delphi在结合所有三个层级的开发方面做得很好。

Python主要属于中间层级，主要调用大量的内置和第三方库。Python有自动垃圾收集器，而Delphi则是需要程序员手动处理和释放内存垃圾。我们可以用C语言扩展

Python。许多第三方Python模块是用C或C++编写的，这给你提供了原始的内存管理，并将Python的范围扩展到低级别的平台访问。

Python提供了在C语言中创建扩展模块并提供Python绑定的功能，使其成为使用最多的语言。此外，我们可以将这些扩展模块作为库导入，使Python具有难以置信的生产力。这是我们的主要关注领域，在这里我们也为Python创建了用Delphi编写的扩展模块。

1.3.6 Python运行Delphi VCL和Delphi FireMonkey框架

Delphi提供了两个用于图形用户界面（GUI）应用开发的库或框架：

- Visual Component Library（VCL）框架；
- FireMonkey（FMX）框架；

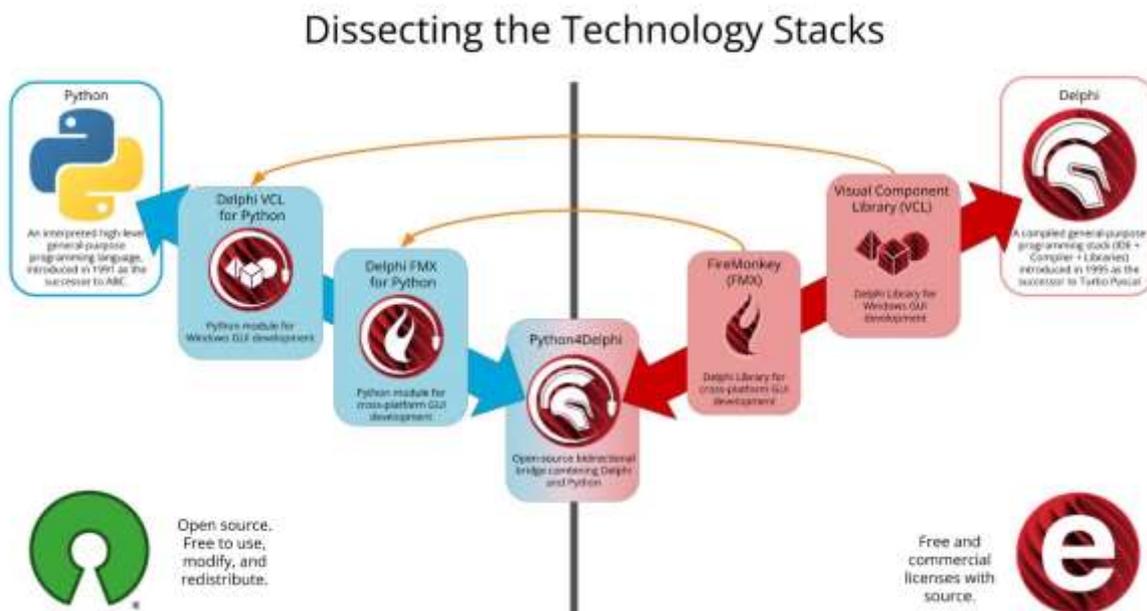


图 剖析Delphi&Python技术栈

VCL框架最初是在1995年与第一版Delphi一起发布的，主要目标针对Windows平台。尽管Delphi属于商业软件，虽然编译器和IDE不开源，但它从发布之日起就附带了全部VCL库源代码，这对它日后被广泛使用产生了巨大的推动力，可以说Delphi是首个开放源代码的商业框架，这使得广大的程序员可以通过阅读世界一流软件公司的商业框架源代码提高自己的开发技能，并由此开启了一个新时代。

Delphi是采用组件模型创建的，以便于重复利用组件，减少开发成本。Delphi让所有的开发者都能接触到VCL框架源代码，以了解Delphi是如何设计编写的，并参考创建他们的自定义组件。所以Delphi创造了第一个由广大第三方厂商组成的丰富的组件生态系统供开发者使用。

VCL是基于Windows组件的，所有的组件都有句柄来拦截消息。VCL框架的一个重要部分封装了标准的Windows组件。因此，你可以自动访问Windows组件所带来的一切，但它增加了一个薄薄的抽象层。VCL极大的简化了Windows API的使用，你不需要考虑句柄、消息或其他东西，就能很容易的编写应用程序，当然Delphi并不限制程序员的能力，你可以自由的调用操作系统提供的底层API，这一点大大领先于同时代的VB和PB，例如：在美国一家典型的软件公司里，有专职的Delphi程序员负责用Delphi开发ActiveX组件以提供给公司的Visual Basic程序员使用，这种强大的开发能力使得Delphi在90年代风靡世界。



图 使用Delphi VCL框架开发的即时通讯软件Skype桌面版

FireMonkey框架于2011年发布，你可以在上面的时间轴上看到至今已有10年的历史，这使得FMX框架经受住了开发者和时间的考验。FireMonkey从一开始就被设计成一个跨平台的GUI框架。它充分利用了GPU的硬件性能优势。因此，硬件加速在Windows上使用DirectX，在其他平台上使用OpenGL，以创建跨平台的非常快速、漂亮的UI。FMX支持Windows、Mac、iOS、Android和Linux。如果你了解VCL，你可以迅速熟悉FireMonkey，但FireMonkey并没有因为要完全向后兼容VCL而受到阻碍。它包括平台服务，这有助于跨平台的抽象化。因此，行为和外观会自动适应你所运行的平台。

下面这张图是法国一家公司用Delphi开发的类似抖音的移动App，使用一套Delphi FMX代码同时开发了iOS和Android应用。

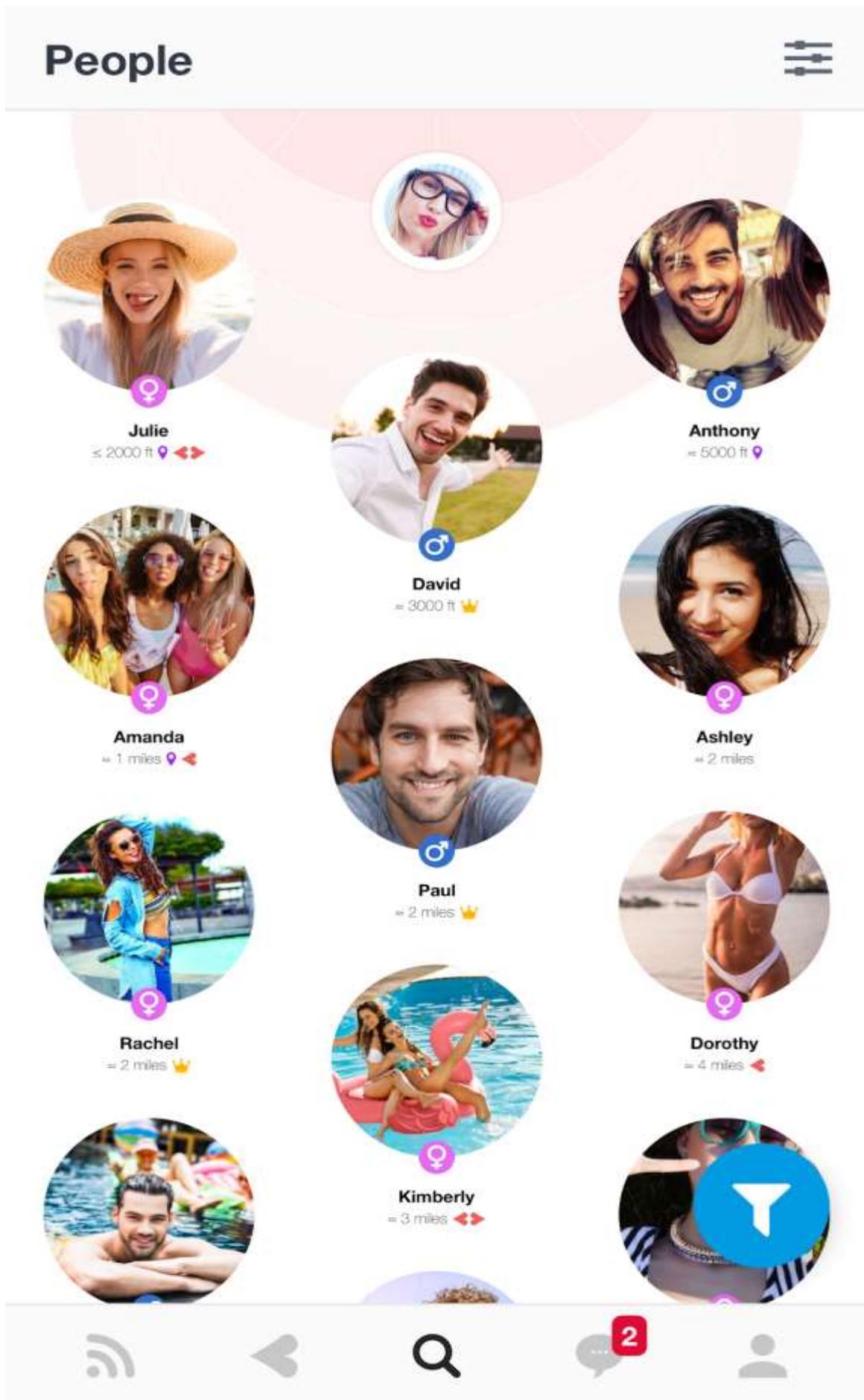


图 法国一家公司使用Delphi FMX开发的社交App支持ios&Android手机

新版的Delphi后端编译器采用的是LLVM，随着LLVM编译器的技术进步，Delphi的速度和C++一样快，生成的2进制代码质量更好。我们使用 Python for Delphi (<https://github.com/Embarcadero/python4delphi>) 桥接库为VCL和FireMonkey创建了Python扩展模块。所以这两个库组成了Delphi for Python。

- Delphi VCL for Python (<https://github.com/Embarcadero/DelphiVCL4Python>)
- Delphi FMX for Python (<https://github.com/Embarcadero/DelphiFMX4Python>)

1.3.7 立刻开始使用DelphiVCL for Python和DelphiFMX for Python

Delphi VCL for Python 是 Delphi 传统 GUI 框架VCL 的 Python 绑定。如前所述，它是一个用于Windows操作系统平台的GUI框架。开发人员不需要安装或了解Delphi就可以使用Delphi VCL for Python开发GUI应用程序。它以Windows为中心，并且是轻量级的，因为它使用了预先存在的Windows控件。唯一的要求是在Python中安装delphivcl (<https://github.com/Embarcadero/DelphiVCL4Python>) 包。

相比之下，Delphi FMX for Python是一个跨平台的GUI应用开发库。如果想用Python为Linux、Mac、Android或iOS开发GUI应用，需要这个 DelphiFMX

(<https://github.com/Embarcadero/DelphiFMX4Python>) 包。使用DelphiVCL构建的应用程序将比使用DelphiFMX构建的应用程序更快，因为DelphiVCL直接与Windows对话。所以在为Windows开发GUI应用程序时，使用DelphiVCL更有优势。

当学习并开始使用DelphiVCL开发时，可以在短时间内转入DelphiFMX来创建跨平台的应用程序，因为这两个框架非常相似。

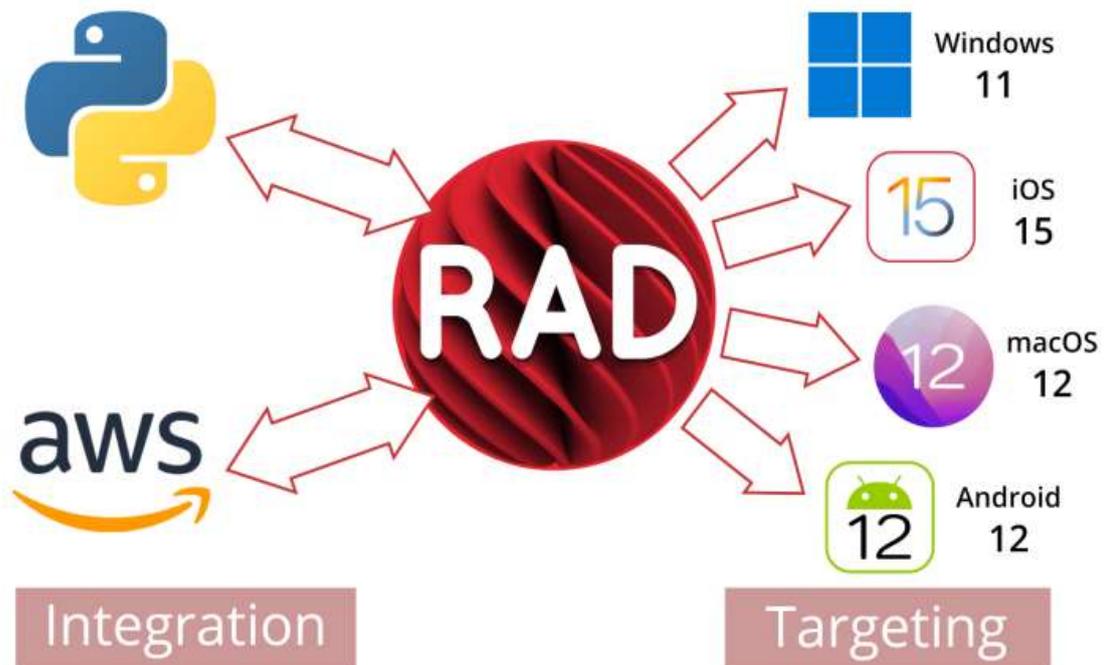


图 RAD Delphi支持跨平台和云平台全栈开发

第2章 Delphi快速入门

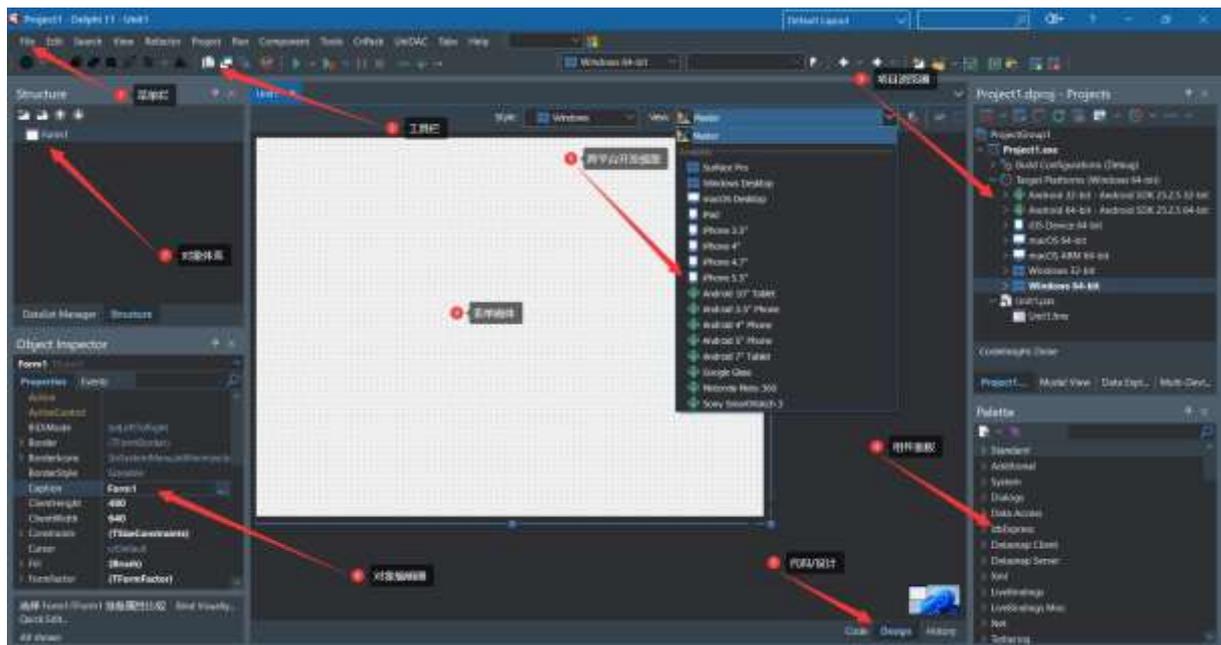
2.1 Delphi语言介绍

2.1.1 Delphi集成开发环境

Delphi是用来快速开发Windows, Android, iOS, macOS, 和 Linux的跨平台应用程序的开发工具, 它可以直观地使用图形化工具来创建Windows, Android, iOS, macOS, 和 Linux的应用程序。它使程序员从完全以代码的形式创建可视化应用程序的繁琐且复杂的工作中解脱出来, 使用户可以不用掌握太多的编程专业知识就能够以低代码方式创建出图形化的、高难度、稳定可靠的应用程序。

Delphi 的集成开发环境 (IDE) 使用窗口、菜单和应用程序来设计可视化界面, 将代码以事件的形式与界面的每一个组件建立联系, 并可以对整个应用程序进行调试。

Delphi 的窗口主要包括9部分。分别为:菜单栏、工具栏、项目浏览器、组件面板、代码/设计、对象编辑器和对象体系浏览器、表单窗体、跨平台开发视图等。下面介绍各部分的构成和功能。



1. 主窗口 (Main Form)

Delphi的主界面窗口位于屏幕的上端，主要包括Menu(菜单栏)、Speed Bar(工具栏)和Palette(组件面板)。Menu由下拉式菜单组成，包括了Delphi集成开发环境IDE的所有功能。SpeedBar位于主窗口的左下端，提供了一些常用的功能和操作选项。Delphi 提供了丰富的组件供程序员使用，这些组件和由用户自己安装的第三方厂商组件都放在Palette组件面板中，整个组件面板分成若干组(页)，可通过组标名选择组件。

2. Structure(对象体系浏览器)

对象体系浏览器显示了项目中的表单窗体及其上面所有组件对象的名字和类型以及对对象之间的组织体系附属关系，用户可以方便地在表单窗体的各个组件之间切换，也可以快速地回到表单窗体本身。当表单窗体中含有较多的对象时，就会发现切换对象时，该方法是回到表单窗体的最快捷的途径。

3. Object Inspector(对象检阅器)

对象检阅器是联系UI界面和代码的纽带，主要用来设置当前对象的初始属性和所涉及的事件处理程序。对象检阅器窗口含有两页：Properties 页(属性页)，显示窗体中当前被选择组件的属性信息，并允许改变组件对象的属性。属性设置可通过直接输入、列表选择(↓)或打开对话框(⋮)的方式选择，标记“+”表示属性含有下层属性，双击属性名称，可展开下层属性。Events(事件)页，列出了当前组件可以响应的事件，例如：点击按钮事件、双击窗体事件等。按下Object Inspector下端的“Events”页标签，使得 Events 页可见，可以指定事件后边的空白处，定义对象接受相应事件执行的动作。首次启动时，Object Inspector 窗口显示的是当前表单窗体。

4. 表单窗体 (Form)

Forms表单窗体是进行软件UI界面设计的工作区域，首次启动Delphi时显示的是表单窗体Form1。可以把菜单、按钮等组件放在表单窗体中，通过移动位置、改变长度、宽度尺寸等操作安排组件在表单窗体中的布局，来开发应用程序的用户界面。表单窗体的栅格(Grids)供放置组件时对齐位置用，在程序运行时栅格Grids是不可见的。一个应用程序可能不止有一个表单窗体，可以选用不同的表单窗体进行设计，其他表单窗体可以是对话框(Dialog Box)、数据录入框等。当创建一个新项目时，Delphi会自动产生一个主表单窗体。

5. 代码编辑器

代码编辑器一开始处于表单窗体窗口之下，用户可以随时通过代码/设计编辑器切换代码编辑器和UI界面设计器。因为在Delphi中，设计用户界面直接在表单窗体中进行，运行结果和设计样式完全一致。当组件被放到表单窗体上时，Delphi会自动生成大部分的用户界面代码。程序设计人员所应做的只是在生成的代码框架中加入完成所需功能的程序段而已。单击代码/设计编辑器切换状态行可使代码编辑器或UI界面设计器处于可见状态。

6. UI界面设计器

通过代码/设计编辑器切换到UI界面设计器窗口，该窗口用来显示用户使用的所有可视化组件，对于管理程序、明确各组件间的关系有很大帮助。调整程序运行时，程序窗体在屏幕中的初始位置，用户用鼠标选择表单窗体上的组件，可以拖动表单上的组件，就设置好了运行时此组件的位置。

2.1.2 Delphi项目的主要文件

在开发应用程序时，要使用项目来管理构成应用程序的所有不同的文件。一个项目包括：

(1) 项目文件 (.dpr)

.dpr文件是在新建一个应用的时候自动创建的，为主程序文件。它是主表单窗体以及其他自动创建的窗体实例化的地方，一般不需要编辑项目文件，除非要执行程序初始化例程、显示启动画面或执行其他必须在程序启动时执行的例程。下面代码是一个典型的项目文件。

通过菜单Projiec->View Source可以看到以下代码：

```
program Project1;
```

```

uses
  Vcl.Forms, //引用依赖的系统库文件, 由 Delphi 自动生成
  //表单窗体 Form1 的代码文件, 由 Delphi 自动生成
  Unit1 in 'Unit1.pas' {Form1};

{$R *.res} //资源文件, 由 Delphi 自动生成

begin //开始, 类似 C/C++ 中的花括号 {
  Application.Initialize; //应用初始化
  //控制在 VCL 应用程序中如何处理 Windows 任务栏按钮。
  Application.MainFormOnTaskbar := True;
  Application.CreateForm(TForm1, Form1); //创建表单窗体类的实例 Form1
  Application.Run; //应用启动
end. //结束, 类似 C/C++ 中的花括号}

```

传统 Turbo Pascal 程序员会把项目文件看作是标准的 Pascal 源程序。{\$R*.res} 用于引用项目的资源文件(扩展名为 res)。

begin...end 之间的语句是应用程序的执行代码, 主要用到了 Application 的三个方法, Initialize 方法是初始化; 控制在 VCL 应用程序中如何处理 Windows 任务栏按钮。如果 MainFormOnTaskBar 为 True, 任务栏按钮代表应用程序的主窗体并显示其标题。如果是 False, 任务栏按钮代表应用程序的(隐藏的)主窗口, 并显示应用程序的标题。MainFormOnTaskBar 必须为 True, 才能使用 Windows Vista 或 Windows 7 Aero 效果, 包括实时任务栏缩略图、动态窗口、Windows Flip 和 Windows Flip 3D。如果需要改变默认行为, MainFormOnTaskBar 应该在 Application.Initialize 之后和主窗体创建之前在项目源文件中设置。MainFormOnTaskBar 设置的目的是在应用程序的生命周期内持续存在; 以后运行时对该属性的改变可能会导致意外的行为。

对于在 Delphi 2007 及以后的产品中创建的应用程序, MainFormOnTaskBar 默认为 "True", 而对于早期产品则默认为 "False"。

MainFormOnTaskBar 属性可以应用于较早的应用程序。注意, 如果你的应用程序依赖于它, 它会影响你的 MainForm 的 Z-order。

要更新现有的 VCL 应用程序, 在项目源文件中添加以下一行(例子包括应用程序初始化行, 它必须在设置 MainFormOnTaskBar 的行之前)。

CreateForm 方法是建立 Form1 表单窗体, Run 方法是运行程序。

(2) 表单窗体文件 (.dfm)

它存储了表单窗体中所有对象的设计属性, 以文本形式保存。在表单窗体上单击右键菜单->View as Text 可以查看表单窗体的脚本如下, 除非你知道自己在干啥, 否则

(试读结束, 全文共196页, 如需全文版本, 请微信搜索关注“ZSHi战石”公众号获取)

微信扫码关注“ZSHi战石”公众号, 获取《Python图形用户界面设计——Delphi For Python高级开发教程》全本版



郑重声明

本文档为企业内部培训资料, 免费赠阅, 仅供个人学习和研究使用, 涉及的软件工具及文档资料、图片均来自互联网, 版权归相应公司所有, 如需商用请购买正版, 如有侵犯您的商标权、著作权和或其他合法权利的, 请联系我公司, 我司法务专员将依法处理, 本公司不承担任何法律责任。

济南战石电子商务有限公司