# 软件介绍

该软件使用C#面向对象程序语言进行开发，主要实现了“剪力滞折减计算”和“受压板件局部稳定折减计算”功能。开发过程中，选择了.Net 6开发平台，采用WPF用户界面框架，构建直观友好的用户界面，遵循MVVM软件架构模式，以提高代码的可维护性和可测试性。

# 技术路线

软件基于.NET 6开发平台。.Net 6是Microsoft于2021年发布的软件开发框架，支持多种应用程序类型，支持开发Windows、Linux、macOS等多个操作系统上的应用，提供了一系列丰富的库和工具，简化了开发过程，提高了开发效率。

在用户界面上，软件采用了WPF（Windows Presentation Foundation）框架。WPF充分利用现代图形硬件，提供了丰富的界面设计能力和非凡的视觉效果，包括布局、样式、模板、动画等。WPF使用可扩展应用程序标记语言 (XAML)，方便开发人员完善应用程序，对核心进行功能进行扩展，更好地组织和管理代码。

在数据存储上，软件使用了 SQLite 数据库。SQLite 是一种轻量级的关系型数据库管理系统，具有小巧、高效、可嵌入等特点。它适用于各种应用场景，尤其适用于嵌入式设备和桌面应用程序。

在数据与界面关系上，软件采用了MVVM（Model-View-ViewModel）架构模式。它将应用程序分为三个层次：Model（模型），View（视图）和ViewModel（视图模型）。Model表示应用程序的数据和业务逻辑，View表示应用程序的用户界面，ViewModel表示View和Model之间的桥梁，它负责处理View的数据绑定和用户交互。这种分离旨在提高代码的可维护性和可测试性，使团队能够更容易协同工作。

总的来说，软件应用了全套的现代化开发技术，提供了可扩展、易维护的软件架构，同时，也确保了软件的跨平台性和高性能。

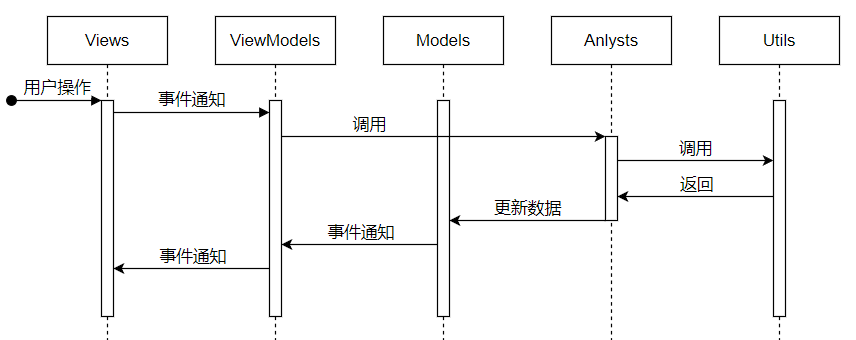
# 模块设计

软件按照功能和职责，划分为多个模块

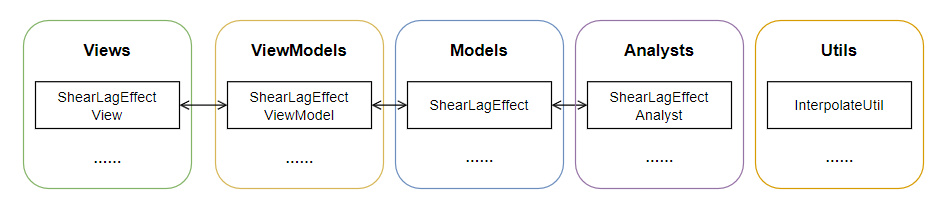
* Views：视图模块，包含每个视图有的代码文件和XAML界面文件。视图通过数据绑定和命令绑定与视图模型连接，实现视图和逻辑的解耦。
* Models：模型模块，包含表达业务逻辑的各种的数据结构。视图模型通过与模型的交互实现对数据的处理和操作，同时，模型依赖属性监听机制实现读写数据库。
* ViewModels：视图模型模块，包含视图和模型的连接关系，用户界面的处理逻辑。视图模型负责处理用户操作的逻辑，为视图和模型建立连接。
* Analysts：分析模块，包含执行特定的计算任务的分析器。视图模型会调用适当的分析器来执行相关的计算，并将结果传递给视图。
* Utils：辅助工具模块，包含一些工具类。它们用于辅助处理数据转换、插值计算、树形结构处理等任务，以提高代码的可复用性和可维护性。

软件的工作流程

* 首先，用户与视图进行交互，触发用户输入事件。
* 视图模型捕获用户输入，并根据业务逻辑，获取或更新模型，或者调用分析器模块执行特定的计算任务。
* 分析器使用工具类提供的方法来辅助计算，并将结果数据传给模型。
* 模型将数据写入数据库文件，并通知视图模型。
* 最后，视图模型获取模型的数据，刷新视图。



以剪力滞计算功能为例，计算过程依赖各个模块的对象协同工作。



另外，软件引入了一些第三方库，以提高代码编写效率

* CommunityToolkit.Mvvm：由Microsoft开发，用于简化MVVM模式的架构和代码。
* Microsoft.EntityFrameworkCore：由Microsoft开发，用于便捷管理和操作Sqlite数据库。
* Fluent.Ribbon：开源库，提供现代化风格的控件和样式，美化用户界面。

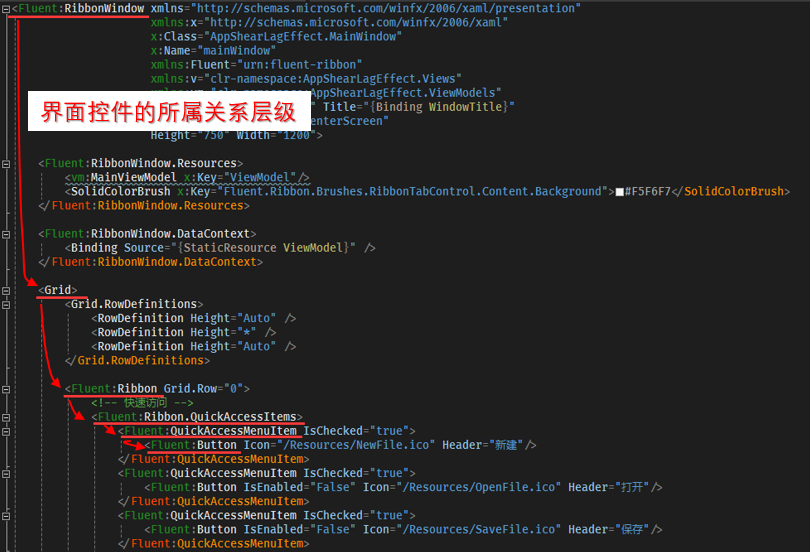
# 主要接口和类设计

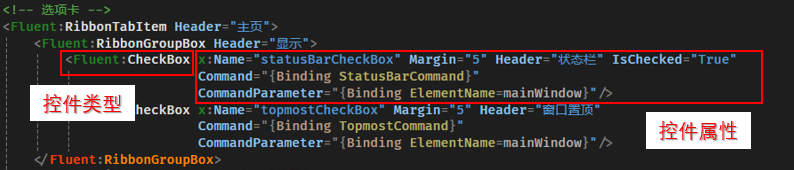
## 视图类

每个视图类都由两个文件\*.xaml和\*.xaml.cs组成，以主窗体为例

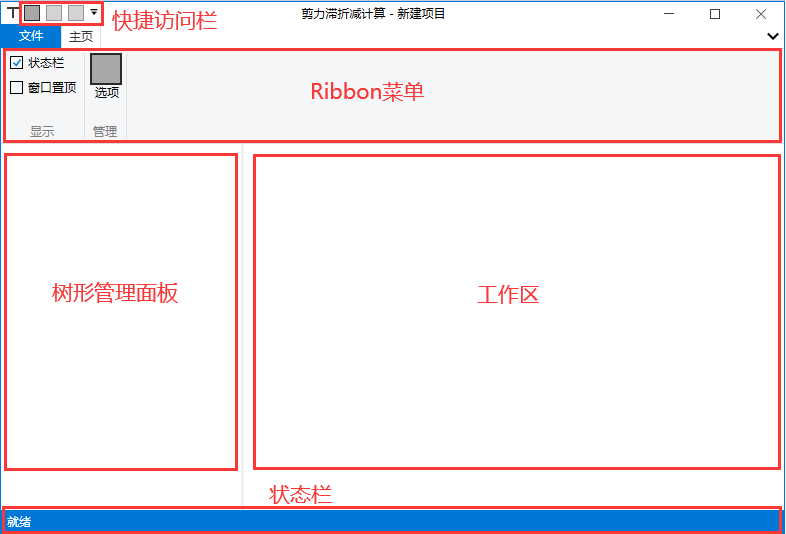


xaml文件使用的是应用程序标记语言，类似于xml，用于定义视图的布局、控件、样式、属性。xaml.cs文件使用的是C#语言，用于定义视图的交互逻辑，例如按钮点击、列表切换等。





MainWindow：主窗体。主窗体派生自Fluent.RibbonWindow（第三方库）基类，实现现代化风格的窗体设计。窗体布局包括上方的快捷访问栏，Ribbon按钮菜单，左侧的树形管理面板，右侧的工作区，下方的状态栏。点击左上角“文件”按钮后，会进入文件选项卡（Backstage）视图，该视图包含了一系列文件级别的功能，包括新建、打开、保存、程序选项、关于和退出程序。由于时间原因，按钮图标和部分功能暂未完成。





AboutView：关于页面。它包含软件的产品介绍和许可信息，属于文件选项卡视图的一部分。

ExplorerView：树形管理面板。

ShearLagEffectView：剪力滞计算的视图

LocalStabilityView：局部稳定折减的视图

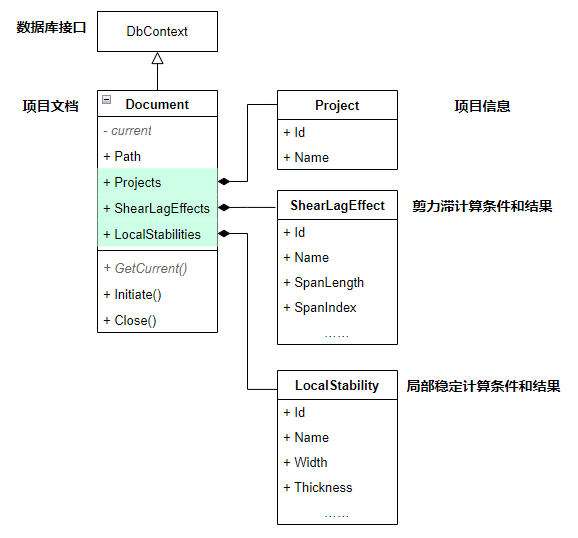
## 模型类

Document：文档，表示整个计算项目的数据对象。项目信息（项目名称）和各类计算的输入条件和计算结果，都以数据表（DbSet）的形式储存在文档中。如果后续新增数据，也必须储存在文档中。文档派生自Microsoft.EntityFrameworkCore.DbContext（第三方库）基类，实现操作数据库，完成数据的序列化和反序列化。数据库的默认路径为%LocalAppData%AppShearLagEffect\，文件后缀名为.db，可以直接用SQLite工具打开。

Project：项目信息，包括项目名称

ShearLagEffect：剪力滞计算的输入条件和计算结果

LocalStability：局部稳定折减的输入条件和计算结果



## 视图模型类

视图模型类作为连接桥梁，往往会将一个视图类和一个模型类关联起来。

MainViewModel：主窗体的视图模型，关联MainWindow视图。

ViewModelBase：工作区的视图模型的基类。

ExplorerViewModel：树形管理面板的视图模型，关联ExplorerView视图。

ShearLagEffectViewModel：剪力滞计算的视图模型，关联ShearLagEffectView视图和ShearLagEffect模型。

LocalStabilityViewModel：局部稳定折减的视图模型，关联LocalStabilityView视图和LocalStability模型。

## 分析器类

IAnalyst：分析器的基类，定义了计算接口。

ShearLagEffectAnalyst：剪力滞计算的分析器，包含剪力滞计算的相关业务规则，根据输入条件，计算剪力滞结果并生成计算报告。

LocalStabilityAnalyst：局部稳定折减的分析器。

## 工具类

ConvertUtil：数据转换类，包含字符串和数组之间的相关转换功能。

InterpolateUtil：插值计算类，包含线性插值计算。

TreeViewHelper：树形工具类，包含树形管理面板的样式和交互行为。

WorkViewFactory：工作视图工厂类，包含创建各类工作视图，并完成模型数据绑定的功能。

# 后续研发方向

由于时间原因，该软件的部分功能还不完善。建议的研发方向有

* 拓展计算能力：参考现有功能，增加更多的桥梁结构相关的计算，如截面特性计算、车道横向分布系数计算等。
* 优化交互流程：调研并梳理用户的主要交互流程，明确先输入什么、后输入什么；哪些是必须输入、哪些是可选输入、哪些是重复输入。组织每个页面的呈现内容，让用户操作更方便。
* 增强数据库能力：实现打开文件、保存文件、撤销、重做和自动备份等功能，打通完整的用户操作流程，保护用户数据。
* 增加图形化能力：根据计算参数绘制示意图、计算结果图，实现在已有的表格交互能力的基础上，增加可视化图形交互能力。
* 增加日志能力：将关键的错误信息打印到日志文件，方便定位问题原因；将用户操作流程打印到日志文件，方便开发人员复现问题。
* 增加自动化测试能力：编写单元测试、自动化脚本等覆盖软件的各个功能。在每次功能修改或者版本发布前运行自动化测试，及时发现问题，以保证软件在每个版本都能正常工作。
* 增加用户管理系统：搭建网络服务，实现注册登录和拉取第三方验证登录服务，支持用户在线登录并领取使用许可证，提前为后续的版权验证、用户数据收集和营销策略做准备。