MVVM:

提供了一种**将数据、业务逻辑和用户界面分离**的解决方案

通过**双向数据绑定**简化了界面与数据的交互，使得开发者能够更专注于业务逻辑的实现。

一个ViewModel可以绑定到**不同**的"View"上

在MVVM模式中，应用程序的**UI**以及**基础表示**和[**业务逻辑**](https://so.csdn.net/so/search?q=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91&spm=1001.2101.3001.7020)被分成三个独立的类：

**视图（View）**：用于封装UI和UI逻辑。

**视图模型（ViewModel）**：用于封装表示逻辑和状态，并作为连接视图（View）和模型（Model）的桥梁。ViewModel可以取出Model的数据，同时处理View中需要展示内容而涉及的业务逻辑。

**模型（Model）**：用于封装应用的业务逻辑和数据。

将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑

允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动MVC的模型层即可

在MVC中View会直接从Model中读取数据而不是通过 Controller。

在MVC里，View是可以直接访问Model的！

https://blog.csdn.net/u013518478/article/details/133755659?ops\_request\_misc=&request\_id=&biz\_id=102&utm\_term=WPF%E6%9E%B6%E6%9E%84&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduweb~default-0-133755659.142^v100^pc\_search\_result\_base9&spm=1018.2226.3001.4187



