评分程序

# 1 技术要求

## 1.1 技术平台

基于.net framework 4.5 实现

使用C#语言和Visual Studio 2017进行开发

## 1.2 客户端要求

安装Windows 7及以上操作系统

安装Windows Office 2003 及以上版本

安装Adobe Photoshop CS3 及以上版本

安装.net framework 4.5

## 1.3 代码要求

命名规范，使用面向对象编程，面向接口编程。

建立单元测试项目，保证关键点被单元测试覆盖。

# 2 程序流程

## 2.1 文档解析

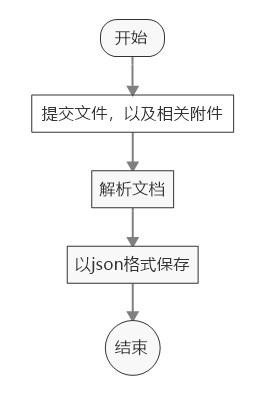
文档解析：解析指定文档并生成Json格式的结构文件。

程序输入：待解析文件（Word, Excel, PowerPoint, Photoshop）（是指原始试题）以及附件（不限格式）



程序输出：Json格式的文档结构文件（以下称 结构文件或）

**注意**： 支持的格式有.doc, .docx. .xls, .xlsx. .ppt, .pptx, .psd



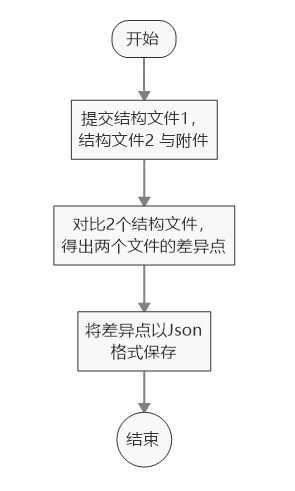
## 2.2 文档检测

文档检测：根据2.1中生成的结构文件对2个结构文件进行对比

程序输入：两个结构文件与相关附件（不限格式）

结构1是原始试题,由2.1生成  
结构2是老师做的标准答案，由老师生成

程序输出：Json格式的差异文件（以下称 差异文件）

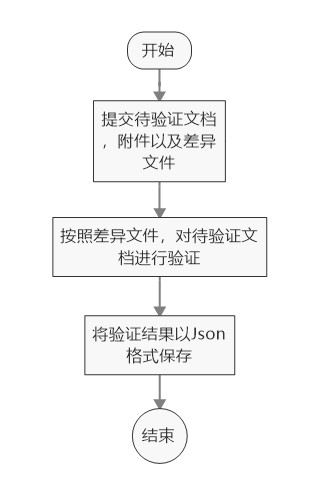


## 2.3 文档验证

文档验证：根据2.2中生成的差异文件对待验证文档与附件进行验证

程序输入：待验证文件（Word, Excel, PowerPoint, Photoshop）（即考生的答案），附件（不限格式）和差异文件

程序输出：Json格式的验证结果文件（以下称 结果文件）



# 3 检测点

上文2.2 文档检测应按照如下的检测点进行检测， 并生成差异点。

**注意：**

1. 各个检测点生成的差异点应尽可能细化，避免在单个差异点中保存大量差异

如： 对一段文字的格式（字体，字号，颜色）进行检测。

推荐的检测结果：

差异点1 字体： 宋体

差异点2 字号： 20

差异点3 颜色： 蓝色

不推荐的检测结果

差异点 1字体： 宋体，字号： 20，颜色： 蓝色

1. 新增的内容，如设置与默认值一致，则不应检测为差异点

如新增一段文字“你好，世界！”字体为宋体，五号黑色

正确的检测结果：

差异点1 新增内容 “你好，世界！”

错误的检测结果：

差异点1 新增内容 “你好，世界！”

差异点2字体： 宋体

差异点3字号： 五号”

差异点4颜色： 黑色

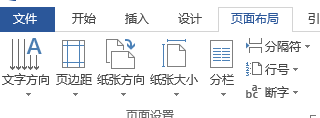
## 3.1 Word检测点

本文以Word 2013为例， 其他版本操作可能略有不同。

### 3.1.1 页面布局

设置方式：页面布局选项卡

检测内容：页边距， 纸张方向， 纸张大小， 分栏设置



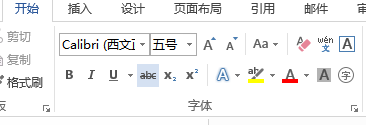
页面布局选项卡

位置是指什么？缩放比例是指什么？

### 3.1.2 字体

设置方式：字体工具栏，字体菜单项

检测内容：字体， 字号， 字样（加粗， 倾斜， 下划线， 删除线， 上标， 下标）， 颜色， 底色，对齐方式（左对齐， 右对齐， 居中对齐， 两端对齐， 分散对齐），缩放， 间距， 位置，字符间距（缩放比例， 位置， 间距）



字体工具栏

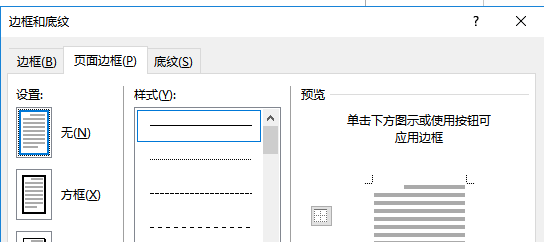


字体菜单项

### 3.1.3 设计

设置方式：边框和底纹菜单项

检测内容：边框设置， 边框样式， 边框颜色， 边框宽度，边框设置， 边框样式， 边框颜色， 边框宽度，底纹颜色， 底纹样式， 底纹样式颜色

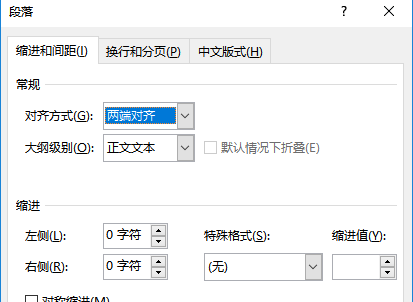


边框和底纹菜单项

### 3.1.4 段落

设置方式：段落菜单项

检测内容：对齐方式，缩进值（左侧，右侧）， 特殊格式（首行缩进值，悬挂缩进值），间距值（端前， 端后），行间距值

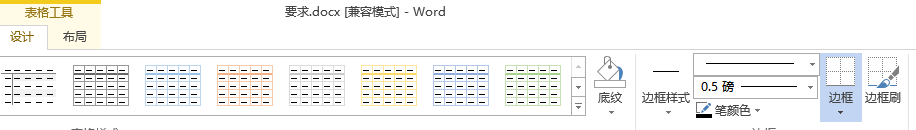


段落菜单项

### 3.1.5 表格

设置方式：表格工具栏

检测内容：表格大小（行数量， 列数量）， 表格属性（底纹，边框样式，边框粗细，边框颜色）



表格工具栏

### 3.1.6 页眉与页脚

设置方式：页眉和页脚工具栏

检测内容：内容（文字，图片）， 格式（字体， 字号， 字样， 颜色， 底色，对齐方式）



插入选项卡

### 3.1.7 文本框

设置方式：设置形状格式菜单

检测内容：填充方式，填充颜色， 填充透明度， 线条样式， 线条颜色， 线条透明度， 线条宽度，文本填充方式， 文本填充颜色，文本填充透明度



未找到该属性

设置形状格式菜单

### 3.18 图片

设置方式：图片工具栏

检测内容：图片边框，图片效果，图片板式



图片工具栏

### 3.1.9 艺术字

设置方式：设置形状格式菜单

检测内艺术字阴影， 映像，发光， 柔化边缘， 三维格式， 三维旋转容：



设置形状格式菜单

### 3.1.10 特殊操作

检测内容1：文字移动或复制

检测内容2：替换操作

检测内容3：插入的特殊符号

检测内容4：是否添加标题，标题的内容和格式（字体， 字号， 字样， 颜色， 底色，对齐方式）

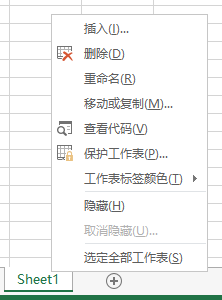
## 3.2 Excel 检测点

本文以Excel 2013为例， 其他版本操作可能略有不同。

### 3.2.1 工作表

设置方式：工作表菜单

检测内容：工作表新增，改名，删除

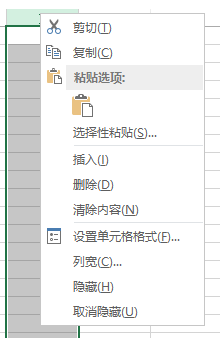


工作表菜单

### 3.2.2 列

设置方式：列操作菜单

检测内容：列宽， 插入列，删除列，隐藏列

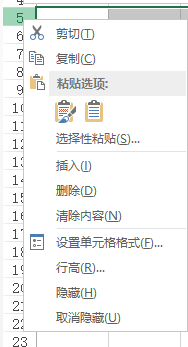


列操作菜单

### 3.2.3 行

设置方式：行操作菜单

检测内容：行高， 插入行，删除行，隐藏行

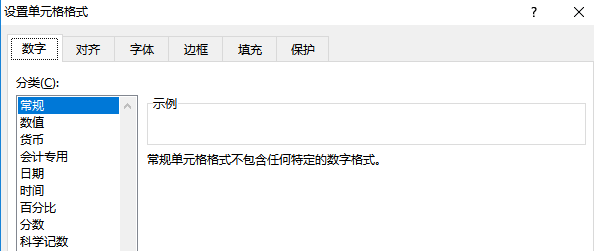


行操作菜单

### 3.2.4 单元格

设置方式：设置单元格格式菜单

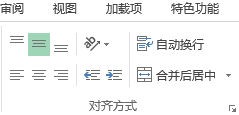
检测内容：单元格分类（常规，数值，货币，日期，时间，百分比，分数，科学计数，文本，特殊，自定义），文本水平对齐，垂直对齐，缩进，字体，字形，字号，颜色，下划线，边框线条，颜色，边框设置（上，下， 左，右），填充颜色



设置单元格格式菜单

设置方式：对齐方式工具栏

检测内容：合并单元格，取消合并单元格



对齐方式工具栏

设置方式：窗口工具栏

检测内容：冻结一个或连续的单元格



窗口工具栏

### 3.2.5 图表

设置方式：图表工具栏

检测内容：柱形图，折线图，饼图，条形图，面积图，散点图，股价图， 曲面图， 雷达图，组合图



图表工具栏

### 3.2.6 公式

设置方式：公式选项卡

检测内容：单元格或连续的单元格是否被设置公式



公式选项卡

### 3.2.7 排序和筛选

设置方式：排序和筛序工具栏

检测内容：是否对列进行筛选，是否对列进行排序



排序和筛选工具栏

### 3.2.8 页面设置

设置方式：页面设置工具栏

检测内容：页边距， 纸张方向， 纸张大小



页面设置工具栏

### 3.2.9 特殊操作

检测内容1： 连续数据填充，行或者列

检测内容2： 相同数据填充，行或者列

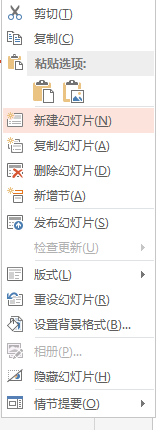
## 3.3 PowerPoint检测点

本文以PowerPoint 2013为例， 其他版本操作可能略有不同。

### 3.3.1 幻灯片

设置方式：幻灯片菜单

检测内容：新建幻灯片， 复制幻灯片，移动幻灯片， 隐藏幻灯片，删除幻灯片， 幻灯片板式



幻灯片菜单

设置方式：设置背景格式菜单

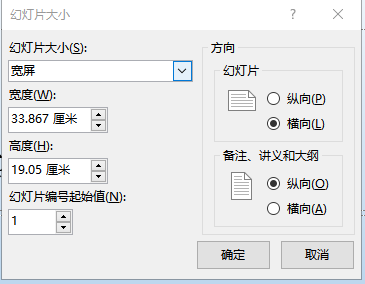
检测内容：背景填充方式，纯色填充设定，渐变填充设定，图片或纹理填充设定， 图案填充设定， 隐藏背景填充



设置背景格式菜单

设置方式：幻灯片大小菜单

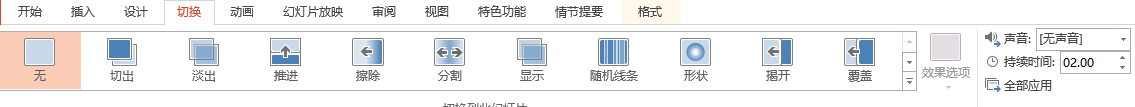
检测内容：幻灯片大小，宽度， 高度， 方向



幻灯片大小菜单栏

设置方式：切换工具栏

检测内容：切换效果， 声音，持续时间， 换片方式



切换工具栏

设置方式：动画工具栏

检测内容：动画效果， 触发，持续时间， 开始，延迟



动画工具栏

### 3.3.2 文本框

设置方式：设置形状格式菜单

检测内容：文本框样式（横排文本框，垂直文本框）， 填充设置， 线条设置， 文本填充设置， 文本边框设置

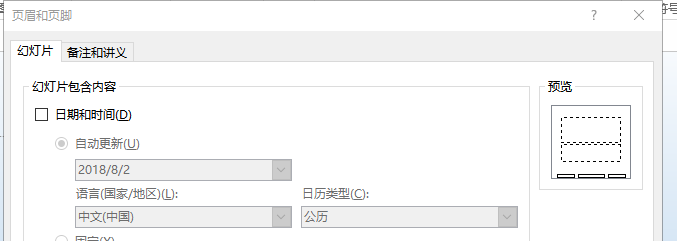


设置形状格式菜单

### 3.3.3 页眉与页脚和幻灯片编号

设置方式：页眉和页脚菜单

检测内容：日期和时间， 幻灯片编号， 页眉，页脚，标题幻灯片中不显示



页眉和页脚菜单

### 3.3.4 艺术字

设置方式：设置形状格式菜单

检测内容：艺术字样式（横排文本框，垂直文本框）， 填充设置， 线条设置， 文本填充设置， 文本边框设置

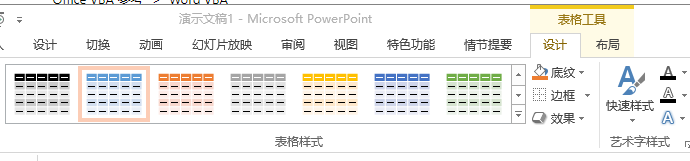


设置形状格式菜单

### 3.3.5 表格

设置方式：表格工具栏

检测内容：表格大小（行数量， 列数量）， 表格属性（底纹，边框样式，边框粗细，边框颜色）



表格工具栏

### 3.3.6 媒体

设置方式：媒体工具栏

检测内容：插入的视频，插入的音频



媒体工具栏

### 3.3.7 链接

设置方式：链接工具栏

检测内容：链接地址，鼠标动作（链接到幻灯片， 运行程序， 播放声音）



链接工具栏

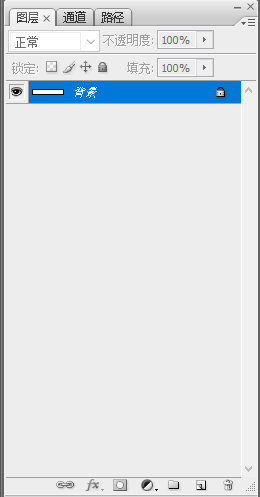
## 3.4 Photoshop检测点

本文以Photoshop CS3为例， 其他版本操作可能略有不同。

### 3.4.1 图层

设置方式：图层工具栏

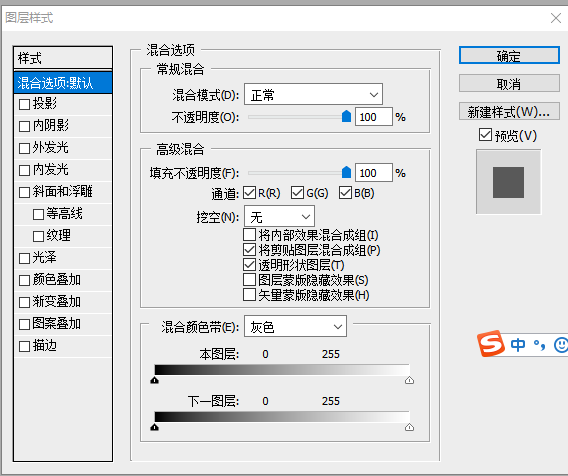
检测内容：新建图层，复制图层，移动图层， 图层改名，删除图层



图层工具栏

设置方式：图层样式菜单

检测内容：混合选项设置，投影设置，内投影设置，外发光设置，内发光设置，斜面和浮雕设置，光泽设置，原色叠加设置，渐变叠加设置， 图案叠加设置，描边设置



图层样式菜单

### 3.4.2 文本

设置方式：文本工具栏

检测内容：方向，字体，字形， 字号， 风格（无， 锐利， 平滑， 浑厚， 犀利， 颜色，变形

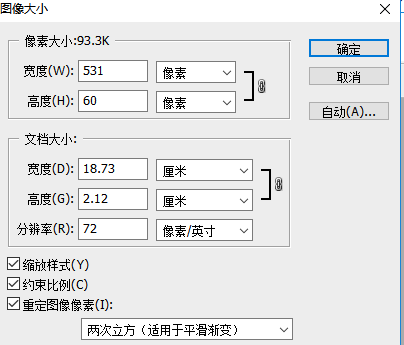


文本工具栏

### 3.4.3 图像大小

设置方式：图像大小菜单

检测内容：像素大小， 文档大小， 缩放样式，约束比例， 重定图像像素

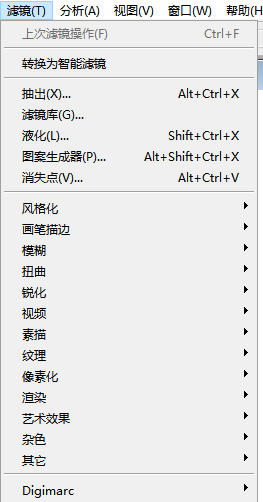


图像大小菜单

### 3.4.4 滤镜

设置方式：滤镜菜单

检测内容：像素大小， 文档大小， 缩放样式，约束比例， 重定图像像素



滤镜菜单

### 3.4.5 工具

设置方式：工具栏

检测内容：污点修复，仿制，印章，橡皮擦，切片，套索，港币



工具栏

# 4 文档格式与设计

## 4.1 差异点格式

差异点使用json描述在文档特定位置的设置。

文档示例：

{

"Parameters": {

"TargetProperty": "frameRate",

"TargetValue": "10",

"TargetIndex": 0

},

"ScoreName": "CSPM.OnlineTest.Business.Flash.Scores.FlashPropertyEqualScore",

"Description": "将文档帧频修改为 10（FPS）。",

"Path": "/FlashDocument[0]/",

"Target": "FlashDocument[0]",

"Priority": {

"Code": 5,

"Name": "Medium"

}

}

格式解析：

对应示例方案中CheckResult类型

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| Parameters | dynamic | 参数，用于Scorer（验证点）判断是否相同 |
| ScorerName | string | 对应的Scorer（验证点） |
| Description | String | 检查结果描述 |
| Path | String | 检查路径，当文档遍历到这个节点时调用 |
| Target | string | 小题主干对象，用于生成试题。执行验证时不会传递这个参数 |
| Priority | QuizPriority | 在小题中的排序，用于生成试题，优先级较低会被排在小题的后面。 验证时不会传递这个参数。 |

## 4.2 结构文档格式建议

建议按照从大到小的顺序生成树形文档，避免循环嵌套。

### 4.2.1 Word建议格式

建议生成的结构文档如下：

文档

文档信息

段落

语句

段落

语句1

语句2

元素

### 4.2.2 Excel 建议格式

建议生成的结构文档如下：

文档

文档信息

列信息

列1

列2

行信息

行1

单元格

单元格[0,0]

单元格[0,1]

行2

单元格

单元格[1,0]

单元格[1,1]

### 4.2.3 PowerPoint建议格式

建议生成的结构文档如下：

文档

文档信息

幻灯片

幻灯片1

元素

艺术字

文本

幻灯片2

元素

艺术字

文本

### 4.2.4 Photoshop 建议格式

建议生成的结构文档如下：

文档

文档信息

图层

元素

元素信息

元素

元素信息

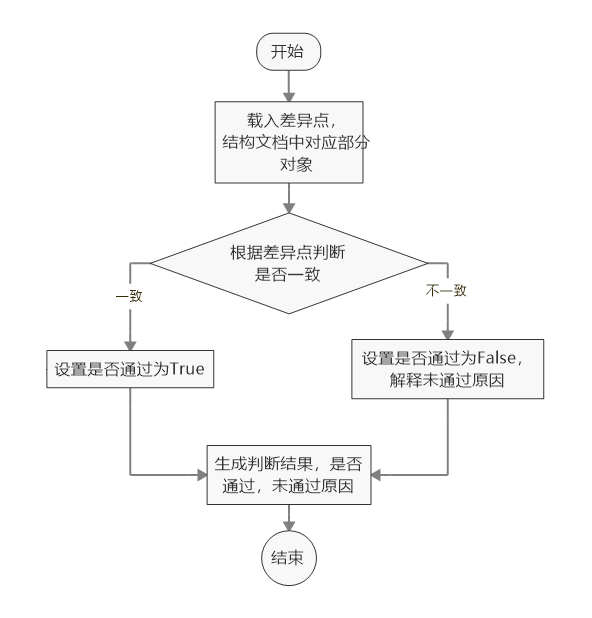
图层

元素

元素信息

## 4.3 验证点设计

### 4.3.1 验证流程



### 4.3.2 验证结果格式

文档示例：

{

"Passed" : "False",

"Description" : "帧数设置错误，正确答案为10，你的答案是15."

}

格式解析：

对应示例方案中ScoreResult类型

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| Passed | bool | 是否通过验证 |
| Description | string | 未通过原因，如通过则留空 |

## 4.4 接口设计

### 4.4.1 文档解析接口

接口定义：

**文档解析接口对应CSPM.OnlineTest.Business.Core中的Itransformer接口**

/// <summary>

/// 转换器接口

/// </summary>

public interface ITransformer

{

/// <summary>

/// 获得中间文件

/// </summary>

/// <param name="originalFile">源文件</param>

/// <param name="exchangeFile">中间文件</param>

void GetExchangeFile(string originalFile, string exchangeFile);

/// <summary>

/// 通过文件获得类型（反序列化）

/// </summary>

/// <typeparam name="T">目标类型</typeparam>

/// <param name="file">文件</param>

/// <returns>反序列化的对象</returns>

T GetTypeFromFile<T>(string file);

}

接口解释：

GetExchangeFile即获得结构文档

GetTypeFromFile 将结构文档反序列化

此处T为结构文档类型

### 4.4.2 文档检测接口

接口定义：

**文档检测接口对应CSPM.OnlineTest.Business.Core中的ICompareHandler接口**

/// <summary>

/// 文档检测接口

/// </summary>

public interface IComparaHandler

{

/// <summary>

/// 试题文件

/// </summary>

string TestFile { set; }

/// <summary>

/// 答案文件

/// </summary>

string AnswerFile { set; }

/// <summary>

/// 附件

/// </summary>

string[] Attachment { set; }

/// <summary>

/// 开始对比

/// </summary>

/// <returns>对比结果</returns>

IList<CheckResult> Compare();

}

接口解释：

Compare() 即生成差异点方法

CheckResult即4.1中定义的差异点

### 4.4.3 文档验证接口

接口定义：

**文档验证接口对应CSPM.OnlineTest.Business.Core中的IValidateHandler接口**

/// <summary>

/// 文档验证接口

/// </summary>

public interface IValidateHandler

{

/// <summary>

/// 答题文件

/// </summary>

string TestFile { set; }

/// <summary>

/// 附件

/// </summary>

string[] Attachments { set; }

/// <summary>

/// 差异点

/// </summary>

IList<CheckResult> Checks { set; }

/// <summary>

/// 开始验证

/// </summary>

/// <returns>验证结构</returns>

IList<ScoreResult> Validate();

}

接口解释：

Validate()方法即生成验证结果

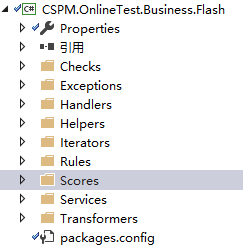
此处Checks即4.4.2 生成的差异点，但其中的Target和Priority属性将为空

**注意**: Validate()生成的验证结果顺序需要与差异点顺序一一对应

# 5 示例方案与参考

## 5.1 Flash文档解析， 检测， 验证

项目结构：



项目结构解释：

|  |  |
| --- | --- |
| 项目 | 解释 |
| Transformers | 文档转换器，用于将flash解释为json结构文档 |
| Iterators | 文档迭代器，用于遍历结构文档 |
| Checks | 检测点 |
| Scores | 验证点 |

## 5.2 可供参考资料

* PSD文件解析方案   
  <https://github.com/NtreevSoft/psd-parser>
* Adobe PSD格式文档  
  <https://www.adobe.com/devnet-apps/photoshop/fileformatashtml/>

<http://www.adobe.com/devnet-apps/photoshop/fileformatashtml/>

* Excel对象模型 <https://docs.microsoft.com/zh-cn/visualstudio/vsto/excel-object-model-overview>
* Word对象模型 <https://docs.microsoft.com/zh-cn/visualstudio/vsto/word-object-model-overview>
* PowerPoint对象模型<https://docs.microsoft.com/en-us/previous-versions/office/developer/office-2010/ff764034(v=office.14)>
* 开发Office解决方案<https://docs.microsoft.com/zh-cn/visualstudio/vsto/developing-office-solutions>

# 演练：Office 编程（C# 和 Visual Basic）

<https://docs.microsoft.com/zh-cn/dotnet/csharp/programming-guide/interop/walkthrough-office-programming>

# 通过使用 Visual C# 功能访问 Office 互操作对:

<https://docs.microsoft.com/zh-cn/dotnet/csharp/programming-guide/interop/how-to-access-office-onterop-objects>

## C#操作Office实例:

<https://www.cnblogs.com/dream844/archive/2012/07/08/2580859.html>

# 比较全的一个C#操作word文档示例:

<https://www.jb51.net/article/67314.htm>

E-iceblue公司的免费[word](http://www.e-iceblue.com/Introduce/free-doc-component.html)组件

<https://www.cnblogs.com/Yesi/p/5190735.html>

<https://www.cnblogs.com/Yesi/p/4203127.html>

# NetOffice组件，通过nuget安装

<https://github.com/netoffice/NetOffice>

NPOI

读取word,首先得添加引用，不同的word版本对应着不同的引用

部分版本对应引用如下：

Microsoft Word 11.0 object library对应Office2003

Microsoft Word 12.0 object library对应Office2007

Microsoft Word 14.0 object library对应Office2010

Microsoft Word 15.0 object library对应Office2013

# DirectoryInfo(String) Constructor

<https://docs.microsoft.com/zh-cn/dotnet/api/system.io.directoryinfo.-ctor?f1url=https%3A%2F%2Fmsdn.microsoft.com%2Fquery%2Fdev15.query%3FappId%3DDev15IDEF1%26l%3DZH-CN%26k%3Dk(System.IO.DirectoryInfo.%2523ctor);k(SolutionItemsProject);k(SolutionItemsProject);k(TargetFrameworkMoniker-.NETFramework,Version%3Dv4.5);k(DevLang-csharp)%26rd%3Dtrue&view=netframework-4.7.2>

string ScoreName = typeof(FlashTests\_zwd).FullName;

结果：

"CSPM.OnlineTest.Business.Core.Tests.FlashTest.FlashTests\_zwd"

var str = "abc";

var i = 10 + 1;

var s = $"跑步我都没吃{str}={i + 0.5};";

结果：

跑步我都没吃abc=11.5;

注释：

/// <summary>

/// 获得节点友好名称

/// </summary>

/// <param name="index">索引</param>

/// <returns>友好名称</returns>

/// <summary>

/// 通过文件获得类型（反序列化）

/// </summary>

**/// <typeparam name="T">目标类型</typeparam>**

/// <param name="file">文件</param>

/// <returns>反序列化的对象</returns>

T GetTypeFromFile<T>(string file);

# 委托之Action<T>和Func<T>

Action委托不返回参数的函数

Func委托带返回参数的函数

void F(int i)

{

Console.WriteLine($"i={i}");

}

public Action<int> B;

[TestMethod]

public void a()

{

B = (i) =>

{

Console.WriteLine("I'm B " + i);

};

B(11);

B = (i) =>

{

Console.WriteLine("2 I'm B " + i);

};

B(11);

B = F;

B(11);

}

**Regex.Match方法：**在输入字符串中搜索匹配正则表达式模式的子字符串，并将第一个匹配项作为单个Match对象返回

正则表达式语言 - 快速参考：

<https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/regular-expression-language-quick-reference?view=netframework-4.7.2>

差异文件中Target相同表示同一个小题，一个小题包括多个考点

"Target": "FlashDocument[0]", FlashDocument表示文档，[0]表示第一个文档，永远只有[0]

"Path": "/FlashDocument[0]/FlashTimeLine[场景1]/",

"Target": "FlashLayer[文本]", // 图层指向名称为“文本”的图层

Newtonsoft.Json:

<https://www.newtonsoft.com/json>

<https://blog.csdn.net/qq_31776303/article/details/78217054?locationNum=7&fps=1>

[Newtonsoft.Json 通过 JObject 读取 json对像 超简单](https://www.cnblogs.com/webenh/p/5745355.html):

<https://www.cnblogs.com/webenh/p/5745355.html>

<https://www.xuebuyuan.com/zh-tw/1833601.html>

**https://www.codeproject.com/Articles/15905/Yet-Another-PSD-Parser**