|  |  |
| --- | --- |
| 2017届研究生硕士学位论文 | |
| 分类号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 学校代码： 10269 |
| 密级：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 学 号：51141211009 |



**East China Normal University**

**硕士学位论文**

**MASTER’S DISSERTATION**

**论文题目： Office主观题自动评阅系统研究和实现**

|  |  |
| --- | --- |
| **院 系：** | **计算机科学与软件工程学院计算中心** |
| **专 业：** | **计算机技术** |
| **研究方向：** | **现代软件技术** |
| **指导教师：** | **朱敏** |
| **学位申请人：** | **彭佳斌** |

2016 年09月22日

|  |  |
| --- | --- |
| Dissertation of 2017 Master Degree | |
|  | School Code：10269 |
|  | Student ID：51141211009 |

**East China Normal University**

**Title: RESEARCH AND IMPLEMENTATION OF OFFICE SUBJECTIVE ITEM AUTOMATIC MARKING**

**Department:**  **Computer Center of CSSE**

**Major: Computer Technology**

**Research Direction:** **Technology of Modern Software**

**Supervisor: Zhu Min**

**Candidate: Peng Jiabin**

September 22th, 2016

**华东师范大学学位论文原创性声明**

郑重声明：本人呈交的学位论文《Office主观题自动评阅系统研究和实现》，是在华东师范大学攻读硕士专业学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

**作者签名： 日期：** 年 月 日

**华东师范大学学位论文著作权使用声明**

《Office主观题自动评阅系统研究和实现》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士专业学位论文，著作权归本人所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和学校指定的相关机构送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

（ ）1.经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文\*，

于 年 月 日解密，解密后适用上述授权。

（ ）2.不保密，适用上述授权。

导师签名 本人签名

年 月 日

\* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

**彭佳斌 硕士学位论文答辩委员会成员名单（必须）**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 职称 | 单位 | 备注 |
|  |  |  | 主席 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **内容摘要**

随着社会信息化程度的不断提高，Office办公软件的应用已经深入各行各业当中，各个大学都会开设一门《计算机应用基础》课程，主要是教授对Office字处理文档，电子表格处理文档以及演示文稿文档的操作，并最终对学生的学习成果进行考核，考核的方式也逐渐从传统手工阅卷转变成自动阅卷。对客观题的自动阅卷在技术上没有太大难度，并且也已经有了很多成熟的策略，然而对主观操作题的自动阅卷还存在很多的技术难点需要解决，目前还没有非常好的解决方案。因此，如何利用一些新的策略、模型来客观、准确地评估被考核人对Office主观操作题的掌握程度，已成为研究的热点问题。本文研究Office主观题的阅卷问题，Office文档的信息量巨大、考核方式多种多样，与纯文本文件不同，其内部结构也非常复杂，用一般的文本对比来实现对Office文档的评阅是没有效果的。

要对Office文档主观操作题进行自动评阅大致存在如下几个方面的问题：从理论上如何建立一个自动评阅的模型；从实践上采用何种方式对该模型进行实现；如何获取Office文档中的信息；如何对考点类型进行划分；如何使得评卷过程更加客观；如何展示最终的评阅结果。这些问题还需要进行更深入的研究。

新版本的Office文档都遵循Open XML标准，本文就Office主观题自动评阅系统的实现提供一个可行的方案。通过分析Office文档的结构以及文档信息的获取方法，提出一种自动批改的实现思路，并就系统实现中的几个关键难点进行分析，提出解决问题的策略，最后根据上述方法和策略实现了一个Office主观题自动评阅系统。在系统实现过程中，注重系统的易用性和准确率，最终实验结果表明该自动评阅系统使用简单，性能良好，运行稳定，有较高的准确率。

关键，词：Open XML；Office主观题；自动评阅；

# **ABSTRACT**

With the development of the information society, office software has penetrated all walks of life, each university sets up a course called "The Foundation of Computer Application", mainly to teach about the operation of the office word processing documents, spreadsheets and presentations, and at the end of the semester has an assessment for students' learning outcomes. The method of assessment has gradually changed from the traditional manual marking into automatic marking. The automatic marking of objective item is technically not too difficult, and also has a lot of mature strategies. However, the automatic marking of subjective item has still many technical difficulties to be solved. There is no very good solution [at present](https://www.baidu.com/link?url=nSSU_XkXANFgYRbc0VGXTikcHQS8571SOcaL6TedS42v1UxdzavqLVtWbn8gflMsMGAXQxTsA5ilTt4Ujze9ZD-0KHU4906P0faL78f30Bq&wd=&eqid=b7d86e830000fcc60000000257df6994). Therefore, how to use some new strategies, models to assess objectively and accurately the level of mastery of the appraised person, has become a hot issue in research. This paper studies the automatic marking of office subjective item. the information of the office documents is huge and the assessment methods are varied. Different from the pure text file, office documents’ internal structure is very complicated. According to comparing the general text to mark office documents is of no effect.

There are several problems in implementation of office subjective item automatic marking system: how to establish an automatic marking model in theory; how to implement the model in practice; how to get the information in the office documents; how to classify the type of [examination point](http://www.baidu.com/link?url=Gtnnjrwl_edDGsZST4LJaadB7CpyCe4ZayrK29CbApGzoFCs2pRok0T1f58AypALMLgq4K1yTCPEp2vRz-ils-hwVi2HzHWyWZgruKdTBb8lw-EXcaAVYNJVAFMA2To0); how to make the marking process more objective; how to show the final evaluation results. These issues need to be further studied.

The new version of the office documents follows the standard of Open XML, this paper provides a feasible method to realize the office subjective item automatic marking system. Through the analysis of the structure of office documents and the acquisition method of document information, put forward a concept of automatic marking. Some key difficult point are analyzed and [propose](http://www.baidu.com/link?url=rtYIOn_Cm4NtLeYPqL-AiUVTnjpCTGz1tURc_ltiVRQs6mI_4OIzPaVTRdOjgb8QmljPtvH_2hUlWTagAv0TjiLNUjxvSuNFSb7nkj5XYXy) some strategies to solve them. At last, according to the methods and Strategies implemented an office subjective item automatic marking system. In the process of system implementation, focus on the system's usability and accuracy. Finally, the experimental results show that the automatic marking system is easy to use, good performance, stable operation and high accuracy.

**Keywords**: Open XML; office subjective item; automatic marking

# **目 录**

[内容摘要 I](#_Toc462066823)

[ABSTRACT II](#_Toc462066824)

[目 录 IV](#_Toc462066825)

[1 绪论 1](#_Toc462066826)

[1.1 研究背景及意义 1](#_Toc462066827)

[1.2 国内外研究现状 1](#_Toc462066828)

[1.3 本文主要工作和创新 1](#_Toc462066829)

[1.4 本文组织结构 2](#_Toc462066830)

[2 Open XML与Office文档结构特点 3](#_Toc462066831)

[2.1 Open XML的概念 3](#_Toc462066832)

[**2.1.1** **Open XML包的结构** 3](#_Toc462066833)

[2.2 Office Word文档结构特点 4](#_Toc462066834)

[**2.2.1** **Paragraph对象** 5](#_Toc462066835)

[**2.2.2** **Table对象** 8](#_Toc462066836)

[2.3 Office Excel文档结构特点 9](#_Toc462066837)

[**2.3.1** **SpreadsheetML中的表格** 10](#_Toc462066838)

[**2.3.2** **SpreadsheetML 中的工作表** 12](#_Toc462066839)

[**2.3.3** **SpreadsheetML 中的数据透视表** 15](#_Toc462066840)

[**2.3.4** **SpreadsheetML中的公式** 18](#_Toc462066841)

[**2.3.5** **SpreadsheetML中的条件格式** 19](#_Toc462066842)

[2.4 Office PowerPoint 文档结构特点 22](#_Toc462066843)

[**2.4.1** **演示文稿部件** 24](#_Toc462066844)

[**2.4.2** **演示文稿属性部件** 24](#_Toc462066845)

[**2.4.3** **幻灯片母板部件** 25](#_Toc462066846)

[**2.4.4** **幻灯片版式部件** 26](#_Toc462066847)

[**2.4.5** **幻灯片部件** 27](#_Toc462066848)

[2.5 本章小结 28](#_Toc462066849)

[3 Office文档信息获取方法 29](#_Toc462066850)

[3.1 Open XML SDK概述 29](#_Toc462066851)

[**3.1.1** **Open XML SDK 1.0** 29](#_Toc462066852)

[**3.1.2** **Open XML SDK 2.0** 29](#_Toc462066853)

[**3.1.3** **Open XML SDK 2.5** 30](#_Toc462066854)

[3.2 Word文档信息获取 30](#_Toc462066855)

[**3.2.1** **获取所有段落** 31](#_Toc462066856)

[**3.2.2** **获取某个段落的边框** 31](#_Toc462066857)

[**3.2.3** **获取边框属性值** 32](#_Toc462066858)

[3.3 Excel文档信息获取 32](#_Toc462066859)

[**3.3.1** **获取指定名称的Sheet** 32](#_Toc462066860)

[**3.3.2** **查找已命名单元格** 33](#_Toc462066861)

[**3.3.3** **获取单元格的值** 33](#_Toc462066862)

[3.4 PPT文档信息获取 33](#_Toc462066863)

[**3.4.1** **获取指定幻灯片** 34](#_Toc462066864)

[**3.4.2** **获取幻灯片内所有段落文本** 34](#_Toc462066865)

[3.5 本章小结 34](#_Toc462066866)

[4 自动评阅系统中的一些策略研究 35](#_Toc462066867)

[4.1 考点类型划分策略 35](#_Toc462066868)

[4.2 考点位置定位策略 36](#_Toc462066869)

[4.3 本章小结 37](#_Toc462066870)

[5 自动评阅系统的实现 38](#_Toc462066871)

[5.1 整体设计思路 38](#_Toc462066872)

[5.2 自动评阅系统模型设计 38](#_Toc462066873)

[5.3 Office自动评阅系统实现效果 39](#_Toc462066874)

[5.4 本章小节 40](#_Toc462066875)

[6 总结与展望 41](#_Toc462066876)

[6.1 工作总结 41](#_Toc462066877)

[6.2 研究展望 41](#_Toc462066878)

[参考文献 43](#_Toc462066879)

[致 谢 45](#_Toc462066880)

# 绪论

1. **研究背景及意义**

随着社会信息化程度的不断提高，办公自动化在这个过程中扮演着重要的角色，而对Office 办公软件的使用又是实现办公自动化的重要组成部分。正因如此，当前大多数高校都会给非计算机专业的大一新生开设一门《计算机应用基础》的课程，在这门课程中就包含了对Office办公软件的操作教学。目前对学生所做Office操作题的批改都是由老师手工进行，这项工作既耗时又容易由于主观因素出现批改标准不一致的情况。因此很有必要开发一款能够自动批改Office操作题的软件，将老师从繁杂的手工批改中解脱出来，实现自动批改。

1. **国内外研究现状**

目前，自动批改功能大致通过以下四种思路来实现：

* 构造一个模拟考试环境的仿真系统，并且记录考生的操作步骤，然后通过考生的操作步骤来进行自动阅卷。
* 采用Office中的VBA技术，通过对Office文档中对象的属性进行分析，并与设置好的的标准答案进行对比，最后得出分数。
* 通过将文档格式转换为其他格式，如RTF进行处理。
* 利用一些成熟的操作Office文档的API，采用“一题一编”的形式，对每一套不同的操作题都专门编写一个程序来进行批改。

1. **本文主要工作和创新**

本文从Office文档的结构出发，对Office Word文档、Office Excel文档以及Office PPT文档的结构进行分析，采用微软官方提供的Open XMLSDK工具对Office文档信息进行获取，同时探讨了Office主观题自动评阅系统实现过程中采取的考点类型划分策略和考点定位策略，并提出了自动批改系统的整体设计思路以及实现该思路所涉及到的三个模型，最终实现了一个基于Open XMLSDK的Office主观题自动评阅系统。该系统的创新之处在于以下三点：

* 采用Open XML SDK对文档信息进行获取，使得系统可以应用于所有遵循Open XML标准的 Office 软件套件。
* 利用配置文件的方式解决了一个自动批改程序只能批改一套试卷的问题。
* 在配置文件中采用粒度适中的考点类型划分策略，在保证批改结果准确的前提下尽可能降低配置文件的复杂性。

1. **本文组织结构**

本文对文Office主观题自动评阅系统进行了深入研究，论文组织结构如下：

第一章，绪论。主要介绍了Office主观题自动评阅系统的研究背景和意义以及当前的国内外研究现状，并对本文主要工作及创新点进行介绍。

第二章，主要介绍了Open XML的基本概念以及Office文档的结构特点，包含Word文档结构特点，Excel文档结构特点以及Power Point文档结构特点。

第三章，主要研究了Office文档信息获取的办法，介绍了进行信息获取的工具——Open XML SDK，并分三部分分别介绍了如何通过Open XML SDK获取Word文档信息，Excel文档信息以及PPT文档信息。

第四章，主要介绍了自动评阅系统中的一些策略研究，主要包括考点类型划分策略和考点位置定位策略。

第五章，主要介绍了Office主观题自动评阅系统的实现，包括系统整体的设计思路以及系统中设计到的三个模型。

第六章，总结与展望。主要对实现Office主观题自动评阅系统所做的研究工作进行一个总结，同时分析了本系统目前存在的一些问题，并指明了下一步的研究方向。

# Open XML与Office文档结构特点

* 1. **Open XML的概念**

Open XML是可由不同平台上的多个应用程序自由实现的字处理文档、演示文稿和电子表格的开放式标准。Open XML旨在如实表示用Microsoft Office应用程序定义的二进制格式进行编码的现有字处理文档、演示文稿和电子表格。使用 Open XML的原因很简单：现在存在数以亿计的文档，但遗憾的是，这些文档中的信息与创建文档的程序紧密耦合。Open XML标准的目的是分离由Microsoft Office应用程序创建的文档，以便其他应用程序可以独立于专有格式操作这些文档且不会丢失数据。

* + 1. **Open XML包的结构**

Open XML文件存储在ZIP存档中以方便打包和压缩。可以使用ZIP查看器来查看任何Open XML文件的结构。一个Open XML文档由多个文档部件构成。这些部件之间的关系存储在文档部件中。ZIP格式支持随机访问每个部件。例如，应用程序可以将一张幻灯片从一个演示文稿中移到另一个演示文稿中，而无需分析幻灯片内容。同样地，应用程序可以删除字处理文档中的所有注释，而不用分析文档的任何内容。

Open XML包中的文档部件是以XML标记的形式创建的，由于XML的结构为纯文本，因此我们可以使用文本读取器查看文档部件的内容，也可以使用类似XPath的进程分析内容。

从结构上来讲，Open XML文档是开放打包约定(OPC)包。如前面所述，一个包由一组文档部件组成。每个部件的部件名称由一个段序列或路径名称组成，如"/word/theme/theme1.xml"。包中包含一个 [Content\_Types].xml 部件，通过该部件可以确定包中所有文档部件的内容类型。以 .rels 扩展名结尾的关系部件中包含一组源包或部件的显式关系。

字处理文档通过使用WordprocessingML标记进行描述。一个WordprocessingML文档由一组文章组成，其中每篇文章属于下列项之一：

* 主文档
* 词汇表文档
* 页眉和页脚
* 注释
* 文本框
* 脚注和尾注

演示文稿通过使用PresentationML标记进行描述。演示文稿包可以包含下列文档部件：

* 幻灯片母版
* 备注母版
* 讲义母版
* 幻灯片版式
* 说明

电子表格工作簿通过使用SpreadsheetML标记进行描述。工作簿包可以包含：

* 工作簿部件（必需部件）
* 一张或多张工作表
* 图表
* 表
* 自定义 XML
  1. **Office Word文档结构特点**

WordProcessingML文档的基本文档结构由<document>和<body>元素组成，后跟一个或多个块级元素，如表示段落的<p>。一个段落包含一个或多个<r>元素。<r> 表示一段连续文本，它是包含一组共同属性（如格式设置）的文本区域。一段连续文本包含一个或多个<t>元素。<t>元素包含一系列文本。例如，对于只包含文本“测试文本”的Word文档来说，它的WordprocessingML 标记如下所示：

<w:document >

<w:body>

<w:p>

<w:r>

<w:t>测试文本</w:t>

</w:r>

</w:p>

</w:body>

</w:document>

WordprocessingML中的各个元素在Open XML SDK里面都有相应的类与之对应，这些类都在DocumentFormat.OpenXml.Wordprocessing命名空间当中，下表2-1列出了与document，body，p，r和t元素相对应的类：

表2-1 基本WordprocessingML元素与Open XML SDK类对应表

|  |  |  |
| --- | --- | --- |
| WordprocessingML元素 | Open XML SDK类 | 描述 |
| document | Document | 主文档部分的根元素 |
| body | Body | 段落、表格等块级元素的容器 |
| p | Paragraph | 段落 |
| r | Run | 一段具有共同属性的文本区域 |
| t | Text | 一段文本 |

一个典型文档可能包含批注、标题、页脚、脚注和尾注。所有这些其他部分都包含在字处理文档的zip包中。下图2-1显示了型文档中包含的许多部分：

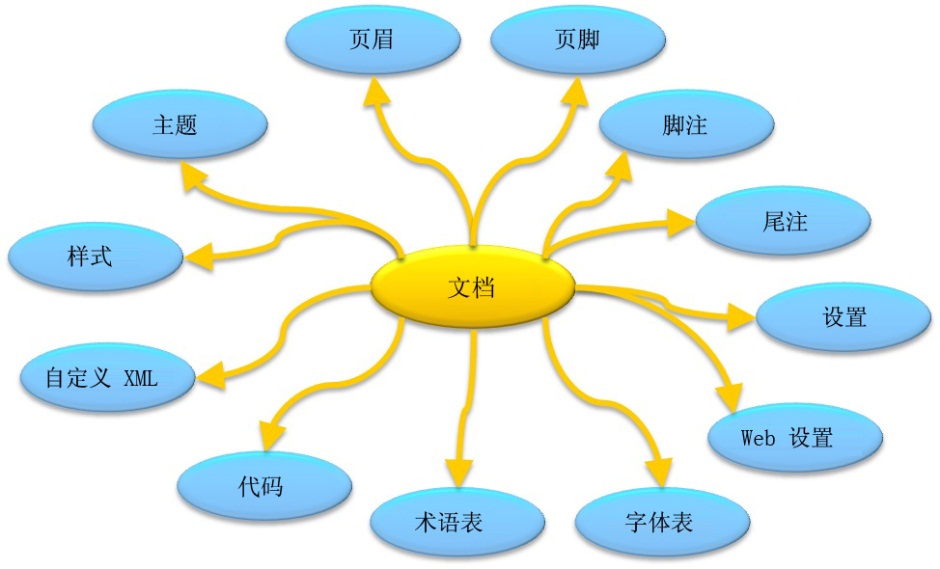


图2-1典型Office Word文档结构

* + 1. **Paragraph对象**

Paragraph对象代表文档中的段落，是文档的主要组成部分之一，每个文档由若干个段落组成。Paragraph对象对应的OpenXML中的标签为<w:p>，一个典型的文档结构如下所示：

<w:document>

<w:body>

<w:p>这是一个段落</w:p>

<w:p>这是一个段落</w:p>

<w:p>这是一个段落</w:p>

</w:body>

</w:document>

<w:p>节点有一个<w:pPr>子节点，对应ParagraphProperties对象，代表的是段落具有的属性，该元素的主要子节点如下表2-2所示：

表2-2 <w:pPr>子节点元素与Open XML SDK类对应表

|  |  |  |
| --- | --- | --- |
| WordprocessingML元素 | Open XML SDK类 | 含义 |
| <w:jc> | Justification | 段落对齐方式 |
| <w:pBdr> | ParagraphBorders | 段落边框 |
| <w:shd> | Shading | 段落底纹 |
| <w:spacing> | SpacingBetweenLines | 段落间距 |

<w:jc>节点代表段落的对齐方式，它有一个属性w:val，该属性值可取值为“center”，代表该段落的对齐方式为居中对齐。

<w:pBdr>节点代表段落的边框，若具有该节点，说明该段落具有边框，该节点用来控制段落的四个边框，如下表2-3所示：

表2-3 <w:pBdr>子节点元素与Open XML SDK类对应表

|  |  |  |
| --- | --- | --- |
| WordprocessingML元素 | Open XML SDK类 | 含义 |
| <w:top> | TopBorder | 上边框 |
| <w:left> | LeftBorder | 左边框 |
| <w:bottom> | BottomBorder | 下边框 |
| <w:right> | RightBorder | 右边框 |

对于上述的四个节点具有几个主要的属性如下表2-4所示：

表2-4 <w:pBdr>子节点元素主要属性取值举例及其含义

|  |  |  |
| --- | --- | --- |
| 属性 | 取值举例 | 含义 |
| w:val | single | 边框类型为单实线 |
| w:color | 0000FF | 边框颜色为蓝色 |
| w:sz | 24 | 边框粗细为24 |
| w:shadow | 1 | 边框具有阴影 |

例如对于上边框，可能具有如下结构：

<w:top w:val="single" w:color="00B050" w:sz="24" w:shadow="1" />

对于<w:shd>节点代表的是段落的底纹，该节点具有的主要属性如下表2-5所示：

表2-5 <w:shd>节点元素主要属性取值举例及其含义

|  |  |  |
| --- | --- | --- |
| 属性 | 取值举例 | 含义 |
| w:val | pct20 | 底纹图案样式为20% |
| w:color | 0000FF | 底纹颜色为蓝色 |
| w:themeColor | accent3 | 主题颜色为强调文字颜色3 |
| w:fill | auto | 图案颜色为自动 |

底纹节点举例如下所示：

<w:shd w:val="pct20" w:color="9BBB59" w:themeColor="accent3" w:fill="auto" />

<w:spacing>节点代表的是段落间距，即该段落的段前间距和段后间距，它具有如下表2-6所示属性：

表2-6 <w: spacing>节点元素主要属性取值举例及其含义

|  |  |  |
| --- | --- | --- |
| 属性 | 取值举例 | 含义 |
| w:before | 180 | 段前间距 |
| w:beforeLines | 50 | 以行为单位的段前间距 |
| w:after | 360 | 段后间距 |
| w:afterLines | 100 | 以行为单位的段后间距 |

<w:spacing>节点举例如下：

<w:spacing w:before="180" w:beforeLines="50" w:after="360" w:afterLines="100" />

w:beforeLines=50代表段前间距为0.5行，w:afterLines="100"代表段后间距为1行。

<w:p>节点的另一个子节点是<w:r>，对应Run对象，代表的是段落中的一块区域，该节点具有一个<w:rPr>节点，对应RunProperties对象，代表该区域的属性。<w:r>还有一个子节点是<w:t>，对应Text对象，代表段落中的文本，该节点的内容即为段落的文本内容。一个典型的段落结构如下所示：

<w:p>  
 <w:r>  
 <w:rPr>...</w:rPr>  
 <w:t>这是一段文本</w:t>  
 </w:r>  
</w:p>

<w:rPr>节点具有的主要子节点如下表2-7所示：

表2-7 <w:rPr>子节点元素与Open XML SDK类对应表

|  |  |  |
| --- | --- | --- |
| WordprocessingML元素 | Open XML SDK类 | 含义 |
| <w:b> | Bold | 粗体 |
| <w:i> | Italic | 斜体 |
| <w:color> | Color | 颜色 |
| <w:sz> | FontSize | 字体大小 |

当<w:rPr>具有<w:b>子节点时，代表该区域内的文本是粗体。当<w:rPr>具有<w:i>子节点时，代表该区域内的文本是斜体。<w:color>节点具有属性w:val，该值代表本文的颜色，如<w:color w:val="0000FF">代表文本的颜色为蓝色。<w:sz>节点具有属性w:val，该值代表文本的字体大小，如<w:sz w:val="21">代表文本的字体大小为21。

* + 1. **Table对象**

Table对象代表文档中的表格，是文档的主要组成部分之一，每个文档可由若干个表格组成。Table对象对应的OpenXML中的标签为<w:tbl>，一个典型的表格结构如下所示：

<w:tbl>

<w:tblPr>...</w:tblPr>

<w:tr>

<w:tc>...</w:tc>

<w:tc>...</w:tc>

</w:tr>

</w:tbl>

<w:tbl>具有一个子节点<w:tblPr>，对应TableProperties对象，代表的是表格的属性，该属性具有的几个主要子节点如下表2-8所示：

表2-8 <w tbl>子节点元素与Open XML SDK类对应表

|  |  |  |
| --- | --- | --- |
| WordprocessingML元素 | Open XML SDK类 | 含义 |
| <w:tblStyle> | TableStyle | 表格样式 |
| <w:shd> | Shading | 表格底纹 |
| <w:tblBorders> | TableBorders | 表格边框 |

<w:tblStyle>节点代表的是表格样式，它具有属性w:val，该属性的的取值代表了表格的内置样式，比如<w:tblStyle w:val="1-6"/>代表的是内置的“中等深浅网格1-强调文字颜色6”。

* 1. **Office Excel文档结构特点**

SpreadsheetML文档的文档结构包括<workbook>元素，后者包含引用工作簿中的工作表的<sheets>和<sheet>元素。将为每张工作表创建单独的XML文件。这些元素是有效电子表格文档所需的最小元素。此外，电子表格文档可能包含<table>、<chartsheet>、<pivotTableDefinition>或其他与电子表格相关的元素。

通过使用Open XML SDK，可以使用SpreadsheetML元素所对应的强类型类创建文档结构和内容。可以在 DocumentFormat.OpenXML.Spreadsheet 命名空间中找到这些类。下表2-9列出了一些重要电子表格元素所对应类的类名称：

表2-9 基本SpreadsheetML元素与Open XML SDK类对应表

|  |  |  |  |
| --- | --- | --- | --- |
| 包 | SpreadsheetML元素 | Open XML SDK 类 | 描述 |
| 工作簿 | <workbook> | Workbook | 主文档部件的根元素。 |
| 工作表 | <worksheet> | Worksheet | 表示包含文本、数字、日期或公式的单元格网格的工作表类型。 |
| 图表工作表 | <chartsheet> | Chartsheet | 表示存储为自己的工作表的图表的工作表。 |
| 表 | <table> | Table | 指定属于单个数据集的一系列数据的逻辑构造。 |
| 数据透视表 | <pivotTableDefinition> | PivotTableDefinition | 显示可理解布局中数据的聚合视图的逻辑构造。 |
| 透视缓存 | <pivotCacheDefinition> | PivotCacheDefinition | 定义数据透视表中的数据源的构造。 |
| 透视缓存记录 | <pivotCacheRecords> | PivotCacheRecords | 数据透视表的源数据的缓存。 |
| 计算链 | <calcChain> | CalculationChain | 指定上次计算工作簿中单元格的顺序的构造。 |
| 共享字符串表 | <sst> | SharedStringTable | 包含每个唯一字符串在工作簿中的所有工作表上的出现次数的构造。 |
| 条件格式 | <conditionalFormatting> | ConditionalFormatting | 定义应用于一个单元格或一系列单元格的格式的构造。 |
| 公式 | <f> | CellFormula | 定义包含公式的单元格的公式文本的构造。 |

典型工作簿可能包含数字、文本、图表、表和数据透视表，这些都包含在电子表格文档的 .zip 包中。下图2-2显示典型电子表格中可能包含的大多数元素：

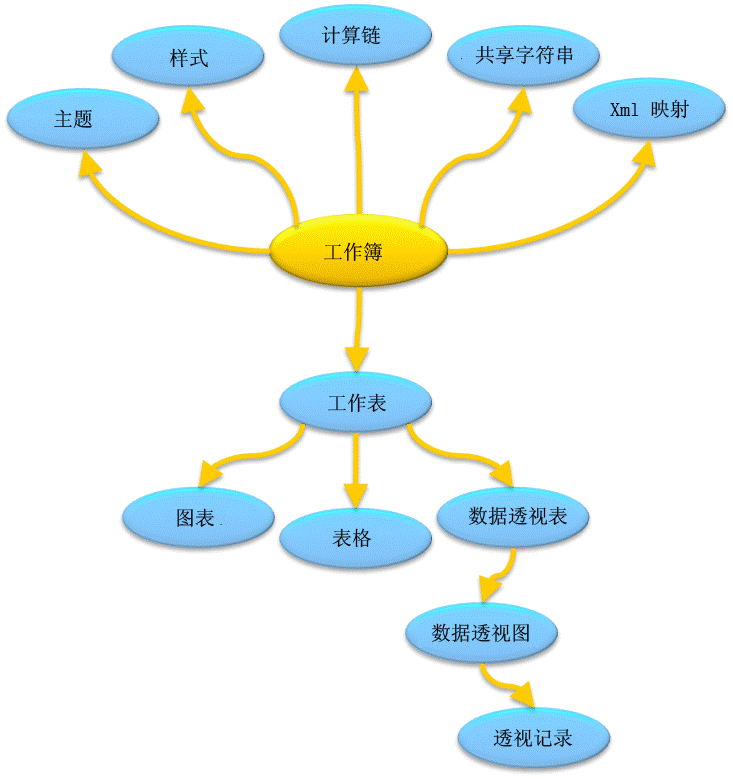


图2-2典型Office Excel文档结构

* + 1. **SpreadsheetML中的表格**

表格帮助组织工作表中的信息列表和向信息列表提供结构。表格具有明确标记的列、行和数据区域。通过表格，用户可以更轻松地排序、分析、格式化、管理、添加和删除信息。

如果将数据区域指定为表格，则可以应用帮助用户执行有用操作的特殊行为。如果用户在与表格底端相邻的行中键入其他数据，则表格可以扩展并自动将该数据添加到表格的数据区域。同样，添加列与在当前列标题的右侧或左侧键入新列标题一样简单。筛选和排序功能可以通过下拉箭头自动呈现给用户，以创建汇总或计算表格数据的特殊计算列，这些列能根据表格大小扩大和缩小，并且能够维护正确的公式引用。

可以基于以下几项创建表格：工作表中已经存在的数据、外部数据查询或者重复 XML 元素集合与工作表区域的映射。工作表 XML 存储数字和文本数据。表格 XML 记录特定表格对象的各个属性。SpreadsheetML 表格是一个逻辑构造，指定属于单个数据集的数据区域。SpreadsheetML 已经使用类似表格的模型指定行和列中的值，但是我们也可以将工作表的子集标记为 table 并向它提供有助于进行分析的某些属性。通过 SpreadsheetML 中的表格，我们可以使用新的方式分析数据，如使用数据筛选、格式化和绑定。与 SpreadsheetML 中的其他构造一样，工作表中的表格存储在包的单独部件中。表格部件不包含任何表格数据，数据在工作表单元格中进行维护。以下列出了在处理 Table 类时使用的常用 Open XML SDK类。

Table 类

Table 类表示在SpreadsheetML文档的Open XML文件格式架构中定义的表 (<table>) 元素。使用Table类可对SpreadsheetML文档中的各个 <table>元素进行操作。

表格部件包含单个表格的定义，如果一个工作表上存在多个表格，则存在多个表格部件。此部件的根元素是表格。从最低限度上讲，表格仅需要有关组成表格的表格列的信息。然而，若要启用自动筛选，则必须至少定义一个自动筛选器（可以为空）。如果不定义任何自动筛选器，则在 Excel 中打开文档时会禁用自动筛选。

table元素具有几个用于标识表格和表格覆盖的数据区域的属性。id 和 name 属性在所有表格部件中必须唯一。displayName 属性在所有表格部件中必须唯一，并且在工作簿的所有已定义名称中也必须唯一。Excel中的对象模型使用name属性，公式中的引用使用displayName属性。ref属性用于标识表格覆盖的单元格区域，这不仅包括表格数据，还包括含有列名称的表格标题。

TableColumn类

若要向表格中添加列，可以向tableColumns集合中添加新tableColumn元素。TableColumn (<tableColumn>) 元素表示此表格的单个列的元素。

AutoFilter 类

自动筛选基于筛选条件暂时隐藏行，筛选条件逐列应用到工作表中的数据表格。比如要描述一个筛选器，该筛选器指明"仅显示大于 0.5 的值"。筛选器将应用到 B3:E8 区域，条件将应用到 colId=1（基于零的列编号，从左向右递增）的列中的值。因此，如果任何行的该特定列中的值小于或等于 0.5，则必须隐藏该行。

<autoFilter ref="B3:E8">

   <filterColumn colId="1">

     <customFilters>

       <customFilter operator="greaterThan" val="0.5"/>

    </customFilters>

   </filterColumn>

</autoFilter>

* + 1. **SpreadsheetML 中的工作表**

工作表是工作簿内的核心结构，用户在工作表中执行大多数电子表格任务，最常见的工作表类型是由单元格网格表示的工作表。工作表单元格可以包含文本、数字、日期和公式，用户还可以设置单元格的格式，工作簿通常包含多个工作表。为了帮助分析数据和做出明智的决策，电子表格应用程序通常实现可帮助计算、排序、筛选、组织和以图形方式显示信息的功能和对象。因为这些功能通常与电子表格网格紧密关联，所以这些功能也包含在磁盘上的工作表定义中。

其他类型的工作表包括图表工作表和对话框工作表。

Worksheet 类

Worksheet 类表示在SpreadsheetML文档的Open XML文件格式架构中定义的 worksheet (<worksheet>) 元素。使用Worksheet类可以处理SpreadsheetML文档中的各个<worksheet>元素。

工作表部件类型的实例包含与给定工作表相关联的所有数据、公式和特征。对于每个工作表，包应该正好包含一个Worksheet部件。具体地说，sheet元素的ID属性应该引用所需的工作表部件，此内容类型的部件的根元素应该为 worksheet。可能的最小（空白）工作表如下所示：

<worksheet>

  <sheetData/>

</worksheet>

空sheetData集合表示一个空网格，此元素是必需的。根据架构中的定义，一些可选的工作表属性集合可以出现在sheetData之前，一些可以出现在sheetData之后。为了简化在现有工作表中插入新sheetData集合所需的逻辑，sheetData集合是必需的，即使对于空表也是如此。

典型的电子表格至少包含一个工作表，工作表包含一个类似于数据定义结构的表格，由sheetData元素表示，包含数据的工作表使用worksheet元素作为根元素来定义工作表。在工作表中，数据分为三个不同的部分，第一部分包含可选的表属性；第二部分包含数据，它使用必需的 sheetData 元素；第三部分包含可选的支持功能，如工作表保护和筛选信息。我们只需使用worksheet和sheetData元素即可定义空工作表。sheetData元素可以为空。

若要为工作表创建新值，需要在sheetData元素中定义行。这些行包含具有值的单元格，row元素定义新行。通常，sheetData中的第一行是可见工作表中的第一行。在行中，我们可以使用<c>元素创建新单元格。单元格的值可以通过在单元格中存储<v>元素来提供。通常，<v>元素包含工作表单元格的当前值。如果值是数字值，则直接存储在 XML文件的<v>元素中。如果值是字符串值，则存储在共享字符串表格中。

以下列出了在处理Worksheet类时常使用的Open XML SDK类。

SheetData类

sheet data (<sheetData>) 元素是工作表的核心结构，它包含网格中的所有文本、数字和公式。

Row 类

Row类对应着Open XML文档中的row (<row>) 元素。单元格表格中的单元格按行排列，每行都具有一个索引（属性 r），因此无需写出空行。每行都指示为它定义的单元格数目以及单元格在工作表中的相对位置。

Cell 类

Cell类对应着Open XML文档中的cell (<c>) 元素。单元格本身由c表示。每个单元格都使用A1样式的引用表示法指明它在网格中的位置。单元格还可以指明样式标识符（属性 s）和数据类型（属性 t）。单元格类型包括string、number和Boolean。为了优化加载/保存操作，不写出默认数据值。

CellValue 类

CellValue类对应着Open XML文档中的 cell value (<v>) 元素。单元格包含值，无论值是直接输入的值还是计算结果。单元格中的字符串值不存储在单元格表格中，除非它们是计算结果。

Chartsheet 类

此部件类型的实例表示一个存储在它自己的工作表中的图表，允许包中包含零个或多个Chartsheet部件。例如：sheet1.xml 引用一个图形，该图形是Chartsheet部件的关系项目中关系的目标：

<chartsheet xmlns:r="…" …>

    <sheetViews>

        <sheetView scale="64"/>

    </sheetViews>

    <drawing r:id="rId1"/>

</chartsheet>

以下列出了在处理Chartsheet类时使用的常用Open XML SDK类：

Drawing类

此部件类型的实例包含此工作表中存在的一个或多个图形元素的呈现方式和布局信息。允许包中包含一个或多个Drawings部件，每个此类部件都应该是Worksheet部件或Chartsheet中显式关系的目标。每个工作表或图表工作表应只有一个Drawings部件。

Dialogsheet 类

此部件类型的实例包含有关用户表单的旧自定义对话框的信息。允许包中包含一个或多个Dialogsheet部件，此内容类型的部件的根元素应为dialogsheet。例如：sheet1.xml 包含以下内容：

<dialogsheet xmlns:r="…" …>

    <sheetPr>

        <pageSetUpPr/>

    </sheetPr>

    <sheetViews>

        …

    </sheetViews>

    …

    <legacyDrawing r:id="rId1"/>

</dialogsheet>

* + 1. **SpreadsheetML 中的数据透视表**

以下信息介绍了PivotTableDefinition (<pivotTableDefinition>) 元素。

数据透视表能够以易于理解的布局轻松显示数据的聚合视图。成百上千条基础信息可以聚合在行和列轴上，以揭示数据背后的含义。数据透视表报表用于以不同方式组织和汇总数据。创建数据透视表报表是指四处移动信息以查看它们的内在关系。通过几次移动，即可将数据透视行和列转变为其他排列方式和布局。

数据透视表对象具有行轴区域、列轴区域、值区域和报表筛选区域。此外，数据透视表还具有相应的字段列表窗格，该窗格显示可以置于某个数据透视表区域中的所有数据字段。工作簿指向并始终拥有pivotCacheDefinition部件，后者指向并拥有 pivotCacheRecords部件。工作簿还指向并拥有工作表部件，并且当工作表上具有数据透视表（一个工作表上可以有多个数据透视表）时，工作表部件会指向并拥有pivotTable部件定义。pivotTable部件指向它使用的相应pivotCacheDefinition。因为多个数据透视表可以使用同一缓存，所以pivotTable部件并不始终拥有 pivotCacheDefinition。

pivotTable部件描述工作表上数据透视表的布局的详细信息。它指明哪些字段位于数据透视表的行轴、列轴、报表筛选和值区域。它还指明有关数据透视表的格式信息。如果对数据透视表应用了条件格式，则该信息也在 pivotTable 部件中表示。

数据透视缓存定义包含数据透视表中所有字段的定义。如果以普通表为基础创建数据透视表，则表中的每列都会成为数据透视缓存定义中的一个字段。数据透视缓存包含字段定义，以及有关该字段中包含的内容类型的信息。它还维护对缓存标记中源数据的引用，以便数据透视缓存可以随数据透视缓存记录部件中缓存的数据一起刷新。

显示在数据透视表中的数据存储在两个位置，数据透视缓存记录部件维护数据透视表的实际数据，工作表中的表单元格也存储数据的缓存版本，但是仅用于显示目的。数据透视缓存记录部件通过两种方式之一存储数据。数据透视表的数据区域的唯一值以内嵌方式缓存，通常存在于行和列中的重复项目以引用方式缓存，此共享数据实际上存储在数据透视缓存定义中。数据透视缓存记录部件中的每个记录都由N个值组成，其中N等于在数据透视缓存定义中定义的字段数。

最后一步是创建数据透视表本身，数据透视表定义部件包含有关数据透视表的哪个位置存在哪个字段的信息，我们可以将字段放置在四个区域：行、列、数据或筛选区域。字段从数据透视缓存定义中缓存的字段中选择。

若要创建在打开工作簿时已可供使用的数据透视表，还需要为表单元格创建标记。数据透视表显示在工作表的单元格中，因此我们还需要构造单元格。我们还可以让用户在打开文档时更新数据透视表单元格。

下表2-10列出了在处理PivotTableDefinition类时常用的Open XML SDK类。

表2-10 PivotTableDefinition相关元素与Open XML SDK类对应表

|  |  |
| --- | --- |
| SpreadsheetML元素 | Open XML SDK类 |
| pivotField | PivotField |
| pivotCacheDefinition | PivotCacheDefinition |
| pivotCacheRecords | PivotCacheRecords |

PivotTableDefinition 类

PivotTableDefinition类表示在Open XML文件格式架构中为SpreadsheetML文档定义的数据透视表定义 (<pivotTableDefinition>) 元素。使用 PivotTableDefinition类可以处理SpreadsheetML文档中的各个<pivotTableDefinition>元素。

数据透视表定义的主要功能是存储有关哪个字段按什么顺序显示在数据透视表的哪个轴上的信息。可以向数据透视表定义中添加许多其他设置，但下面只介绍一些基本设置。

根元素用于命名数据透视表，以便它可以用作数据源。根元素还使用添加到工作簿部件的ID来引用数据透视缓存，并且定义显示在数据透视表的数据区域上方的标题标签。所有这些元素都是必需的。

pivotTableDefinition的三个主要部分为：表的位置，缓存字段的显示信息和缓存字段的位置信息。

PivotField 类

PivotTableDefinition元素包含PivotField (<pivotField>)元素，表示数据透视表中的单个字段，此元素包含有关字段的信息，其中包括字段中的项目集合。

首先，使用pivotFields元素定义显示在数据透视表上的字段集合。每个字段都充当该字段在数据源中的数据的缓存，无需定义缓存，实际上，我们可以将item元素设置为default，并让用户在打开文档时更新表。showAll属性用于隐藏该数据维度的某些元素，定义表中包含哪些字段后，字段将应用于数据透视表的四个区域。

PivotCacheDefinition 类

pivotCacheDefinition 部件定义pivotCacheRecords部件中的每个字段，包括字段名称以及有关字段中包含的数据的信息。pivotCacheDefinition部件还定义在pivotTable和 pivotRecords部件之间共享的数据透视项目。

数据透视缓存定义数据透视表中数据的来源，以允许更新缓存，并定义该数据中的字段列表。需要注意的是，缓存可定义适用于数据透视表的所有字段，而不仅仅是实际使用的字段。数据透视表定义会定义特定数据透视表使用哪些可用字段。

数据源定义引用数据透视表中显示的数据，数据透视表还维护缓存记录部件中的数据，以便在数据连接不可用时表仍可更新。不能依赖数据透视表的单元格来存储数据，因为这些单元格中的数据从本质上讲是临时数据，在我们透视相应的表时，这些数据会更改。有多种类型的数据源，例如工作表、数据库、OLAP 多维数据集和其他数据透视表。

缓存定义的最后一部分使用cacheField元素来定义数据源的字段。cacheField 元素有两个用途：定义字段的数据类型和格式，以及用作共享字符串的缓存。数据透视值存储在数据透视缓存记录部件中。当将某个重复字符串用作值时，缓存记录使用对共享项目的cacheField 集合的引用。

PivotCacheRecords类

pivotCacheRecords 部件包含要聚合的基础数据。它是源数据的缓存。缓存记录部件可以存储任意数目的缓存记录。每个记录定义的值的数目都与缓存定义中的字段数相同。每个记录都使用r元素定义。此记录包含以子元素形式存在的值项目。我们可以提供特定类型的值（如 Numeric、Boolean或Date-Time），也可以引用共享项目。

* + 1. **SpreadsheetML中的公式**

可使用公式来创建与计算相关的模型。通过公式，可根据电子表格中和电子表格外的数据或电子表格中的其他已计算单元格的输出来自动计算值。在工作表XML文件中，公式存储在使用公式的每个单元格中。使用CellFormula (<f>)元素可定义公式文本。公式可包含包括大量预定义函数的数学表达式。

CellValue (<v>)元素根据最后一次计算公式的时间来存储缓存的公式值。这样，用户便可在打开电子表格时延期计算公式值，从而在打开工作表时节省时间。我们无需指定值，并且如果我们省略了值，则Open XML读者负责在打开工作表时根据公式定义来计算值。

SpreadsheetML 公式是电子表格应用程序将一系列计算解析或解释为根据零到多输入计算一个值或一组值的函数的语法表示形式。公式是可包含以下内容的表达式：常量、运算符、单元格引用、函数调用和名称。

例如：考虑公式PI()\*(A2^2)。在这种情况下，PI()会产生对函数PI的调用，并返回π的值。单元格引用A2返回该单元格中的值。2是数字常量。插字号 (^)运算符计算其左侧操作数的右侧操作数次幂。圆括号 ( 和 )用于分组。星号 (\*)运算符执行两个操作数的乘法运算。

运算符是指定要对一个或多个操作数执行的运算类型的符号。包括算术、比较、文本和引用运算符。

工作表中的每组水平单元格为一行，每组垂直单元格为一列。单元格的行和列组合指定该单元格的位置。

单元格引用指定同一工作表上的一个或多个单元格。通过引用，可以：

* 在一个公式中使用同一工作表的不同部分中包含的数据。
* 在多个公式中使用一个单元格中的值。
* 引用同一工作簿（甚至其他工作簿）中其他工作表上的单元格。（引用其他工作簿中的单元格称为链接。）

函数是采用零或多个参数，执行运算并返回结果（可选）的命名公式。以下是函数调用的一些示例：PI()、POWER(A1,B3)和SUM(C6:C10)。

Office Open XML规范定义了300多个预定义函数。还允许使用用户定义的函数。

名称是常量、单元格引用或公式的别名。通过公式中的名称，更易于了解该公式的用途。例如，公式 SUM(FirstQuarterSales)比SUM(C20:C30) 更容易识别。

每个表达式都具有一种类型。SpreadsheetML公式支持以下类型：数组、错误、逻辑、数字和文本。

数组值或常量表示一个或多个元素的集合，这些元素的值可为任意类型（即，数组的元素不必都为同一类型）。

* + 1. **SpreadsheetML中的条件格式**

基于单元格的条件格式可为工作表内的数据提供结构。在显示值的同时显示颜色有助于区分这些值的相对高度。可以基于单元格的值对单元格应用多个格式选项。可以突出显示 排在顶部或底部的项、提供数据栏来显示进度条类型用户界面，或者使用色阶指示高值和低值。条件格式可直接应用于工作表中的单元格。值不一定为表的一部分。

所有条件格式设置存储在工作表级别。工作表会为应用于单元格或单元格序列的每个格式存储一个<conditionalFormatting>元素。应用了格式的单元格集合是使用 sqref属性来定义的。sqref属性可使用"from:to"表示法指定单元格区域，例如"A1:A10"。

下表2-11列出了在处理 ConditionalFormatting 类时常用的 Open XML SDK类。

表2-11 ConditionalFormatting相关元素与Open XML SDK类对应表

|  |  |
| --- | --- |
| SpreadsheetML 元素 | Open XML SDK 类 |
| cfRule | ConditionalFormattingRule |
| dataBar | DataBar |
| colorScale | ColorScale |
| iconSet | IconSet |

条件格式类

ConditionalFormatting类表示在Open XML文件格式架构中为 SpreadsheetML文档定义的表(<conditionalFormatting>)元素。使用ConditionalFormatting 类可对SpreadsheetML文档中的各个 <conditionalFormatting>元素执行操作。

条件格式是在指定条件成立时电子表格应用程序可以自动应用于单元格的格式（如单元格底纹或字体颜色）。此集合表示应用于特定单元格或区域的条件格式规则。 例如：此示例对单元格 C3:C8 应用"top10"规则。@dxfId 引用与条件匹配的单元格所要应用的格式（在样式部件中定义）。

<conditionalFormatting sqref="C3:C8">

<cfRule type="top10" dxfId="1" priority="3" rank="2"/>

</conditionalFormatting>

条件格式规则类

例如要演示可突出显示其值大于0.5的单元格的条件格式规则。需要注意的是，在此情况下，<formula> 的内容是静态值，但也可以是公式表达式。

<conditionalFormatting sqref="E3:E9">

<cfRule type="cellIs" dxfId="0" priority="1" operator="greaterThan">

<formula>0.5</formula>

</cfRule>

</conditionalFormatting>

只有具有表达式的类型属性值的规则才支持公式语法。

每个条件格式可以指定多种格式规则。例如，可以同时应用色阶和数据栏格式。每个条件格式用单独的<cfRule>元素来表示。若要指定其用户界面显示优先级，可以使用priority属性。由于<conditionalFormatting>元素可能与工作表中的其他格式区域重叠，因此对于为该工作表定义的所有条件格式而言，优先级是全局性的。

<cfRule>元素具有许多可应用的格式类型，如 cellIs 和 top10。每种格式类型都使用常用元素来定义其设置。

数据栏类

DataBar (<dataBar>) 元素描述数据栏条件格式规则。在下面示例中，表达了一个数据栏条件格式，该格式分布在单元格区域内的所有单元格值中，其颜色为蓝色。

<dataBar>

<cfvo type="min" val="0"/>

<cfvo type="max" val="0"/>

<color rgb="FF638EC6"/>

</dataBar>

可按如下方式计算任意单元格的数据栏长度：

数据栏长度 = 最小长度 + (单元格值 - 区域中的最小值) / (区域中的最大值 - 区域中的最小值) \* (最大长度 - 最小长度)，其中，最小长度和最大长度是列宽的固定百分比（默认情况下分别为 10% 和 90%。）

数据栏采用单一颜色并显示为条。条的长度指示单元格值的相对高度。数据栏使用条件格式规则内的单独模型定义其设置。<dataBar>元素存储所有相关数据。数据栏需要三个设置：与单元格值进行比较的最小值和最大值以及颜色。第一个<cfvo>元素（即条件格式值对象）定义最小值，第二个<cfvo>元素定义最大值。可以使用不同方式指定值，如使用公式或硬编码值。另一种常用的选择方案是使用"min"和"max"类型。这些 <cfvo>元素类型指定在单元格区域中找到的已应用格式的最小值和最大值。这样可以在最低项和最高项之间提供清晰的阶梯式渐变。此外，还可以使用<color>元素指定数据栏的颜色。

色阶类

ColorScale (<colorScale>) 元素描述条件格式规则中的渐变色阶。例如：

<colorScale>

<cfvo type="min" val="0"/>

<cfvo type="max" val="0"/>

<color theme="5"/>

<color rgb="FFFFEF9C"/>

</colorScale>

色阶提供的显示可指示所有单元格项之间的相对值，这类似于数据栏。色阶使用条件格式规则中的单独模型定义其设置。最多可以指定三个<cfvo>（即条件格式值对象）元素值：一个用于色阶的开始、一个用于色阶的中间、一个用于色阶的结尾。中间值是可选的。此外，还可以使用<color>元素指定色阶的颜色。

图标集类

IconSet (<iconSet>) 元素描述图标集条件格式规则。例如：

要演示"3Arrows"样式的图标。如果单元格的值小于第33百分位数，则必须显示集合中的第一个图标。如果单元格的值小于第67百分位数并大于或等于第33百分位数，则必须显示集合中的第二个图标。如果单元格的值大于或等于第67百分位数，则必须显示集合中的第三个图标。

<iconSet iconSet="3Arrows">

<cfvo type="percentile" val="0"/>

<cfvo type="percentile" val="33"/>

<cfvo type="percentile" val="67"/>

</iconSet>

使用图标集可以对包含数据的单元格应用不同的图标集合。图标集使用一系列值标识要对哪一组单元格应用格式规则。第一个<cfvo>元素标识该范围的最低值，第二个 <cfvo>元素标识中间点，第三个<cfvo>元素标识最高值。图标集标识要应用于单元格的图标。可以从各种硬编码图标中进行选择。

* 1. **Office PowerPoint 文档结构特点**

PresentationML文档的文档结构由<presentation>（演示文稿）元素（包含 <sldMaster>（幻灯片母版）、<sldLayout>（幻灯片版式）、<sld >（幻灯片））和引用演示文稿中的幻灯片的 <theme>（主题）元素组成。（主题元素是DrawingMLTheme部件的根元素。）这些元素是有效的演示文稿文档至少要包含的元素。此外，演示文稿文档还可能包含<notes>（备注幻灯片）、<handoutMaster>（讲义母版）、<sp>（形状）、<pic>（图片）、<tbl>（表格）和其他与幻灯片相关的元素。（表格元素在DrawingML架构中定义。）

一个PresentationML文档可以包含的其他功能如下：动画、音频、视频以及幻灯片之间的切换。PresentationML文档不会存储为单个部件中的一个大型正文，而是实现某些功能组合的元素会存储在各个部件中。例如，文档中的所有注释存储在一个注释部件中，而每个幻灯片都有自己的部件，会为每个幻灯片创建一个单独的XML文件。

通过 Open XML SDK ，可以使用与PresentationML元素对应的强类型类创建文档结构和内容。可在[DocumentFormat.OpenXml.Presentation](https://msdn.microsoft.com/zh-cn/library/office/documentformat.openxml.presentation.aspx)命名空间中找到这些类。下表2-12列出了与一些重要的演示文稿元素对应的类的类名。

表2-12 基本PresentationML元素与Open XML SDK类对应表

|  |  |  |  |
| --- | --- | --- | --- |
| 包 | PresentationML元素 | Open XML SDK 类 | 描述 |
| 演示文稿 | <presentation> | Presentation | 演示文稿部件的根元素。此元素中会指定演示文稿范围的基本属性。 |
| 演示文稿属性 | <presentationPr> | PresentationProperties | 演示文稿属性部件的根元素。此元素用作父元素，其中包含演示文稿范围的其他文档属性。 |
| 幻灯片母版 | <sldMaster> | SlideMaster | 幻灯片母版部件的根元素。在幻灯片母版幻灯片中，包含描述演示文稿幻灯片中的对象及其对应格式的所有元素。 |
| 幻灯片版式 | <sldLayout> | SlideLayout | 幻灯片版式部件的根元素。此元素为幻灯片母版中使用的每个幻灯片版式指定关系信息。 |
| 主题 | <officeStyleSheet> | Theme | 主题部件的根元素。此元素保留通过主题可供文档使用的所有不同的格式选项并定义在文档中使用主题对象时文档的总体外观。 |
| 幻灯片 | <sld> | Slide | 幻灯片部件的根元素。此元素在幻灯片列表中指定幻灯片。 |
| 备注母版 | <notesMaster> | [NotesMaster](https://msdn.microsoft.com/zh-cn/library/office/documentformat.openxml.presentation.notesmaster.aspx) | 备注母版部件的根元素。在备注母版幻灯片中，包含描述备注幻灯片中的对象及其对应格式的所有元素。 |
| 备注幻灯片 | <notes> | NotesSlide | 备注幻灯片部件的根元素。此元素指定是否存在备注幻灯片及其对应数据。备注幻灯片包含所有常用的幻灯片元素以及特定于备注元素的添加属性。 |
| 讲义母版 | <handoutMaster> | HandoutMaster | 讲义母版部件的根元素。在讲义母版幻灯片中，包含描述讲义幻灯片中的对象及其对应格式的所有元素。 |
| 注释 | <cmLst> | CommentList | 注释部件的根元素。此元素为特定幻灯片指定注释列表。 |
| 注释作者 | <cmAuthorLst> | CommentAuthorList | 注释作者部件的根元素。此元素指定在当前文档中具有注释的作者的列表。 |

典型演示文稿可能包含多个幻灯片，每个幻灯片都引用幻灯片版式和幻灯片母版，还可能包含注释。此外，演示文稿还可能包含讲义和备注幻灯片，每个幻灯片都由单独的部件来表示。这些附加部件包含在演示文稿文档的 .zip 包中。下图2-3显示可在典型演示文稿中找到的大多数元素：

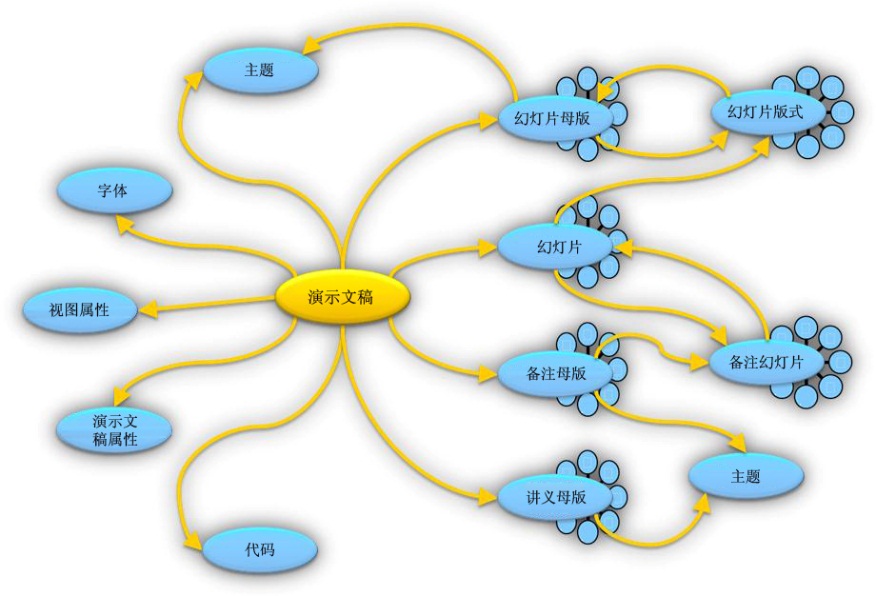


图2-3典型Office PPT文档结构

* + 1. **演示文稿部件**

PresentationML包的主要部件以<presentation>根元素开头。该元素包含演示文稿，演示文稿又引用幻灯片列表、幻灯片母版列表、备注母版列表和讲义母版列表。幻灯片列表引用演示文稿中的所有幻灯片。幻灯片母版列表引用演示文稿中使用的整套幻灯片母版。备注母版包含有关备注页格式的信息。讲义母版描述讲义的外观。（讲义是打印出来的一组幻灯片，可分发给访问群体，供他们将来参考。）

* + 1. **演示文稿属性部件**

演示文稿属性部件的根元素为<presentationPr>元素。此部件类型的实例包含演示文稿的所有属性。一个包恰好包含一个演示文稿属性部件，该部件应是演示文稿部件中的隐式关系的目标。

例如：下面的演示文稿部件关系项包含与演示文稿属性部件的关系，该部件存储在ZIP项presProps.xml中：

<Relationships xmlns="…">  
    <Relationship Id="rId6"  
        Type="http://…/presProps" Target="presProps.xml"/>  
 </Relationships>

此内容类型的部件的根元素应为presentationPr。

例如：

<p:presentationPr xmlns:p="…" …>  
    <p:clrMru>  
        …  
    </p:clrMru>  
    …  
</p:presentationPr>

演示文稿属性部件位于包含关系部件的包中（从句法的角度表达，Relationship元素的TargetMode属性应为Internal）。

演示文稿属性部件与ISO/IEC 29500定义的任何其他部件之间都不应存在隐式或显式关系。

* + 1. **幻灯片母板部件**

幻灯片母版部件的根元素为<sldMaster>元素。此部件类型的实例包含由此幻灯片母版派生的演示文稿中的每个幻灯片上显示的格式、文本和对象的母版定义。一个包应包含一个或多个幻灯片母版部件，每个部件应为以下关系的目标：来自演示文稿部件的一个显式关系，以及来自根据此幻灯片母版定义任何幻灯片版式的该幻灯片版式部件的一个隐式关系。每个部件也可以选择作为幻灯片版式部件中的一个关系的目标。

例如：下面的演示文稿部件关系项包含与幻灯片母版部件的关系，该部件存储在 ZIP项slideMasters/slideMaster1.xml中：

<Relationships xmlns="…">  
    <Relationship Id="rId1"  
        Type="http://…/slideMaster"   
 Target="slideMasters/slideMaster1.xml"/>  
</Relationships>

此内容类型的部件的根元素应为sldMaster。

例如：

<p:sldMaster xmlns:p="…">  
  <p:cSld name="">  
      …  
  </p:cSld>  
  <p:clrMap … />  
</p:sldMaster>

* + 1. **幻灯片版式部件**

幻灯片版式部件的根元素为<sldLayout>元素。此部件类型的实例包含用于此演示文稿的幻灯片版式模板的定义。此模板定义创建幻灯片时绘制对象在此幻灯片类型中的默认外观和位置。一个包应包含一个或多个幻灯片版式部件，每个部件应为以下关系的目标：幻灯片母版部件中的一个显式关系，以及来自与此幻灯片版式关联的每个幻灯片部件的一个隐式关系。

例如：下面的幻灯片母版部件关系项包含与多个幻灯片版式部件的关系，这些部件存储在 ZIP 项 ../slideLayouts/slideLayoutN.xml 中：

<Relationships xmlns="…">  
    <Relationship Id="rId1"  
        Type="http://…/slideLayout"  
        Target="../slideLayouts/slideLayout1.xml"/>  
    <Relationship Id="rId2"  
        Type="http://…/slideLayout"  
        Target="../slideLayouts/slideLayout2.xml"/>  
    <Relationship Id="rId3"  
        Type="http://…/slideLayout"  
        Target="../slideLayouts/slideLayout3.xml"/>  
</Relationships>

此内容类型的部件的根元素应为 sldLayout。

例如：

<p:sldLayout xmlns:p="…" matchingName="" type="title" preserve="1">  
   <p:cSld name="Title Slide">  
       …  
   </p:cSld>  
   <p:clrMapOvr>  
      <a:masterClrMapping/>  
   </p:clrMapOvr>  
   <p:timing/>  
</p:sldLayout>

* + 1. **幻灯片部件**

幻灯片部件的根元素为<sld>元素。除了文本和图形，每个幻灯片还可以包含注释和备注，可以有布局，可以是一个或多个自定义演示文稿的组成部件。注释是供维护演示文稿幻灯片平台的人员参考的批注。备注是供演示者或访问群体参考的提醒信息或一段文字。幻灯片部件包含单个幻灯片的内容，一个包应包含每个幻灯片的一个幻灯片部件，每个部件应为来自演示文稿部件的一个显式关系的目标。

例如：假设一个 PresentationML文档有两个幻灯片。对应的演示文稿部件关系项包含两个与幻灯片部件的关系，这些部件存储在ZIP项slides/slide1.xml和 slides/slide2.xml 中：

<Relationships xmlns="…">  
    <Relationship Id="rId2"  
        Type="http://…/slide" Target="slides/slide1.xml"/>  
    <Relationship Id="rId3"  
        Type="http://…/slide" Target="slides/slide2.xml"/>  
 </Relationships>

此内容类型的部件的根元素应为sld。

例如：slides/slide1.xml 包含：

<p:sld xmlns:p="…">  
  <p:cSld name="">  
    …  
  </p:cSld>  
  <p:clrMapOvr>  
    …  
  </p:clrMapOvr>  
  <p:timing>  
   <p:tnLst>  
   <p:par>  
     <p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot"/>  
    </p:par>  
   </p:tnLst>  
  </p:timing>  
</p:sld>

* 1. **本章小结**

本章主要介绍了Open XML的基本概念以及Office文档的结构特点，包含Office Word文档结构特点，Office Excel文档结构特点以及Office Power Point文档结构特点。Office文档结构非常复杂，本章只是简单介绍了一下Office文档中常见的元素以及文档中的几个核心部件。对Office文档结构的了解是利用Open XML SDK获取文档信息的基础，下一章将会论述如何通过Open XML SDK来对Office文档的信息进行获取。

# Office文档信息获取方法

* 1. **Open XML SDK概述**

Open XML SDK是微软的一个开源项目，为开发者提供了一套功能强大的类库，使开发者能够轻松读、写和使用诸如通过Microsoft Office创建的docx、xlsx或pptx等等各类Open XML文档中的内容。

* + 1. **Open XML SDK 1.0**

Open XML SDK版本1简化了对Open XML包的操作。Open XML SDK应用程序编程接口(API)封装我们对Open XML包所执行的许多常见任务，以便我们只使用少数几行代码就可以执行复杂的操作。一些常见任务有：

* + 搜索——通过几行代码，我们可以在一组Excel 2007工作表中搜索某些任意数据。
  + 文档合并——我们可以通过编程方式合并现有文档的文档部件来创建文档。例如，我们可以从各种 PowerPoint 2007 演示文稿中提取幻灯片来创建单个演示文稿。
  + 验证——通过几行代码，我们可以验证包中的文档部件，也可以对照架构验证整个包。
  + 数据更新——使用 Open XML 对象模型，我们可以轻松修改多个包中的数据。
  + 隐私——通过几行代码，我们可以在文档发布之前，从中移除注释和其他个人信息。
    1. **Open XML SDK 2.0**

Open XML SDK 2.0 将强类型类支持从版本1.0中提供的部件类扩展到每个部件中的XML内容，仍支持版本1.0中提供的所有函数。凭借版本2.0，我们能够针对部件内部的XML内容进行编程。SDK支持采用LINQ to XML样式进行编程，与传统的W3C XML DOM编程模型相比，此编程方式使针对XML内容进行编码更加容易。SDK 2.0支持以下常见任务/方案：

* 强类型类和对象——不再需要依赖一般XML功能操作XML（这需要我们了解元素/属性/值拼写以及命名空间），通过使用Open XML SDK，我们只需操作表示元素/属性/值的对象即可实现相同的解决方案。所有架构类型都表示为强类型的公共语言运行时 (CLR) 类，所有属性值都表示为枚举。
* 内容构造、搜索和操作——LINQ 技术直接内置到SDK中。因此，我们可以直接对表示Open XML元素的对象执行功能构造和lambda表达式查询。此外，SDK还提供对对象集合（如表和段落）的支持，从而允许我们轻松地遍历和操作内容。
* 验证——Open XML SDK 2.0 for Microsoft Office提供验证功能，从而使我们能够对照Open XML格式的不同验证方式来验证Open XML文档。
  + 1. **Open XML SDK 2.5**

Open XML SDK 2.5 提供支持 Microsoft Office 2013 的命名空间和成员。Open XML SDK 2.5还可读取ISO/IEC 29500严格格式文件。严格格式是不包括旧功能的过渡格式的子集——在理论上，这使得新的实施者更容易提供支持，因为该格式具有较小的技术涉及面。SDK 支持以下常见任务/方案：

* 支持Office 2013 Preview文件格式——除了Microsoft Office类的Open XML SDK 2.0，Open XML SDK 2.5还提供了新的类，利用这些类，我们可以编写和生成用于处理新的Office 2013功能的Open XML文件扩展名的应用程序。
* 读取ISO严格文档文件——Open XML SDK 2.5可读取ISO/IEC 29500严格格式文件。当Open XML SDK 2.5 API打开严格格式文件时，通过将 http://purl.oclc.org/ooxml/命名空间映射到对应的 http://schemas.openxmlformats.org/命名空间，文件中的每个Open XML部件被加载到Open XML SDK 2.5的OpenXmlPart类。
* 修复 Open XML SDK 2.0 for Microsoft Office——Open XML SDK 2.5包括在 Open XML SDK 2.0 for Microsoft Office中对已知问题的修复。这些已知问题包括PowerPoint演示文稿中丢失的空白和Word文档（其中指定参数被报告为超出有效值范围）中有关自定义 UI 的问题。
  1. **Word文档信息获取**

在Open Xml SDK中，WordprocessingDocument类代表了Word 文档中的一个包。要处理一份Word文档，首先需要通过文档创建一个WordprocessingDocument类的实例，然后操作这个实例从而实现对Word文档的操作。获取该实例之后，我们就能够访问包含了文本的主文档部分。每一个打开的XML包都包含若干个部分，一个WordprocessingDocument至少包含了一个主文档部分，该部分作为一个容器用来容纳Word文档中的主要文本内容，该WordprocessingDocument所代表的包也可以包含其他的一些部分。在一个Word文档中，主文档部分中的文本内容在XML包中使用WordprocessingML来标记。

要从一份Word文档创建WordprocessingDocument类的实例，我们可以调用Open（string，bool）方法，Open XML SDK中提供了好几个Open方法，我们采用具有2个参数的Open方法来获取实例，第一个参数代表一个文档路径，该路径关联了我们想要打开的Word文档，第二个参数可以取值为true或者false，代表我们打开的文档是否是可编辑的。在我们的应用当中，由于对Word文档都是只读的，所以该参数均取值为false，表示不能修改打开的Word文档。创建WordprocessingDocument类实例的示例代码如下：

WordprocessingDocument document = WordprocessingDocument.Open(path, false);

要访问主文档部分的body，我们可以把存在的文档体赋值给Body引用，如下代码所示：

Body body = document.MainDocumentPart.Document.Body;

* + 1. **获取所有段落**

可以通过body的Descendants<Paragraph>()方法来得到<body>标签内所有段落的迭代器，之后通过这个迭代器来遍历文档中的所有段落。示例代码如下：

IEnumerator<Paragraph> paragraphs = body.Descendants<Paragraph>().GetEnumerator();

while (paragraphs.MoveNext())

{

Paragraph paragraph = paragraphs.Current;

HandleParagraph(paragraph);

}

* + 1. **获取某个段落的边框**

段落边框是段落属性节点的子节点，我们可以先获取段落属性节点，然后再获取段落边框节点，示例代码如下：

ParagraphProperties pPro = paragraph.ParagraphProperties;

ParagraphBorders borders = pPro.ParagraphBorders;

要获取上边框，可以如下操作：

TopBorder top = borders.TopBorder;

其余下、左、右边框操作类似。

* + 1. **获取边框属性值**

段落的上下左右四边框均包含各自的属性，获取这些属性值的方法如下。我们以获取上边框的大小、颜色、类型为例：

获取上边框大小：string size = top.Size;

获取上边框颜色：string color = top.Color;

获取上边框类型：string type = top.Val;

* 1. **Excel文档信息获取**

在 Open XML SDK中，SpreadsheetDocument类表示Excel文档包。若要创建Excel文档，需要创建SpreadsheetDocument类的一个实例并用部件填充该实例。该文档至少必须具有一个充当文档容器的工作簿部件和至少一个工作表部件。在此文档包中，将使用SpreadsheetML标记将文本表示为XML形式。若要从文档中创建类实例，我们可以调用Open（string，bool）方法，Open XML SDK中提供了好几个Open方法，我们采用具有2个参数的Open方法来获取实例，第一个参数代表一个文档路径，该路径关联了我们想要打开的Excel文档，第二个参数可以取值为true或者false，代表我们打开的文档是否是可编辑的。创建SpreadsheetDocument类实例的示例代码如下：

SpreadsheetDocument document = SpreadsheetDocument.Open(path, false);

打开电子表格文档包后，即可访问主工作簿部件。若要访问主工作簿部件，需要获取一个对现有工作簿部件的引用，如下代码示例中所示：

WorkbookPart wbPart = document.WorkbookPart;

* + 1. **获取指定名称的Sheet**

Sheet theSheet = wbPart.Workbook.Descendants<Sheet>().

Where(s => s.Name == sheetName).FirstOrDefault();

FirstOrDefault方法会返回第一个匹配的引用（在本例中为工作表）或空引用（如果未找到匹配）。如果传入无效工作表名称，则该代码会检查空引用并引发异常。现在，我们已经拥有工作表的相关信息，该工作表信息将提供Id属性，而提供Id属性后，就可以获取对相应WorksheetPart的引用，方法是调用WorkbookPart的GetPartById方法，如下所示：

WorksheetPart wsPart =

(WorksheetPart)(wbPart.GetPartById(theSheet.Id));

* + 1. **查找已命名单元格**

和查找已命名工作表一样，当查找已命名单元格时，该代码会使用Descendants方法，搜索CellReference属性等于指定addressName参数的第一个匹配。在该方法调用后，名为theCell的变量将包含对单元格的引用或将包含空引用，如下所示：

Cell theCell = wsPart.Worksheet.Descendants<Cell>().

Where(c => c.CellReference == addressName).FirstOrDefault();

* + 1. **获取单元格的值**

经过上述操作之后，名为 theCell 的变量将包含一个空引用或包含对我们请求的单元格的引用。如果检查该单元格的Open XML内容（即theCell.OuterXml），将会发现与以下示例代码类似的 XML。

<x:c r="A1">

<x:v>12.345000000000001</x:v>

</x:c>

InnerText属性包含单元格的内容，这样一来，我们可以通过下面的方法检索到这个值：

if (theCell != null)

{

value = theCell.InnerText;

}

* 1. **PPT文档信息获取**

在Open XML SDK中，PresentationDocument类表示演示文稿文档包。若要处理演示文稿文档，需要首先创建PresentationDocument类的实例，然后处理该实例。若要从文档中创建类实例，可以使用PresentationDocument.Open(string, bool) 方法，第一个参数指定文档所在路径，并以一个布尔值作为第二个参数来指定文档是否可编辑。若要打开文档进行读/写访问，此参数应指定值true；若要进行只读访问，指定值false，示例代码如下所示：

PresentationDocument document = PresentationDocument.Open(path, false);

* + 1. **获取指定幻灯片**

根据幻灯片索引位置slideIndex获取指定幻灯片的核心代码如下所示：

PresentationPart presentationPart = document.PresentationPart;

Presentation presentation = presentationPart.Presentation;

var slideIds = presentation.SlideIdList.ChildElements;

string slidePartRelationshipId = (slideIds[slideIndex] as SlideId).RelationshipId;

SlidePart slidePart = (SlidePart)presentationPart.GetPartById(slidePartRelationshipId);

Slide slide = slidePart.Slide

* + 1. **获取幻灯片内所有段落文本**

遍历幻灯片内所有文本的示例代码如下所示：

foreach (var paragraph in slide.Descendants< Paragraph>())

{

foreach (var text in paragraph.Descendants< Text>())

{

HandleText(text);

}

}

* 1. **本章小结**

本章主要介绍了Office文档信息获取的办法，首先介绍了进行信息获取的工具——Open XML SDK，其次分三部分分别介绍了如何通过Open XML SDK获取Word文档信息，Excel文档信息以及PPT文档信息。这些文档信息获取的方法是构建自动批改系统的基础。

# 自动评阅系统中的一些策略研究

1. **考点类型划分策略**

根据对大量Word操作题的题干进行分析的结果，将Word操作题考点类型分为以下十一个大类：段落、艺术字、表格、文本、图片、分栏、首字下沉、SmartArt、文本位置、文档排版、制表位。其中，每个大类所覆盖的小类如下表格所示：

表4-1 考点大类及其所含小类

|  |  |
| --- | --- |
| 大类 | 所含小类 |
| 段落 | 边框类型、边框颜色、边框粗细、边框阴影、底纹 |
| 艺术字 | 形状样式、文字样式、位置 |
| 表格 | 表格样式、表格边框、表格底纹、表头对齐、表头字体、其他对齐、其他字体 |
| 文本 | 大小、颜色、加粗、删除线、下划线、斜体、字体、对齐方式 |
| 图片 | 图片大小、图片样式 |
| 分栏 | 分栏数、分隔线、间隔宽度 |
| 首字下沉 | 下沉/悬挂、下沉行数、距正文 |
| SmartArt | 布局、颜色、样式、文字 |
| 文本位置 | 提升/降低 |
| 文档排版 | 段落首行缩进、段落间距 |
| 制表位 | 字符数、对齐方式、前导符 |

在考点的类型划分上面，若划分粒度过大，则会使批改操作产生较大误差，若划分粒度过小，则会给系统的开发以及使用带来较大的复杂性，因此在该系统中采取了一种适中的粒度，既能防止粒度过大导致的较大误差，又能减轻粒度过小所带来的复杂性。以“制表位”这个大类来进行说明：

若考点划分成“制表位”这个一个大类，则对于该考点，要么得满分，要么得零分，在这个考点上无法进行更细致的划分，因此会产生较大误差。若将制表位的对齐方式考点再细化成“左对齐”、“居中”、“右对齐”、“小数点对齐”、“竖线对齐”，将前导符划分成“无前导符”、“…..”、“——”、“\_\_\_\_”、“……”这五种前导符类型，虽然在批改上面可以达到非常精确的匹配，但是在系统的实现上，以及配置文件的设置上则会大大增加复杂度。

下图（4-1）~（4-2）显示了制表位粒度适中、粒度过细在配置文件中的体现，我们可以直观的看出这2种粒度的复杂度：粒度过细的复杂度是适中粒度的将近4倍。要准确无误地配置出来这样一个细粒度配置文件，对系统的使用者来说是一个极大的挑战，与要减轻老师批改负担这一目标不符，鉴于此，在该系统的实现上，采用了适中粒度的考点类型划分策略。

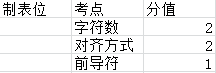
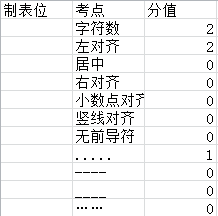
 

图4- 1 粒度适中 图4- 2 粒度过细

1. **考点位置定位策略**

策略一：根据字符串进行定位

对于考点涉及到具体文本的情况，可采用该文本或者文本中的一部分作为定位信息。比如对于考题“给第3段增加上下的双线边框和浅蓝色底纹。”我们可以截取第三段中的部分文本作为定位信息，若第三段的文本不多，也可以考虑将整个第三段的文本作为定位信息，并将该定位信息配置在考点配置文件中，在进行批改的时候根据这段文字对考点进行定位。

上述这种定位策略存在一个定位文本的长度该如何确定的问题。文本太短则有可能不能实现唯一定位，文本太长则万一学生不小心删除了该段文本中的某一个或几个字就会导致定位不到该考点而被判零分。针对这个问题，有一条指导原则就是：在能唯一确定考点的前提下，定位文本应尽可能短。

策略二：根据文档标签进行定位

对于有些考点，则可以根据其所涉及的文档标签进行定位。比如对于考题“表格样式采用“中等深浅网格1 – 强调文字颜色6”样式。”我们可以在文档中搜索<tbl>标签进行定位。这种定位策略也存在一个问题，即如果整篇文档中存在不止一个表格，那如何判断当前定位到的表格是否就是我们要考查的表格。对于这个问题，一种方法是限制每次考试的文档中最多只能出现一个表格，另一种解决办法是：与文本定位策略相结合，除<tbl>标签外，再选取表格中的某段文本（如表格标题）进行定位。

策略三：对于PPT的考点来说，可以根据幻灯片的ID进行定位。

在PPT文档中，可以考虑直接在配置文件中指定要批改的是第几张幻灯片，然后直接定位到该幻灯片。同时也可以结合前面的定位策略，定位到某张幻灯片中的更具体的考点。

策略四：对于Excel的考点来说，可以根据行和列进行定位。

由于Excel文档本身就对行和列进行了很好的划分，对于要批改的考点，可以在配置文件中配置要批改的目标行和列的信息，对该坐标位置处的单元格进行信息的获取，实现起来比较简单。

1. **本章小结**

本章主要介绍了自动评阅系统中的一些策略研究，主要包括考点类型划分策略和考点位置定位策略。在考点类型划分策略中，分析划分粒度过大和过小带来的问题，并提出了一种适中的粒度划分策略。在考点位置定位策略中讨论了根据字符串进行定位，根据文档标签进行定位，根据幻灯片的ID进行定位以及根据Excel文档中的行和列进行定位的几种策略。

# 自动评阅系统的实现

1. **整体设计思路**
2. 读取配置文件，获取各个考点的内容，分值以及考点定位信息。
3. 读取标准答案文件，根据配置文件中的定位信息找到对应的考点，并通过Open Xml SDK获取该考点对应的属性值。
4. 与第二步相似，读取学生文件，根据配置文件中的定位信息找到对应的考点，并通过Open Xml SDK获取该考点对应的属性值。
5. 根据配置文件中配置的考点信息，将第二步和第三步中获取到的标准答案信息与学生文件信息进行匹配，并根据配置文件中相应考点配的分值进行分数的统计。
6. **自动评阅系统模型设计**

Office主观题自动评阅系统主要由三部分构成——考点配置文件、标准答案文件、学生文件。

考点配置文件包含考点类型、考点分值、定位信息。即可以描述为如下所示的三元组：配置文件[考点类型,考点分值,定位信息]。

标准答案文件可将其转化为一个二元组，即标准答案[考点类型，考点属性值]。

学生文件与标准答案一样，学生文件[考点类型，考点属性值]。

图5- 1 自动评阅系统设计思路示意图

如上图5-1所示：首先读取配置文件，获取配置文件[考点类型,考点分值,定位信息]三元组，再读入标准答案，根据考点类型和定位信息，找到考查点，并通过Open XML SDK获取文档信息，生成标准答案[考点类型，考点属性值]模型。采用类似的方式读取学生文件，并生成学生文件[考点类型，考点属性值]模型。最后一步，遍历配置文件[考点类型,考点分值,定位信息]模型中的考点类型，分别在标准答案[考点类型，考点属性值]模型和学生文件[考点类型，考点属性值]模型中对考点属性值这一维度进行匹配，若能匹配则根据配置文件模型中配置的分值给该考点生成对应的[考点类型，得分]，若不匹配则该对应考点分值为0，最终将生成一个[考点类型，得分]的结果。

1. **Office自动评阅系统实现效果**

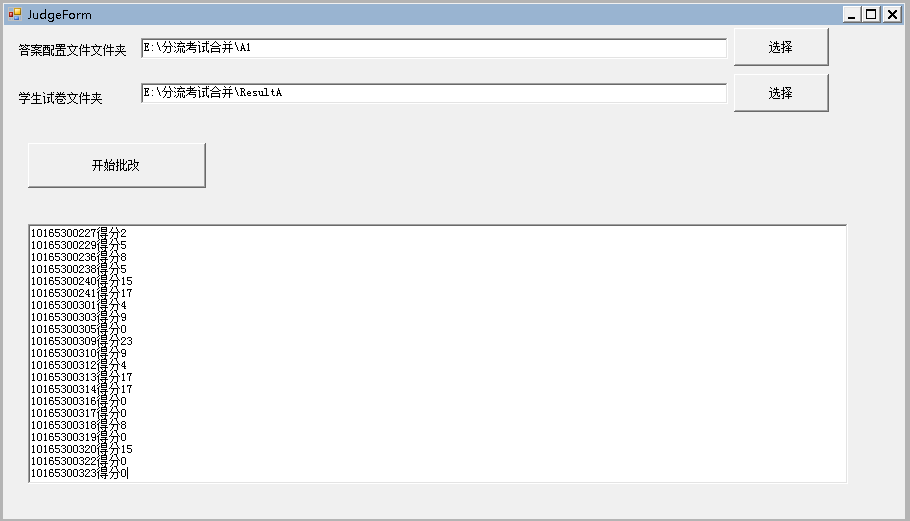
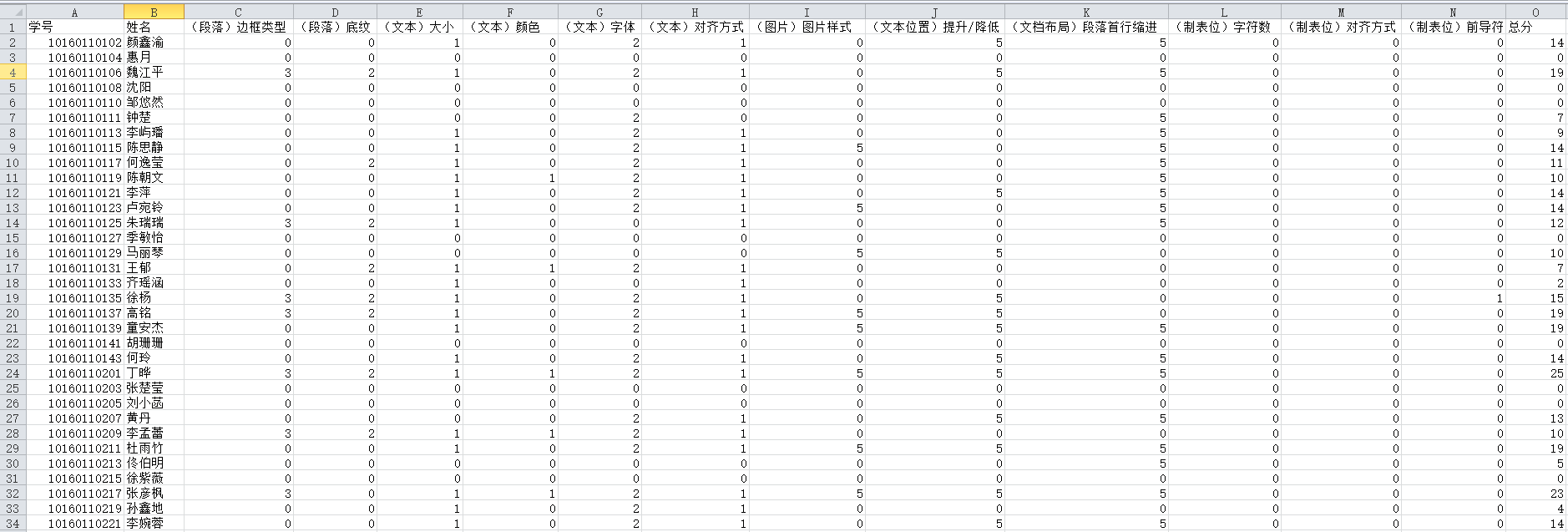
该系统经过测试，批改1500份学生Office主观题试卷大概需要15秒钟左右的时间，最终的界面如下图5-2所示：

图5-2 自动评阅系统界面图

界面上的“答案配置文件文件夹”选择的是标准答案文件以及配置文件所在的目录，“学生试卷文件夹”选择的是所有学生试卷的所在目录。该目录下包含所有学生以学号进行命名的文件夹，在对应文件夹中存放着该学生的试卷。点击“开始批改”将会对学生的试卷进行批量批改，并将最终的批改结果存入Excel文件中，最终的结果不仅包含总分，还包含每一个考点的得分，效果如下图5-3所示：

图5-3 自动评阅系统学生最终得分效果图

1. **本章小节**

本章主要介绍了Office主观题自动评阅系统的实现，包括系统整体的设计思路以及系统中涉及到的三个模型。经过实践证明，该设计思路是可行的，并且取得了良好的批改效果。

# 总结与展望

**6.1 工作总结**

本文首先介绍了Office主观题自动评阅系统的研究背景和意义以及当前的国内外研究现状，之后从Open XML的基本概念以及Office文档的结构特点出发，分别介绍了Word文档结构特点，Excel文档结构特点以及Power Point文档结构特点。在这个基础上，研究了利用Open XML SDK工具对Office文档信息获取的办法，并分三部分介绍了如何通过Open XML SDK获取Word文档信息，Excel文档信息以及Power Point文档信息。之后介绍了对自动评阅系统中的一些策略研究，主要包括考点类型划分策略和考点位置定位策略。又介绍了Office主观题自动评阅系统的实现，包括系统整体的设计思路以及系统中最核心的三个模型，最后，给出了系统实现的界面以及最终的批改结果示例图。根据实验结果表明，该系统达到了预期的目标。

**6.2 研究展望**

Office主观题自动评阅系统虽然已经得到了广泛的关注与研究，但在技术上仍然需要不断地完善。无论是考点类型划分还是考点定位策略等都值得深入研究。本文采用Open XMLSDK对Office文档信息进行提取，并通过选取适中粒度的考点类型配置方式，在一定程度上改善了Office主观题自动评阅系统的适用范围和系统复杂度，同时也取得了良好的性能和效果，但该系统也存在一些不足之处，未来可以考虑从如下几个方面进行进一步的研究：

(1) 在考点类型划分上面需要进一步调研，既要控制好考点类型划分的粒度，同时也要尽量避免各个考点之间产生重叠，使得考点之间会相互影响，从而使最终的评阅结果产生较大的误差。

(2) 考点定位策略的选取是该系统实现过程当中非常关键的一步，如何保证该定位策略能够定位到要批改的考点，以及对于同一个考点考查多处的情况，我们该如何进行区分还有很多值得我们探索的地方。

(3) 人工批改与程序自动批改的一个区别在于：若整个文档产生了较小的变动，在程序的视角来看可能变化是巨大的，而人工看起来则可能意识不到这种变化，从而使得两者最终批改结果有较大误差。这就需要进一步完善系统，在一些容易由于学生误操作的地方采取一些措施，将这种误差降到最低。

# 参考文献

唐学军.VBA的OFFICE操作题自动批阅技术的实现[J].软件,2014,36(2):43-45.

周建良.Word操作题自动阅卷功能的实现[J].宁波职业技术学院学报,2007,11(2):83-85.

邵晓兵,廖慧芬.Office实践操作自动评分的实现[J].计算机与现代,2009,8(12):135-136.

李菲.考试系统中Excel操作题自动阅卷的实现[J].辽宁师专学报(自然科学版).2009(01).

李剑波,李小华.考试系统中操作题自动阅卷技术研究与实现[J].信息技术.2005(10).

王亚利,李井竹.基于VBA的Office操作题自动阅卷技术的研究与实现[J].廊坊师范学院学报(自然科学版).2008(04).

李贵洋,王世伦,俞晓,左友东.Word文档自动判卷的实现[J].中国测试技术.2004(04).

李桂成,张永奎,闫锐.Word操作题自动阅卷的设计与实现[J].计算机工程与设计.2002(04).

李文江.利用VBA实现对Word文档的自动评分[J].重庆文理学院学报(自然版).2007(02) .

骆红波,李舟军,金炳尧,马永进.PowerPoint文档对象分析与自动阅卷的实现[J].计算机与现代化.2006(02) .

殷如文.信息技术操作题自动阅卷系统的研究与实现[J].电脑编程技巧与维护.2011(16) .

李旌燕,王国军.EXCEL操作题自动出题系统设计[J].惠州学院学报(自然科学版).2008(03) .

周波.基于C#的Word操作题自动批改系统的分析与设计[J].昆明冶金高等专科学校学报.2013(03) .

杨秋叶.Excel操作题考试系统设计[J].陕西广播电视大学学报.2013(03) .

王读祥.Word操作题自动判卷关键性技术实现[J].计算机与现代化.2011(10) .

于志奇.Office操作题自动阅卷的探讨与实现[J].电脑知识与技术.2009(09).

张鉴新,周晓晴.Office操作题自动评阅的研究与实现[J].电脑知识与技术.2009(23).

杜定强.用VBA实现Office操作题的自动阅卷[J].电脑知识与技术.2009(36).

# 致 谢

短暂而难忘的研究生生活即将结束，毕业之际，我心中充满了对各位老师、家人和同学的感激之情，正是你们的帮助和默默支持使我能够顺利完成硕士论文。

感谢导师朱敏老师，您严谨的治学态度、一丝不苟的工作精神深深影响了我，不仅在学习上，而且在研究上提出各种忠告和建议，给了我很大帮助，让我顺利完成了硕士研究生阶段的学习。感谢陈优广老师，在我读研期间，对我学习研究上给予了很多帮助和指引，开阔了我的理论视野。在此对所有关心指导教育我的老师表示深深的谢意！

感谢实验室其它同学，感谢你们在平日的生活中给我的热情帮助和支持。

感谢我的家人，感谢你们用辛勤劳动支持我完成了近二十年的求学生涯，感谢你们在我多年成长道路上付出的艰辛和汗水，感谢你们一直以来对我的鼓励、信任和支持。你们善良、勤劳、朴实、正直的品格深深烙印在我心里，让我受用一生。

再次感谢所有教育、关心和帮助过我的人，由衷感谢为评审本文而辛勤工作的老师们！向你们致以最诚挚的谢意！