**VBA二次封装及其应用**

**摘 要**：随着数字化技术的不断发展，各行各业每天都有大量的报告需要编制，如此周而复始的劳动，如果依靠人工通过计算机编辑表格、录入数据，不仅劳动量巨大，而且容易出错。因此研究VBA二次封装技术并自动生成Word文档的设计具有重要意义。本文以面向对象的思想，在VB.NET环境中实现了VBA编程元素的二次封装，将图、表、文本等Word元素的编程操作封装构成动态链接库，极大地简化用户编写VBA应用程序的工作，减少开发工作量，提高访问速度，并通过源代码封装提供一定程度的程序安全性。论文将这一动态链接库，成功应用于学生简历生成系统。该系统提供界面化的简历标题、正文、图、表的编辑，简历模板的存储与修改，基于简历模板的快速、自动简历生成等功能。基于这一VBA动态链接库，简历生成系统的设计简单、高效，且能够灵活扩展至其他领域，充分说明这一VBA二次封装动态链接库设计的正确性与合理性。

**关键词**：VBA；二次封装；自动生成Word；简历；VB.NET

**Secondary encapsulation of VBA and its application**

**Abstract**: With the continuous development of digital technology, all walks of life have a large amount of report to prepare every day. It is such the cycle of labor. If relying on people to edit tables and input data by computer, there will be a huge amount of labor and easy to make mistakes. So the VBA secondary encapsulation technology and automatically generate Word document design is of great significance. Based on the object-oriented idea, this system implement the VBA programming element in the VB.NET environment of secondary encapsulation, it encapsulates the programming code of graph, table, text the Word element into a dynamic link library operation. It greatly simplify the work of VBA application, reduce the code workload and improve the access speed, and through the source code encapsulation to provide a certain degree security of application .The article will successfully apply this dynamic link library into the application of the student resume generation system. The system provides interfaces to manage the edit of title, text, graph, table, storage and modification of the resume template. It also quickly and automatically generate resume based on your resume template and it has other functions .Based on the dynamic link library with VBA secondary encapsulation, the design of this resume generate system is simple, efficient, flexible and able to expand to other areas, fully show the correctness and rationality of the design of dynamic link library.

**Keywords**: VBA; secondary encapsulation; automatically generate Word; resume generation; VB.NET

# 目 录

[1 绪论 1](#_Toc453147135)

[1.1 课题的意义 1](#_Toc453147136)

[1.2 国内外研究现状 1](#_Toc453147137)

[1.3 毕业设计的主要内容 2](#_Toc453147138)

[1.4 论文组织与结构 3](#_Toc453147139)

[2 开发环境 4](#_Toc453147140)

[2.1 系统开发环境搭建 4](#_Toc453147141)

[2.2 软件工具 4](#_Toc453147142)

[2.2.1 Visual studio 2015 community 4](#_Toc453147143)

[2.2.2 Office Word 、Access 4](#_Toc453147144)

[2.3　VB.NET环境 4](#_Toc453147145)

[2.4 数据库OLEDB对象 6](#_Toc453147146)

[3 系统分析 8](#_Toc453147147)

[3.1 VBA封装类的需求 8](#_Toc453147148)

[3.1.1 VBA封装的背景 8](#_Toc453147149)

[3.1.2 VBA对Word的开发与存在问题 8](#_Toc453147150)

[3.1.3 DLL的必要性 9](#_Toc453147151)

[3.2 DLL类需求 10](#_Toc453147152)

[3.3 简历生成系统框架 10](#_Toc453147153)

[3.4 简历生成系统功能流程分析 10](#_Toc453147154)

[3.5 简历生成系统功能需求 11](#_Toc453147155)

[3.6 数据库需求设计 14](#_Toc453147156)

[3.6.1 数据库表列单 14](#_Toc453147157)

[4 DLL类设计 17](#_Toc453147158)

[4.1 VBA程序封装成DLL文件 17](#_Toc453147159)

[4.1.1 关于DLL与VBA代码封装 17](#_Toc453147160)

[4.1.2 消除宿主程序的对象模型对封装DLL的影响 17](#_Toc453147161)

[4.1.3 相关类库的引用 18](#_Toc453147162)

[4.2 文本类 19](#_Toc453147163)

[4.3 标题类 19](#_Toc453147164)

[4.4 图片类 20](#_Toc453147165)

[4.5 表格类 20](#_Toc453147166)

[4.6 文档输出类 21](#_Toc453147167)

[4.7 DLL生成 21](#_Toc453147168)

[4.8 DLL调用 22](#_Toc453147169)

[5 VBA封装类应用——简历生成系统 23](#_Toc453147170)

[5.1 人机界面设计 23](#_Toc453147171)

[5.2 数据库应用 23](#_Toc453147172)

[5.2.1 数据库创建并连接 23](#_Toc453147173)

[5.2.2 数据库的应用 23](#_Toc453147174)

[5.3 系统开发文件部署 26](#_Toc453147175)

[5.4 系统模块详细设计与实现 26](#_Toc453147176)

[5.4.1 简历模板模块 26](#_Toc453147177)

[5.4.2 标题编辑框模块 27](#_Toc453147178)

[5.4.3 文本编辑框模块 29](#_Toc453147179)

[5.4.4 图片编辑框模块 30](#_Toc453147180)

[5.4.5 表格编辑框模块 31](#_Toc453147181)

[5.4 系统运行 33](#_Toc453147182)

[6 结论 36](#_Toc453147183)

[参考文献 38](#_Toc453147184)

[致 谢 39](#_Toc453147185)

# 1 绪论

## 1.1 课题的意义

随着数字化技术的不断发展，各行各业每天都有大量的报告需要编制，Office 如今使用越来越频繁，人们日常生活中已经离不开Word的使用。而在大多数情况下仅仅是其中的数据、图表不同而已，如此周而复始的劳动，如果依靠人工通过计算机编辑表格、录入数据，不仅劳动量巨大，而且容易出错。在Office中，对于一些烦琐、重复的操作，用户可以通过“宏”来实现。对需要应用开发的场合，Office操作则通过提供VBA(Visual Basic for Application)作为二次开发工具，满足不同应用需求，由此VBA成为了实现Word文档自动生成的强有力的编程工具。但是仅仅使用VBA还有不完善的地方，在一些特殊的应用中，有力所不及之处。比如：毕业论文任务书自动生成、工程设计报告模板定制设计、数学计算报告格式生成等，单纯使用VBA往往导致重复开发，大量代码重复，且所开发的程序不具备可移植性，极大地影响了开发成果的共享。因此，VBA二次开发程序快速有效开发的问题，是亟待每个程序员研究、解决的问题。

动态链接库(Dynamic--Link Library)，是基于windows程序的设计的一个非常重要的组成部分。将程序制作成为动态链接库，不仅节省内存开销，减少开发的工作量，提高访问速度，更重要的是能将程序源代码进行封装，提供程序的安全性。利用这一特点，可以将VBA代码封装到动态链接库文件中，然后在Office文件中进行调用，从而提高安全性。因此研究VBA二次封装技术并自动生成Word文档的设计具有重要意义。

随着中国高校教育的发展，大学生的数量急剧增加，大学生就业问题也日益凸显，大学生就业难问题已成为社会热点之一。因此本设计以开发简历生成系统作为VBA二次封装动态链接库的应用实例，所生成系统能够允许用户快速编辑文本、输入表格信息并上传图片，最后按照用户的要求自动生成简历报告文档，具有一定的实用价值。

## 1.2 国内外研究现状

现如今国内外也共同采取了一些方法来解决自动生成文档，但是仍有缺陷。如下即是不同方法的缺陷。

以WinCC在纺织业数据采集与监控项目为例，单纯使用VBA虽然提高了WinCC项目组态的效率，但是项目架构复杂导致很多代码重复。AutoCAD VBA二次开发在弯管模参数化设计也有重要贡献，帮助设计师们从繁杂的脑力劳动和体力劳动中解放出来，以便更多时间放在设计上，但是该程序不具备可移植性。

在报表的自动生成方面，文献[9]主要采用 VBA (Visual Basic for Application)编程。但在软件重用性与代码安全性上存在缺陷和漏洞，不能满足继承和封装的要求， 不利于软件的进一步扩展。

文献[10]主要采用Delphi来实现。Delphi与BDE(Borland Database Engine)的无缝集成，以及Delphi提供的现成数据库操作控件，使得Delphi在数据库支持上有着无可比拟的优势，但其在兼容性和稳定性等方面存在着许多的不足。

文献[11]介绍了用VC++与VBA实现复杂报表自动生成的方法。但是，在日常工作中，报告里不仅仅只有表格，还有文字、图片等其他内容，该方法无法满足这些要求。

本文采用VB.NET编程实现了文字、图表并茂的规范文档的自动生成。Word提供的强大的编程接口技术让我们能够通过各种编程工具使用Word。在VB.NET环境中，我们将Microsoft Office安装目录中的Microsoft Word 15.0 Object Library类库导入VB.NET中，利用COM技术编程，在源代码中调用OLE自动化对象来操作Word。鉴于报告文档的格式相对固定，开发人员多采用事先设计好的模板文件，在需插入内容的地方设置TreeView节点。所以，文中通过节点定位和节点单击事件跟随相结合的方法来实现特定位置的内容插入，以减少代码量，提高程序执行率。

## 1.3 毕业设计的主要内容

(1) 课题研究的主要内容：学习VBA，并对VBA的应用程序接口进行二次封装，实现Word文档标题、正文、图、表等元素的编程接口，以及Excel表格基本操作的编程接口；

要求二次封装能屏蔽VBA操作的复杂性，接口定义明确，功能划分清楚，使用方便，使用面向对象方法来完成；

依据应用要求，设计数据库表存储报告模板；

设计用户界面，能对报告模板进行显示、编辑、维护；

利用二次封装接口，自动生成基于符合存储模板格式的Word文档。

(2) 目标：应用于学生生成简历，使其拥有一个可操作的平台，在该平台上编辑Word文档标题、正文、图、表，提高编辑简历的效率。当然，如果稍加改动，还会适合其他行业使用。

(3) 要求：实现学生简历生成系统。具体包括以下功能：

* 根据简历需求，提供必要功能，如：默认简历模板、文本编辑框、标题编辑框、图片编辑框以及表格编辑框；
* 提供强大的查询功能，不同用户可以根据需求，以不同的文本、标题、表格为目标进行修改；
* 效率高，界面友好，操作方便；
* 代码要求重用和接口编程，便于维护，稳定可靠；
* 系统要求部署简单，性能良好。

## 1.4 论文组织与结构

第1章绪论，首先讲解本课题的意义，从国内外研究现状分析VBA二次封装开发的必要性。同时也叙述了本毕业设计的主要内容以及目标，论文结构。

第2章开发环境，从系统开发环境搭建、软件工具分析，介绍VB.NET环境，同时也介绍了数据库OLEDB对象。

第3章系统分析，从VBA封装背景、开发问题分析揭示VBA封装类的必要性，从而需求哪些DLL类。然后涉及到简历生成系统，从系统框架、系统功能流程、功能模块、数据库设计中分析系统的需求。

第4章DLL设计，讲解了如何将VBA封装成DLL文件，再从office操作中分成五个类：文本类、标题类、图片类、表格类、文档输出类。最后讲解如何生成DLL和调用DLL。

第5章VBA封装类应用——简历生成系统，这是界面功能实现部分，从数据库连接应用分析，如何详细设计简历模板模块、标题编辑框模块、文本编辑框模块、图片编辑框模块、表格编辑框模块。最后系统运行测试结果，并展示效果图。

第6章结论，分析应用DLL的优缺点，以及叙述在开发过程中的问题和解决方案。

最后有参考文献和致谢。

# 开发环境

## 2.1 系统开发环境搭建

.NET Framework 4.5.2 win8/win10系统 Office2013

## 2.2 软件工具

### 2.2.1 Visual studio 2015 community

VB.NET是Visual Basic的下一个版本。微软不是简单的在VB 6.0的基础上增加了一些新特性,而是重新更换定制了这个产品。VB.NET有两个新的窗体包(WindowsForms和WebForms)；一个用于无连接数据源的，新的ADO版本；改进的语言,去除了继承关键字,增强了类型安全性和高级开发者需要的低级结构。这些新的特点为VB开发者打开了新的门户；使用网页窗体和ADO.NET,可以快速开发可扩展的网站；使用继承,VB.NET现在真正支持面向对象编程；Windows窗体天然地支持可访问性和可视化继承特点。VB.NET现在已经与Microsoft Visual Studio.NET其他的几种语言完全综合在一起了。像所有其他.NET语言一样,VB.NET可以充分利用.NET框架必须提供的一切。

### 2.2.2 Office Word &Access

选用的是Microsoft Office Professional Plus 2013系列中的Word和Access，而且office2013 提供了Microsoft Word 15.0 Object Library类库。

简历生成系统采用的数据库是Microsoft Office 2013 Access数据库，用Access 2013作为后台支持数据库，通过VB.NET的数据库控件来连接Access 2013中并对其编程来实现各种功能。因为Access 2013与Visual Basic有着很好的兼容性。

## 2.3　VB.NET环境

第一次启动VB.NET时，你首先注意到的不会是VB优秀的面向对象支持，而应该是它的IDE（VB.NET集成开发环境，Integrated Development Environment）。VS.NET的IDE给人以非常熟悉的感觉，因为设计这个IDE的工作组以前就设计过VB的IDE，VS.NET IDE的改进建立在设计VB IDE的经验之上。

对于IDE的改进并不停留在表面上。所有.NET语言都使用相同的IDE，IDE内提供的新工具非常全面和强大。例如，所有的设计窗口都可以自动隐藏（就像Windows任务条一样），从而显著地减少了混乱的感觉。

启动VB.NET时，“VS Home Page”是我们看到的第一个界面。Home Page中显示了最近修改的工程，并提供了打开现有工程、创建新工程的选项。所有这些选项都以HTML链接形式显示。注意这并不是什么模拟出来的效果，Home Page的布局事实上就是由DHTML编写而成。由于这个原因，用户可以定制Home Page显示其他对自己有用的选项、信息以及HTML链接。

New Project对话框中提供了用任意一种语言创建工程的选项，注意树形列表中列出了好几种语言。如果系统中已经安装了支持VS.NET的第三方语言，Microsoft或许还会在这个列表中包含这些语言。

树形列表中的VB节点处于打开状态，右边的一些图标显示了创建新工程的几个选项。注意这里出现了一些新的选项，如Web Service、Web Control Library、Class Library。最后一个选项显示出：在VB.NET中，多个相关的类可以合并放入单个模块文件之中，而不是像VB6那样把每个类单独放入一个文件。

激活WinForms工程时的IDE与传统VB IDE最为相似。

◆隐藏工具框。默认情况下，工具框大多数时候都是隐藏的，点击它的标题可以把它“拉出来”。对于VB开发者来说，这里最大的变化在于工具框中的控件现在改为垂直排列，每个图标都加上了文字说明，而不像VB6那样只有一个图标。

◆卡式子窗口（Tabbed Child Window）。VS.NET利用位于屏幕上方的卡式子窗口布局取代了原来的MDI子窗口布局。卡包括：Home Page，窗体布局窗口，代码窗口，帮助屏幕。

◆任务列表（Task List）。Task List是一个重要的改进，它帮助开发者跟踪所有待完成的任务。代码生成器、编译器会在Task List中自动加上一些内容。手工添加任务项目可以按照如下方式进行：找到代码中需要修改的地方，然后插入一行以“TODO:”开头的注释，此后该任务就会自动出现在任务列表中。点击Task List中的某个任务，代码编辑器中就会出现完成该任务的位置。这种快速定义任务、返回任务的机制将极大地提高开发效率。

◆方案资源管理器（Solution Explore）。它替代了原来的工程资源管理器（Project Explorer），与Visual InterDev中的资源管理器非常相似。和InterDev中的资源管理器一样，Solution Explorer显示了工程所用到的各种资源，包括用其他语言编写的代码模块、各种非代码的资源。其中References是Solution Explorer中一种全新的资源形式，References管理着命名空间，类似于VB6中的组件引用。

◆属性框（Properties）。Properties框与VB6中的属性框非常相似，但它现在支持更多的布局方式（显示方式）。

◆窗体布局工具条（Form Layout）。在VB6中，在窗体上排列控件的任务通过一个名为“Form Editor”的工具条完成。改进后的新工具条提供更多的功能，而且在WinForms类型的工程中默认打开。

除了上面六处改进之外，VB.NET的代码编辑器还有一个重要的新功能：在代码编辑窗口中，点击代码行左边的加号或者减号可以扩展或者折叠代码块。这个功能原本为隐藏可视化设计器自动生成的代码而设计，但它也可以用于隐藏与当前任务无关的代码。

Web Forms工程的设计环境也和原来VB6的设计环境略有不同。在表单布局窗口的下面只显示了两个卡，分别为Design和HTML。这两个卡的功能和Visual InterDev中同一位置的卡相似。Design窗口用来编辑窗体的外观，HTML窗口用于查看DHTML代码。

VB.NET另外一个引人注目的变化是它的窗体引擎。Microsoft放弃了原来的窗体引擎，现在它给我们提供的是Windows Forms窗体引擎。所有基于CLR的语言都使用Windows Forms引擎，与VB 6的窗体引擎相比，Windows Forms有着许多突出的优点。例如，Windows Forms支持创建那些自动改变组件大小的窗体，允许把控件锚定到窗体的特定位置。也就是说，完成这些特殊任务时再也不需要第三方的工具了。Windows Forms还有一些有趣的技术，比如创建透明窗体等。

以前，VB隐藏了窗体构造过程中的所有细节。我们在IDE 中设计窗体，然后把代码加入到Initialize事件，但对这两者之间的处理过程却无法进行任何控制。现在，窗体成了类，类里面包含了构造窗体的全部代码。大多数程序员都不会去看这种描述窗口构造细节的代码，如果说有某种操作一定会破坏应用，这个操作就是搞乱这里的代码。另一方面，由于这些代码允许用户深入控制VB.NET构造窗体的幕后细节，有经验的用户能够利用这些代码实现各种高级功能。如果你不想看到这些代码，你可以不看，因为新的代码编辑器有扩展和折叠代码区域的功能，描述窗体构造过程细节的代码默认处于折叠状态。代码编辑器还有其他一些新的实用功能，比如自动缩进、内建的行号显示功能等。

## 2.4 数据库OLEDB对象

OLEDB(Object Linking and Embedding Database,又称为OLE DB或OLE-DB，对象链接和嵌入数据库)是微软的战略性的通向不同的数据源的低级应用程序接口。OLE DB不仅包括微软资助的标准数据接口开放数据库连通性（ODBC）的结构化问题语言（SQL）能力，还具有面向其他非SQL数据类型的通路。作为微软的组件对象模型（COM）的一种设计，OLE DB是一组读写数据的方法（在过去可能被称为渠道）。OLD DB中的对象主要包括数据源对象、阶段对象、命令对象和行组对象。OLEDB的作用如下。

(1) Data Providers数据提供者

凡是透过OLEDB将数据提供出来的，就是数据提供者。例如SQL Server数据库中的数据表，或是附文件名为mdb的Access数据库档案等，都是Data Provider。

(2) Data Consumers数据使用者

凡是使用OLEDB提供数据的程序或组件，都是OLEDB的数据使用者。换句话说，凡是使用ADO的应用程序或网页都是OLE DB的数据使用者。

(3) Service Components 服务组件

数据服务组件可以执行数据提供者以及数据使用者之间数据传递的工作，数据使用者要向数据提供者要求数据时，是透过OLEDB服务组件的查询处理器执行查询的工作，而查询到的结果则由指针引擎来管理。

# 系统分析

首先将office操作代码封装成DLL文件，然后程序调用这个DLL文件，完成文档输出和输入文本、标题、图片、表格的内容和格式的功能，实现简历生成系统。该章节则先分析封装的意义和如何封装，并如何调用，再研究简历生成系统的结构和模块功能实现。

## 3.1 VBA封装类的需求

### 3.1.1 VBA封装的背景

因为宏具有一些缺陷，如缺少伸缩的弹性、记录了许多不需要的步骤、记录了许多不需要的资料、无法使用较深入的功能、无法完成重复性的工作、无法完成需要条件判断的工作等，因此显现出VBA突出的优越性：VBA是一个通用程序语言，可共享Microsoft各种相关的重要软件；以VBA编写的程序允许用户将其复制到Visual Basic中加以调试，用Visual Basic宏来控制Excel；当记录宏不能满足用户需要或用记录宏无法记录命令时，可以用VBA语言创造宏来控制工作簿的各项操作；创建用户对话框、工具按钮或自定义命令，把Excel提供的多个工具按钮连接起来，作为一个工具按钮；利用重复语句控制循环宏内操作，利用不同参数执行某个相关功能模块，使宏指令完成一系列复杂操作来连接多个宏；VBA提供了许多内部函数，还允许用户自定义函数来简化对工作簿、工作表、图表等复杂操作。

但是VBA在代码的保护上却存在着缺陷。如果不对VBA工程设置密码保护。代码很容易被人浏览乃至窃取。而且即使利用其自身提供的密码保护机制设置了密码保护，也很容易被破解，互联网上有很多针对于Word、Excel的密码破解器都可以很容易地对密码进行破解。而Visual C++、Visual Basic、C++Builder和Delphi等编译器所编译出来的程序安全性相对较高，这是因为其代码可编译成可执行文件或者动态链接库文件。动态链接库(Dynamic--Link Library)，是基于Windows程序设计的一个非常重要的组成部分。将程序制作成动态链接库，不仅能节省内存开销，减少开发的工作量，提高访问速度，更重要的是能将程序源代码进行封装，提供程序的安全性。利用这一特点，可以将VBA代码封装到动态链接库文件中，然后在Office文件中进行调用，从而提高安全性。

本文以VBA环境为基础，通过将VBA代码封装成DLL文件，并在Word中引用DLL的方法，保证VBA源代码的数据安全，提供了一种针对于开放式平台下软件开发的安全处理措施。以下操作的前提是已经建好一个VBA工程，且调试通过，正确无误。

### 3.1.2 VBA对Word的开发与存在问题

(1) 录制宏

录制宏是当前VBA对Word最简单、常见的开发模式。这种方法优点是简单，易实现，完全可视化操作，用户不需编写任何代码，只要按照功能需求，一步步操作Word即可，VBA会根据用户的操作，自动生成相应的VBA程序代码，并保存在指定的文件中。

但该法的缺点也很明显，就是能扩展的功能非常有限。所谓录制宏，实际就是利用已有的Word功能，将需要反复进行的一系列操作，组合为一步操作，并作以VBA程序的形式保存下来。用户在打开该文档的同时，启动文档中已保存的VBA程序， 并根据用户的指令运行、发挥作用。因此，这种开发完全依赖于Word中已有的可操作功能， 只能在功能组合的基础上进行。

(2) VBA环境开发

利用VBA对Word进行二次开发，更重要的方法是在Word中集成的VBA环境里，通过编写程序实现。 VBA环境不仅为开发者准备了窗体、按钮、组合框等常用的Windows图形化控件，还继承了VB中的模块、类、对象等概念。开发者不仅可以摆脱Word已有功能的掣肘，还可以像在VB中一样，自由编写出功能强大的程序。甚至可以结合其它软件，相比于“录制宏”，更能扩展Word的功能。

如在日常办公中， 有时需将Word与Excel两个软件结合使用。那么，在VBA中，只要以下简单的程序，即可实现Word对Excel的调用：

Set E\_obj = CreateObject(“Excel.Applicattion”)

E\_obj.visible = True

但无论是利用“录制宏”还是利用VBA环境进行Word的二次开发，其共同的缺点都是所开发出来的程序，必须与Word文件捆绑保存，无法独立于Word程序独立存在运行。因为在VBA程序中的许多对象、 系统变量与系统常量都完全来源并被识别于Word，并且只有开发所在的机器与用户能够正常使用。例如若在VBA 中有以下程序：

ActiveDocument.Paragraphs(3).Range.Characters(2).Font.Size = 14

程序中的ActiveDocument，Paragraphs，Range，Characters都属于Word的专有对象，如果程序被独立运行的话，便会报错中断运行，因为脱离Word环境以后，这些对象就无法再被识别，成为无效引用。

### 3.1.3 DLL的必要性

DLL文件中存放的是各类程序的函数(子过程)实现过程，当程序需要调用函数时需要先载入DLL，然后取得函数的地址，最后进行调用。使用DLL文件的好处是程序不需要在运行之初加载所有代码，只有在程序需要某个函数的时候才从DLL中取出。由此DLL很好地解决了VBA录制宏和开发过程中的缺陷，可生成扩展功能并可以被有效引用。

因此本设计中VBA二次开发为了解决VBA的缺陷，DLL是必不可少的。

## 3.2 DLL类需求

(1) 文本类

实现插入文本，设置文字大小、文字颜色、文字字体、加粗、斜体、下划线、删除线的格式，并设置该段落的首行缩进或者悬挂缩进。

(2) 标题类

实现插入标题及其一、二、三标题序号，设置标题大小、标题颜色、标题字体等格式。

(3) 图片类

实现插入图片，设置图片布局、图片路径、图片大小、图片透明度、图片对比度、图片缩放比例等功能。

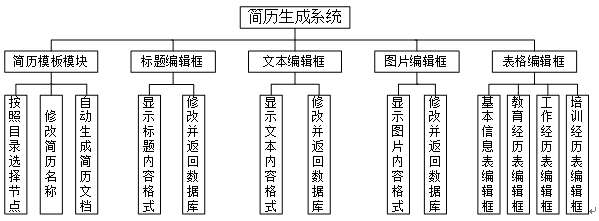
(4) 表格类

实现插入表格，能够在其某个单元格输入文本。

(5) 文档输出类

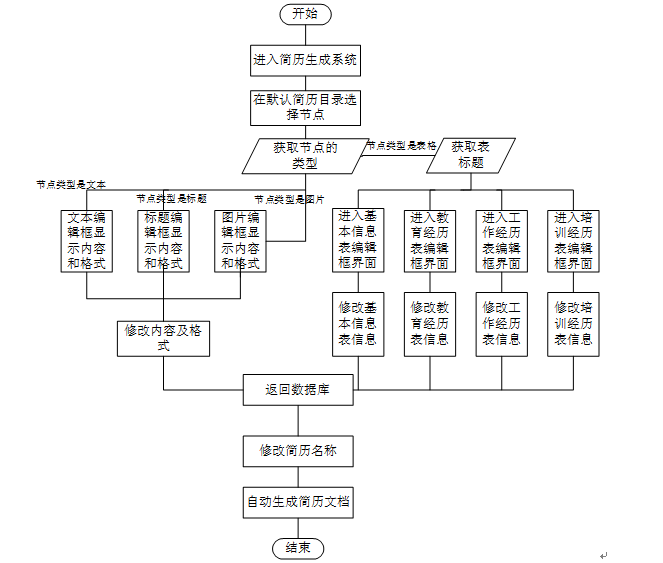
实现Word文档输出，并能够按照上面用户所添加的内容及其格式排版。

## 3.3 简历生成系统框架

在充分进行用户需求分析的基础上把简历生成系统分为五个子模块：简历模板模块、标题编辑框、文本编辑框、图片编辑框、表格编辑框。这五个模块之间，紧密结合，共享信息资源，形成一种简便的个人简历管理的解决方案。结构图如图3-1所示。

**图3-1 简历生成系统体系结构**

## 3.4 简历生成系统功能流程分析

简历生成系统的主要功能流程图，如图3-2所示。

**图3-2 简历生成系统功能图**

## 3.5 简历生成系统功能需求

简历生成系统是为了对简历信息实行计算机化的管理，以提高工作效率，方便用户。主要功能有对文本编辑、标题编辑、图片编辑等进行管理和检索。提供多种信息的录入，比如基本信息表、培训经历表等。系统不但含有默认简历中各标题对应的信息，而且还包含了对信息的查询编辑功能。对文本能够设置文字大小、文字字体、颜色等属性，对标题能够检索其标题序号，对图片能够设置图片布局以及其图片路径，对表格能够输入相应的信息，最后输入到Word文档中。

该系统是基于Windows窗体，以VB.NET连接Access 2013数据库进行开发的简历管理系统，力求与实际相结合具有学生简历管理和检索等功能，旨在达到使Word文档的管理数字化，使之更加方便快捷，以提高工作效率。

MyDll类：封装各种office功能代码并被调用。

简历模板模块: 用户使用说明，显示模板各目录的控件，简历名称，简历的输出。

标题编辑模块：文本编辑框，标题级别，一级标题序号，二级标题序号，三级标题序号，标题颜色，标题大小，标题字体和确认、取消。

文本编辑模块：文本编辑框，文字颜色，文字大小，文字字体，对齐方式，加粗，斜体，下划线，删除线，首行缩进，悬挂缩进和确认、取消。

图片编辑模块：图片编辑框，图片路径，图片布局，图片高度，图片宽度，图片对比度，图片透明度，图片缩放比例和确认、取消。

基本信息表模块：姓名，性别，出生日期，手机号码等信息和确认、取消。

教育经历表模块：起止时间，学校，专业，学历，专业描述信息和确认、取消。

工作经历表模块：起止时间，公司，公司规模，公司性质等信息和确认、取消。

培训经历表模块：起止时间，培训机构，地点，课程，证书，描述信息和确认、取消。

## 3.6 数据库需求设计

### 3.6.1 数据库表列单

根据系统的功能分析，可以大致地看出简历所需要存储的一些数据信息，比如标题级别，颜色，对齐方式，内容等等，根据这些数据信息可以抽象出本系统所需要的数据库表——MyOffice表，其结构如表3-3所示。

**表3-3 MyOffice数据表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 说明 |
| ID | 数字 | 长整型 | 数据记录的序号，主键 |
| 类型 | 短文本 | 255 | 包括文本、标题、表格、图片、单元格 |
| 标题级别 | 数字 | 长整型 | 分为1,2,3级别 |
| 颜色 | 短文本 | 255 | 包括红、蓝、绿、白、灰、粉、黄、紫、黑色 |
| 对齐方式 | 短文本 | 255 | 包括居中、左对齐、右对齐、分散对齐、两端对齐 |
| 图片布局 | 短文本 | 255 | 包括嵌入型、紧密型环绕、四周型环绕、上下型、穿越型、浮于（衬于）文字上（下）方 |
| 文字大小 | 短文本 | 255 | 包括初号、二号、小二、5、5.5、6.5等 |
| 文字字体 | 短文本 | 255 | 包括宋体、黑体、华体、Calibri等 |
| 内容 | 长文本 |  | 要输入或编辑的内容 |
| 加粗 | 是/否 |  | 文字是否加粗 |
| 斜体 | 是/否 |  | 文字是否斜体 |
| 下划线 | 是/否 |  | 文字下方是否有下划线 |
| 删除线 | 是/否 |  | 文字下方是否有删除线 |
| 图片路径 | 短文本 | 255 | 选择的图片的路径 |
| 目录内容 | 短文本 | 255 | 模板的目录 |
| 图片透明度 | 数字 | 长整型 |  |
| 图片对比度 | 数字 | 长整型 |  |
| 图片高度 | 数字 | 长整型 |  |
| 图片宽度 | 数字 | 长整型 |  |
| 图片缩放比例 | 数字 | 长整型 |  |
| 表行数 | 数字 | 长整型 |  |
| 表列数 | 数字 | 长整型 |  |
| 表标题 | 短文本 | 255 |  |
| 段落行距 | 数字 | 长整型 | 该文本的段落行距 |
| 首行缩进 | 数字 | 长整型 | 该文本段落首行缩进几个字符 |
| 悬挂缩进 | 数字 | 长整型 | 该文本段落悬挂缩进几个字符 |
| 表高度 | 数字 | 长整型 |  |
| 表宽度 | 数字 | 长整型 |  |
| 一级标题 | 数字 | 长整型 | 该标题的一级标题序号 |
| 二级标题 | 数字 | 长整型 | 该标题的二级标题序号 |
| 三级标题 | 数字 | 长整型 | 该标题的三级标题序号 |
| 页边距 | 数字 | 长整型 |  |
| 表数 | 数字 | 长整型 | 辨别是哪张表格 |

# DLL类设计

## 4.1 VBA程序封装成DLL文件

### 4.1.1 关于DLL与VBA代码封装

DLL即动态链接库（Dynamic Link Library），是由可被其它程序调用的函数集合组成的可执行文件模块。DLL不是应用程序的组成部分，而是运行时链接到应用程序中。将VBA程序封装成DLL就是通过编译软件，将二次开发的VBA程序编译成DLL文件，以方便地为其它用户引用到各自的Word程序中运行使用。其基本的实现流程如图4-1所示：



**图4-1 实现流程**

具体的实现过程，不是本文阐述的重点，在此不再累述。通过DLL封装进行程序移植，其优点主要有以下几点：

(1) 一个DLL可以为多个Word文档同时共享， 而且当多个Word文件调用DLL中的同一个函数时，装入的只是该函数的内存地址，从而节省内存和磁盘空间；

(2) 便于开发者维护用户程序，即使对动态链接库进行修改也不会影响正在使用的用户程序；

(3) VBA代码通过编译封装成为DLL文件以后，是以二进制的形式保存在磁盘中的，这就可以保证代码的安全。

(4) 省去了运行时的编译过程，可以提高代码的运行速度。

因此，不考虑实现过程相对繁琐的话，通过DLL封装， 是VBA程序移植的最佳方案。

### 4.1.2 消除宿主程序的对象模型对封装DLL的影响

Word中的VB编辑器不能够将VBA代码编译成DLL。因此，必须借助微软的VB软件。但直接将VBA代码导入VB进行编译，通常都会报错导致编译中断。错误信息是运行时错误，对象的引用无效。这是因为所编译的代码中，含有Word程序的特有对象，在Word中运行这些代码时，程序能够识别这些对象。代码移植到VB 中以后，VB本身并没有这些对象，因此，编译时，就会认为是无效的引用而报错。

因此，将VBA代码移到VB中进行编译工作之前，必须详细了解Word的对象模型，并对VB 中的代码进行修改，使其能够通过VB的语法检查，又能为Word所运行，才能完成DLL的编译封装。

编译前的代码修改，主要从以下两个方面检查：

(1) 修改VBA代码中Word所特有的对象。如果所要编译移植的代码中，含有Document、Selection、Tables等等Word所特有类型的对象(如图4所示)，那么VB是无法识别的。必须给这些对象所在的过程或函数，定义一个数据类型为Object的参数，将这些对象通过这个参数进行传递调用。如在一个过程Removal\_Test 中要访问对象Tables，即Removal\_Test可定义为Removal\_Test(Test\_Tab As Object)。那么Tables对象就可以用以下程序进行调用：

Dim new\_table as object

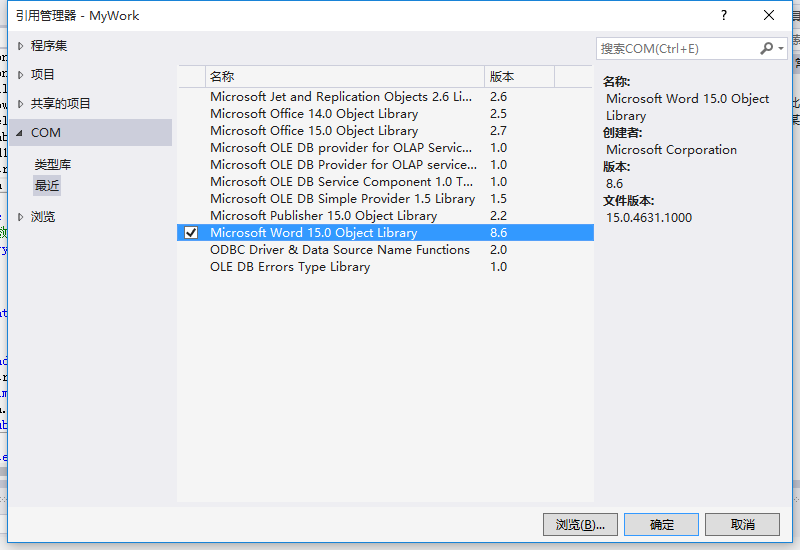
Set new\_table = CreateObject (“,” Word.Application”) ‘

Removal\_Test (new\_table)

(2) 修改Word的VBA专属常量。Word的VBA专属常量以“wd”开头，不同的宿主程序的VBA专属常量具体有所区别，必须根据实际情况，进行代码的修改编写。比如代码中有设置段落格式的命令“. ParagraphFormat. Alignment = wdAlignParagraphCenter”。其中的wdAlignParagraphCenter就属于Word 的专属常量。如果在VB中直接编译，就会出现“类型不匹配”或“该属性值不是一个有效值”的错误。 解决的方法是需要使用常量的地方，用“常量值”来替代“常量名”。. ParagraphFormat.Alignment= 1。常量值的获取，可以查阅有关文档资料，也可以在Word的VB编辑器中用Msgbox方法， 让Word自己将值返回。如Msgbox.wdAlignParagraphCenter。

### 4.1.3 相关类库的引用

VB环境中，许多对象依然需要相关的类库文件来支持，才能支持识别其相关的属性与方法。

因此，将程序编译封装前，还必须在VB环境中的【工程】菜单下，通过【引用】菜单项，调出“引用”对话框，然后将所需的类库引用到VB中来，如图4-2所示。

**图 4-2 类库引用**

在Word中利用VBA进行的二次开发程序，必须引用的至少有两个类库： Microsoft Word 15.0 Object Library与Microsoft Office 15.0 Object Library。不同版本的Office，类库的版本也相应不同。 如果在开发的过程中，还曾经引用过其他类库，在VB中编译时也必须先将这些类库引用进来。

## 4.2 文本类

文本类实现了插入文本及其属性设置的功能。

方法 ：

①插入文本 ：InsertText(ByVal text As String)

使用了Range.InsertAfter() 方法

②文本字体大小 ：FontSize(ByVal FontSize As String)

使用了Range.Font.Size属性

③文本文字字体 ：FontName(ByVal FontName As String)

使用了 Range.Font.Name 属性

④文字加粗 ：Bold()

使用了 Range.Font.Bold属性

⑤文字颜色 ：Fontcolor(ByVal ColorIndex As String)

使用了 Range.Font. ColorIndex 属性

⑥文本对齐方式 ：Alignment(ByVal position As String)

使用了 Range. Paragraphs.Alignment 属性

⑦文字斜体 ：Italic()

使用了 Range.Font. Italic属性

⑧文字下划线 ：Underline()

使用了 Range.Font.Underline 属性

⑨文字删除线 ：Strikethrough()

使用了 Range.Font.StrikeThrough 属性

⑩文本首行缩进（悬挂缩进）：FirstLineIndent (ByVal Indent As Integer)

使用了 Range. ParagraphFormat.CharacterUnitFirstLineIndent 属性

## 4.3 标题类

标题类实现了插入标题及其属性设置的功能。

方法 ：

①插入标题 ：Title(ByVal text As String, ByVal FontName As String, ByVal FontSize As Integer)

使用了MyRange.InsertAfter()方法Range.Font.Name 、.Font.Size 、.Font.Bold 、.Paragraphs.Alignment属性

②设置标题级别 ：TitleLevel(ByVal level As Integer)

使用了Range.ParagraphFormat.OutlineLevel属性

## 4.4 图片类

图片类实现了插入图片及其属性设置的功能。

方法 ：

①插入图片 ：InsertPic(ByVal position As String)

使用了Range.InlineShapes.AddPicture() 方法

②设置图片大小 ：PicSize(ByVal Height As Integer, ByVal width As Integer)

使用了Range.InlineShapes(1).Height 和Range.InlineShapes(1).Width 属性

③设置图片缩放比例 ：PicScale(ByVal value As Integer)

使用了Document.InlineShapes(n).Height \*= value的方法

④设置图片对比度 ：PicContrast(ByVal value As Double)

使用了Range.InlineShapes(1).PictureFormat.Contrast属性

⑤设置图片透明度 ：PicBright(ByVal value As Double)

使用了Range.InlineShapes(1).PictureFormat.Brightness属性

⑥设置图片布局方式 ：PicLayout(ByVal text As String)

使用了Range.ShapeRange(1).WrapFormat.Type属性

## 4.5 表格类

表格类实现了插入表格、输入单元格文本及其属性设置的功能。

方法 ：

①插入表格 ：InsertTable(ByVal Row As Integer, ByVal Column As Integer)

使用了Range.Tables.Add() 方法

②设置表格大小 ：TableSize(ByVal RowHeight As Double, ByVal ColumnWidth As Double)

使用了Range.Tables(1).Rows.Height和Range.Tables(1).Rows.Width 属性

③在表格某单元格输入文本 ：TableText(ByVal RowNum As Integer, ByVal ColumnNum As Integer, ByVal text As String)

使用了Range.Tables(1).Cell(Row:=RowNum, Column:=ColumnNum)属性

④在下方插入行 ：TableAddrow()

使用了Range.Rows.Add()方法

⑤在右方插入列 ：TableAddcolumn()

使用了Range.Columns.Add()方法

⑥选中表格某单元格位置 ：TableCell(ByVal Row As Integer, ByVal Column As Integer)

使用了Range.Tables(1).Cell(Row:=RowNum, Column:=ColumnNum)属性

⑦跳出表格 ：OutTable()

使用了Range.InsertParagraphAfter()方法

## 4.6 文档输出类

文档输出类实现了输出、保存、打印文档及其属性设置的功能。

方法 ：

①创建Word Application对象 ：New()

使用了Word = CreateObject("Word.Application") 和Word.Visible属性

②创建新文档 ：NewDocument()

使用了Document = Word.Documents.Add 和Document.Activate()方法

③使用模板生成新文档 ：NewDocWithModel(ByVal FileName As String)

使用了Document = Word.Documents.Add 和Document.Activate()方法

④打开已有文档 ：OpenFile(ByVal FileName As String)

使用了Document = Word.Documents.Open 和Document.Activate()方法

⑤退出文档 ：Quit()

使用了System.Runtime.InteropServices.Marshal.ReleaseComObject()方法

⑥另存为文档 ：SaveAs(ByVal FileName As String)

使用了Document.SaveAs()方法

⑦打印文档 ：PrintOut()

使用了Document.PrintOut方法

⑧设置文档页边距 ：FormatMargins(ByVal Left As Integer, ByVal Right As Integer, ByVal Top As Integer, ByVal Bottom As Integer)

使用了Document.PageSetup .LeftMargin .RightMargin .TopMargin .BottomMargin属性

⑨换行 ：NewLine()

使用了Range.InsertParagraphAfter()方法

## 4.7 DLL生成

新建类库项目，命名为“MyDll”，编辑完可行代码后，即可在“生成”菜单中选择“生成MyDll”选项，便在该项目中生成了DLL文件。提示：若是要修改代码，修改完后注意要重新生成DLL文件方可成效。

## 4.8 DLL调用

第一步 ：在MyWork 项目中，右击解决方案资源管理器的“引用”，选择“添加引用”，然后浏览文件找到MyDll文件，添加并确认。

第二步 ：在代码声明部分，添加”Imports MyDll”

第三步 ：在代码处声明Dim MyOffice = New MyDll.Office，然后即可使用MyDll的方法和类。

比如 ：MyOffice.NewDocument()即可实现生成新文档的功能。

# VBA封装类应用——简历生成系统

## 5.1 人机界面设计

人机界面，是人与机器进行交互的操作方式，即用户与机器互相传递信息的媒介，其中包括信息的输入和输出。好的人机界面美观易懂、操作简单并且具有引导功能，使用户感觉愉快、增强兴趣，从而提高使用效率。本系统基于Windows操作系统，因此在人机界面上遵循Windows的传统风格，由标题栏、菜单栏、快捷工具按钮、对话框等要素组成。与Windows操作系统一致的人机界面，使得用户只要具备Windows操作基础，就能很快适应本管理系统的操作使用，因此能够大大降低系统培训费用，提高系统的可推广性。

## 5.2 数据库应用

### 5.2.1 数据库创建并连接

第一步：打开Office 2013 Access ，创建空白数据库，建立MyOffice表，并保存为office.accdb。

第二步：打开VS2015 已创建的Visual Basic Windows 窗体应用程序”MyWork”项目

第三步：找到服务器资源管理器后，在“数据连接”右键“添加数据库”，数据源则是“Microsoft Access 数据库文件 (OLE DB)”，数据库文件则是刚刚创建的office.accdb。最后连接成功。

### 5.2.2 数据库的应用

本设计中OLEDB 12.0连接Access，并放入dsMsg数据集中的代码如下：

Public Sub Data ()

'数据库连接

Try

conn = New OleDb.OleDbConnection ()

conn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & Application.StartupPath & "\office.accdb"

conn.Open ()

Catch ex As Exception

MessageBox.Show (ex.Message)

Exit Sub

End Try

strSQL = "Select \* from MyOffice order by ID"

Dim da As New OleDb.OleDbDataAdapter (strSQL, conn)

da.Fill (dsMsg, "MyOffice")

End Sub

使用dsMsg数据集，将用户信息传入到数据库MyOffice表中的具体代码：

Dim MyOffice = New MyDll.Office

MyOffice.NewDocument ()

Try

row = dsMsg.Tables ("MyOffice").Rows.Count

For i = 0 To (row - 1)

type = dsMsg.Tables ("MyOffice").Rows(i)("类型").ToString

textA = dsMsg.Tables（"MyOffice").Rows(i)("内容").ToString

titleLevel = dsMsg.Tables ("MyOffice").Rows(i)("标题级别").ToString

color = dsMsg.Tables（"MyOffice").Rows(i)("颜色").ToString

picLayout = dsMsg.Tables（"MyOffice").Rows(i)("图片布局").ToString

fontSize = dsMsg.Tables（"MyOffice").Rows(i)("文字大小").ToString

fontName = dsMsg.Tables（"MyOffice").Rows(i)("文字字体").ToString

firstLineIndent = dsMsg.Tables("MyOffice").Rows(i)("首行缩进")

hangingIndent = dsMsg.Tables("MyOffice").Rows(i)("悬挂缩进")

alignment = dsMsg.Tables("MyOffice").Rows(i)("对齐方式").ToString

Select Case (type)

Case "文本"

MyOffice.InsertText(textA)

If fontSize IsNot "" Then

MyOffice.FontSize(fontSize)

End If

If fontName IsNot "" Then

MyOffice.FontName(fontName)

End If

If color IsNot "" Then

MyOffice.Fontcolor(color)

End If

If alignment IsNot "" Then

MyOffice.Alignment(alignment)

End If

If dsMsg.Tables("MyOffice").Rows(i)("加粗").Equals(True) Then

MyOffice.Bold()

End If

If dsMsg.Tables("MyOffice").Rows(i)("斜体").Equals(True) Then

MyOffice.Italic()

End If

If dsMsg.Tables("MyOffice").Rows(i)("下划线").Equals(True) Then

MyOffice.Underline()

End If

If dsMsg.Tables("MyOffice").Rows(i)("删除线").Equals(True) Then

MyOffice.Strikethrough()

End If

If firstLineIndent > 0 Then

MyOffice.FirstLineIndent(firstLineIndent)

ElseIf hangingIndent > 0 Then

MyOffice.FirstLineIndent(-hangingIndent)

ElseIf firstLineIndent = 0 And hangingIndent = 0 Then

MyOffice.FirstLineIndent(0)

End If

MyOffice.NewLine()

Case “标题” ………….

Case “图片” ………….

Case “表格” .…………

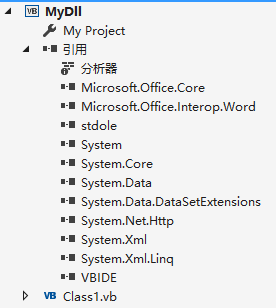
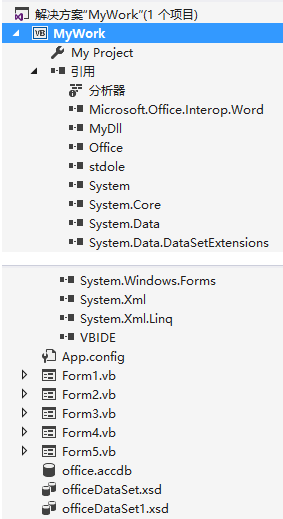
End Select

Next

Catch ex As Exception

End Try

## 5.3 系统开发文件部署

开发文件部署如图5-1所示。

**图5-1 系统文件**

## 5.4 系统模块详细设计与实现

### 5.4.1 简历模板模块

用户通过treeview控件选择要编辑的目录，并在相应的编辑框显示被选中的内容，根据需求修改后生成简历文档。

Form1窗体中

Label2 ：用户说明。

TreeView1 ：默认简历根节点下拉选择你要编辑的节点，并在相应模块显示内容或者格式。

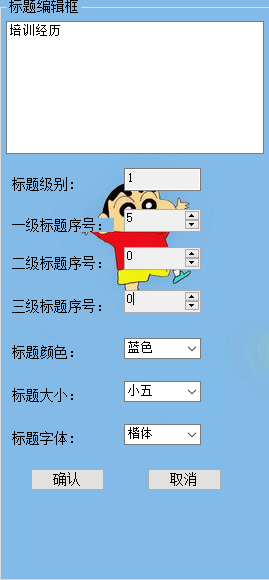
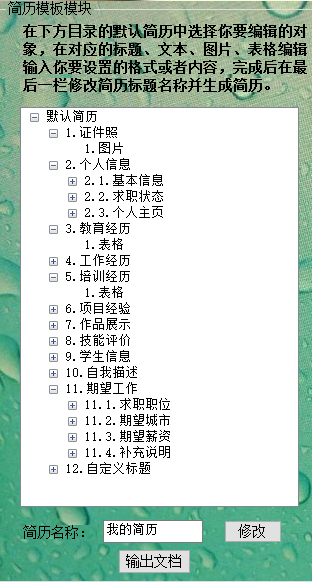
TextBox4 ：简历的名称。

Button9 ：将修改的简历名称返回到数据库中并弹出“修改成功！”的对话框。

Button1 ：点击触发自动打开Word2013并生成文档，并按照数据库的内容排版。

引用的类方法 ：NewDocument()、InsertText()、FontSize()、FontName()、FontColor()、Alignment()、Bold()、Italic()、Underline()、Strikethrough()、FirstLineIndent()、NewLine()、InsertPic()、PicLayout()、PicSize ()、PicBright()、PicContrast()、PicScale()、InsertTable()、TableText()、OutTable()

效果 ：实现了点击节点在相应编辑框显示内容的功能，点击“输出文档”按钮即可生成简历，并按照用户的需求排版。

模块界面如图5-2所示。

**图5-2 简历模板模块 图5-3 标题编辑模块**

### 5.4.2 标题编辑模块

模块界面如图5-3所示。

用户通过此模块来修改标题的内容，标题颜色，标题大小，并能查看其标题级别和序号。

Form1窗体中

TextBox3 ：显示被选中的标题的文本内容。

TextBox5 ：显示被选中的标题的级别（不可修改）。

NumericUpDown3 ：显示被选中的标题的一级标题序号（不可修改）。

NumericUpDown4 ：显示被选中的标题的二级标题序号（不可修改）。

NumericUpDown5 ：显示被选中的标题的三级标题序号（不可修改）。

ComboBox6 ：能够下拉选择标题的颜色。

ComboBox7 ：能够下拉选择标题的大小。

ComboBox7 ：能够下拉选择标题的字体。

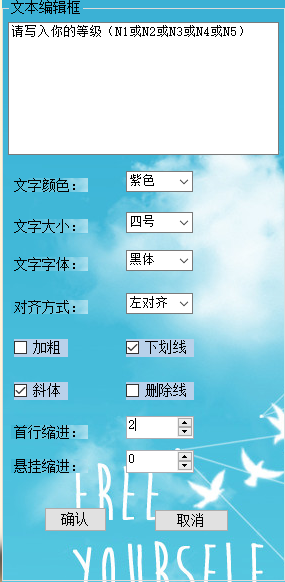
Button7 ：确认并将其内容和格式返回到数据库中并弹出“修改成功！”的对话框。

Button2 ：取消并不将做任何修改。

引用的类方法 ：Title()、TitleLevel()、FontSize()、FontName()、FontColor()、NewLine()

效果 ：能够显示标题内容和序号，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变。

### 5.4.3 文本编辑模块

模块界面如图5-4所示。

**图5-4 文本编辑模块**

用户通过此模块向数据库输入文本，并编辑其文字大小、文字字体、对齐方式、加粗、斜体、下划线、删除线，以及对该段落的首行缩进（悬挂缩进）格式进行修改。

Form1窗体中

TextBox1 ：显示被选中的文本的内容。

ComboBox1 ：能够下拉选择文字的颜色。

ComboBox2 ：能够下拉选择文字的大小。

ComboBox3 ：能够下拉选择文字的字体。

ComboBox4 ：能够下拉选择文字的对齐方式。

CheckBox1 ：能够选中是否加粗。

CheckBox2 ：能够选中是否斜体。

CheckBox3 ：能够选中是否有下划线。

CheckBox4 ：能够选中是否有删除线。

NumericUpDown1 ：能够选择首行缩进的字符。

NumericUpDown2 ：能够选择悬挂缩进的字符。

Button10 ：确认并将其内容和格式返回到数据库中并弹出“修改成功！”的对话框。

Button4 ：取消并不将做任何修改。

引用的类方法 ：InsertText()、FontSize()、FontName()、FontColor()、Alignment()、Bold()、Italic()、Underline()、Strikethrough()、FirstLineIndent()、NewLine()

效果 ：能够显示文本内容和格式，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变。

### 5.4.4 图片编辑模块

模块界面如图5-5所示。

**图5-5 图片编辑模块**

用户通过此模块浏览要上传的图片，并选择该图片的路径，编辑其布局、高度、宽度、对比度、透明度、以及缩放比例的属性。

Form1窗体中

PictureBox1 ：显示被选中的图片。

Textbox1 ：显示该图片的路径。

Button11 ：打开浏览文件夹对话框，并能够选中图片并返回其路径到Textbox1。

Combobox9 ：能够下拉选择图片的布局方式。

NumericUpDown6 ：能够设置该图片的高度。

NumericUpDown7 ：能够设置该图片的宽度。

NumericUpDown8 ：能够设置该图片的透明度。

NumericUpDown9 ：能够设置该图片的对比度。

NumericUpDown10 ：能够设置该图片的缩放比例。

Button8 ：确认并将其内容和格式返回到数据库中并弹出“修改成功！”的对话框。

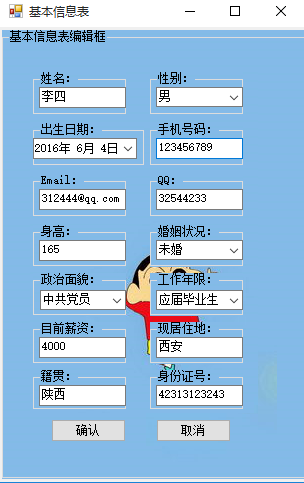
Button5 ：取消并不将做任何修改。

引用的类方法 ：InsertPic()、PicLayout()、PicSize ()、PicBright()、PicContrast()、PicScale()

效果 ：能够显示图片、路径和格式，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变。

### 5.4.5 表格编辑模块

用户通过此模块来修改个人资料，共有基本信息表、教育经历表、工作经历表、培训经历表四张表，每张表度对应相应的编辑界面。

1. ****基本信息表编辑界面如图5-6所示。

**图5-6 信息经历表编辑界面**

Form2窗体中

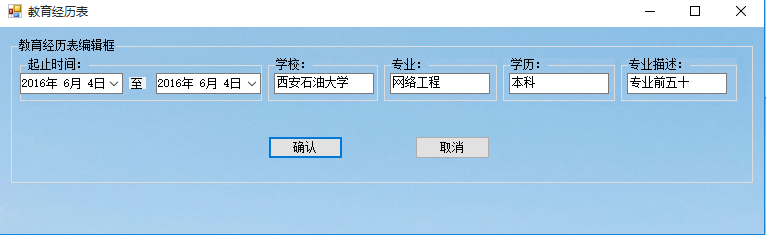
Textbox5、combobox10、DateTimePicker1、textbox8、textbox9、textbox10、textbox11、combobox11、combobox12、combobox13、textbox15、textbox16、textbox17、textbox18分别对应姓名、性别、出生日期、手机号码、Email、QQ、身高、婚姻状况、政治面貌、工作年限、目前薪资、现居住地、籍贯、身份证号的信息。

Button5 ：确认并将其表格信息返回到数据库中并弹出“修改成功！”的对话框。

Button1 ：取消并退出这个界面窗体。

引用的类方法 ：InsertTable()、TableText()、OutTable()

效果 ：能够显示基本信息表格信息，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变并关闭窗体。

1. ****教育经历表编辑界面如图5-7所示。

**图5-7 教育经历表编辑界面**

Form3窗体中

DateTimePicker2、DateTimePicker3 ：起止时间信息。

Textbox7 ：学校名称。

Textbox12 ：专业名称。

Textbox19 ：学历。

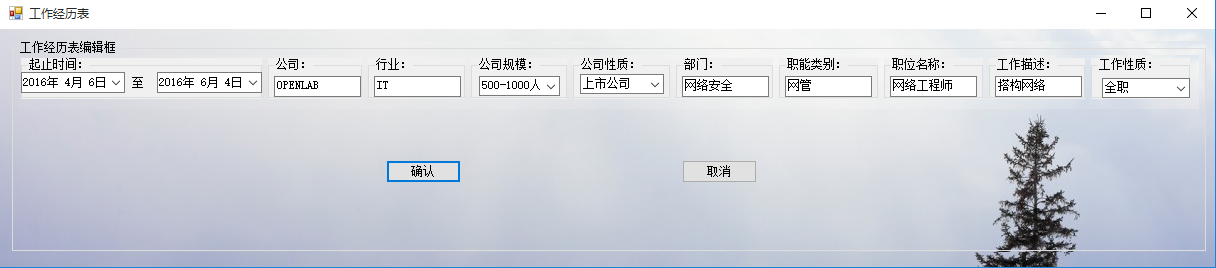
Textbox20 ：专业描述。

Button6 ：确认并将其表格信息返回到数据库中并弹出“修改成功！”的对话框。

Button1 ：取消并退出这个界面窗体。

引用的类方法 ：InsertTable()、TableText()、OutTable()

效果 ：能够显示基本信息表格信息，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变并关闭窗体。

1. ****工作经历表编辑界面如图5-8所示。

**图5-8 工作经历表编辑界面**

Form4窗体中

DateTimePicker4、DateTimePicker5 ：起止时间信息。

Textbox21 ：公司名称。

Textbox22 ：行业名称。

Combobox15 ：公司规模。

Combobox16 ：公司性质。

Textbox25 ：部门名称。

Textbox26 ：职能类别。

Textbox27 ：职位名称。

Textbox28 ：工作描述。

Combobox14 ：工作性质。

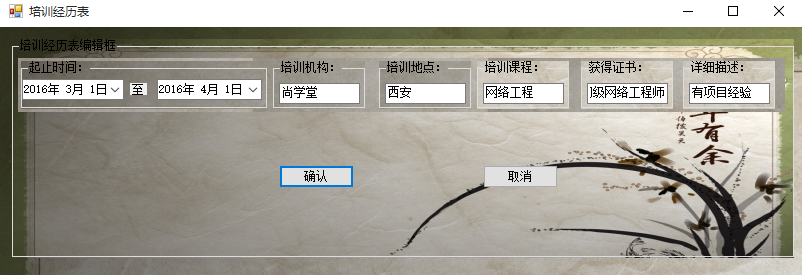
Button12 ：确认并将其表格信息返回到数据库中并弹出“修改成功！”的对话框。

Button1 ：取消并退出这个界面窗体。

引用的类方法 ：InsertTable()、TableText()、OutTable()

效果 ：能够显示基本信息表格信息，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变并关闭窗体。

1. 培训经历表编辑界面如图5-9所示。

** 图5-9 培训经历表编辑界面**

Form5窗体中

DateTimePicker6、DateTimePicker7 ：起止时间信息。

Textbox30 ：培训机构名称。

Textbox31 ：培训地点。

Textbox32 ：培训课程。

Textbox33 ：获得证书。

Textbox34 ：详细描述。

Button12 ：确认并将其表格信息返回到数据库中并弹出“修改成功！”的对话框。

Button1 ：取消并退出这个界面窗体。

引用的类方法 ：InsertTable()、TableText()、OutTable()

效果 ：能够显示基本信息表格信息，点击“确认”按钮并能成功修改数据。点击“取消”则不作任何改变并关闭窗体。

## 系统运行

最后修改简历名称为“我的简历”，按“生成简历文档”按钮，即可自动弹出Word2013并生成你想要的简历。生成简历文档展示效果图如图5-10所示。



****

**图5-10 简历文档展示效果图**

# 结论

经过几个多月的设计和开发，我用VBA二次开发的<简历生成系统>已经开发完毕。基本实现了对VBA的应用程序接口进行二次封装，对应Word文档标题、正文、图、表的元素的编程接口。使用了面对对象方法，将简历生成系统功能划分为默认简历模板模块、标题编辑框模块、文本编辑框模块、图片编辑框模块、表格编辑框模块，并能够和Access数据库表连接。最后生成的简历系统可供学生用户在该平台上编辑Word文档标题、正文、图、表，并能生成简历文档。其功能基本符合用户需求，能够完成基本功能。并提供部分系统维护功能,对于数据的一致性的问题也通过程序进行了有效的解决。但是由于毕业设计时间较短，所以该系统还有许多不尽如人意的地方，比如增添删除等功能比较少，用户界面不够美观，出错处理不够等多方面问题。这些都有待进一步改善。

总的来说，在这次的毕业设计种遇到了不少的问题，大部分得到了很好的解决，但也有少部分遗留的问题。

首先便是调研的不充分。由于以前根本未曾接触过软件的整体开发设计，所以在前期的调研中相当不充分，开题也有些盲目，以为把属性和表写得越多越好，这个问题越到后来越严重，而且因为客观的原因耽误了进度。

其次是软件的编程问题，由于以前对VB只是了解并不能够灵活运用，所以在毕设期间进行了苦读，终于可以基本掌握了常用控件的用法和语句编程。数据库的建设和使用，虽然Access数据库的安全性和普及性很高，但我们在日常中还是接触得比较少，所以这自然又是一番努力的学习，就拿安装配置以及连接数据库控件方面来说就不那么容易。

最终，会用了两种软件，如何把它们连接在一起还真是一件困难的事：选择连接的方法就很重要，通过各方面的比较，我选择了OLEDB，通过OLEDB与数据库中各表进行连接，同时也用到少数编程实现VB和ACCESS数据库的连接。

最后我从这几个月中感受到了VBA二次封装的各种便利和好处：更加节省内存并减少页面交换；DLL文件与Word文件独立，只要输出接口不变（即名称、参数、返回值类型和调用约定不变），更换DLL文件不会对Word文件造成任何影响，因而极大地提高了可维护性和可扩展性；不同编程语言编写的程序只要按照函数调用约定就可以调用同一个DLL函数；适用于大规模的软件开发，使开发过程独立、耦合度小，便于不同开发者和开发组织之间进行开发和测试。使用DLL文件的好处是程序不需要在运行之初加载所有代码，只有在程序需要某个函数的时候才从DLL中取出。另外，使用DLL文件还可以减小程序的体积。由于系统程序代码和动态链接库处于同一目录，可以保证程序的可移植性，在以后使用系统时就可以不用再打开VBA编辑器进行代码改动。

但是VBA终究不是万能的，同样也有各种缺陷，比如： 使用动态链接库的应用程序不是自完备的，它依赖的DLL模块也要存在，如果使用载入时动态链接，程序启动时发现DLL不存在，系统将终止程序并给出错误信息。而使用运行时动态链接，系统不会终止，但由于DLL中的导出函数不可用，程序会加载失败；速度比静态链接慢。当某个模块更新后，如果新模块与旧的模块不兼容，那么那些需要该模块才能运行的软件，则要删掉。

# 参考文献

[1] 何振林,胡绿慧.MS Office与VBA高级应用案例教程[M]. 北京:水利水电社,2010,09.

[2] 刘春裕.基于VBA开发技术的接口测试技术研究[J].计算机技术与发展,2014(1):69-72.

[3] 梁洁.利用VB实现OFFICE对象的访问控制[J]. 成都医学院学报,2007(1):48-51.

[4] 周维京.VBA封装技术分析[J]. 电脑与电信,2008(4): 34-35.

[5] 罗雨滋.在应用程序中使用Office对象的方法[J]. 辽宁师专学报(自然科学版),

2007(1):37-38.

[6] 邵小兵,廖慧芬.利用VBA实现Word自动排版[J]. 中国教育技术装备,

2005,(10):45-46.

[7] 林丕源.VBA在Word文档中的应用[J]. 现代计算机(专业版),2001,(9):85-89.

[8] 王鸣.用VBA实现数据库中数据在Word中置放[J]. 希望月报,2007,(3):4.

[9] 金良锋,周文详.WORD试验报告的自动生成[J]. 中国测试技术,2007.33(4):112-115.

[10] 鲁保玉,杨新芳.用Delphi生成Word报告及动态结构表格[J]. 计算机应用与软件,2007.24(3):180-183.

[11] 朱敏,沈同圣,王学伟,周晓东.VC++与VBA结合实现复杂报表[J]. 计算机应用与软件,2005.22(2):42-43,101.

# 致 谢

首先，要感谢我的指导老师李润洲老师。本课程设计的选题、构思、写作与定稿都离不开李老师的悉心指导和关怀。在毕业论文和设计的完成过程中，李老师在百忙之中查阅和修改本论文，给予了很多悉心的指导，对论文的修改建议很细致，给予了很多完善论文的启发。在指出论文中存在的问题和提供建设性修改意见的同时，也不忘鼓励我发扬长处，让我很受欣慰和鼓舞，从李老师身上，我学到了很多东西，从为人处事到求学求知，包括发现问题，分析问题，解决问题的态度和技巧，都将是我今后工作和生活中最宝贵的财富。通过与李老师的交流，以及整个毕业设计的完成实现，我在各个方面都得到了很大的提高，在这里，学生真诚地对涂老师表示深深的感激与谢意。

其次，还要感谢我的朋友们，在设计过程中他们也给予了很多帮助，给予了我很多新奇的创意和开阔的思路，在此向他们表示感谢。

通过这一阶段的学习和研究，系统终于得以实现，其中耗费了很多精力和时间，但本次设计是大学对即将走进社会的我们的一次知识和能力的综合考验，这将是激励我去创造的一个起点，会永远激励着我前进。

最后，衷心感谢各位老师在百忙中审阅我的论文。