模式被认为是草根阶层所依赖的训练有素的设计师和软件工程师集体经验的结晶。

对于反复出现的设计问题，专家已找到解决方案，而模式以易于理解且规范的格式记录了这些经过实践检验的解决方案。

对于新手，应对中等规模项目；对于专家，设计具有指定特征且复杂度而庞大的系统。

采用规范格式记录设计智慧的理念。

（良好（模式描述））格式

GoF专注于设计模式，而这里介绍的模式设计多个抽象层次：从高层的架构模式到设计模式，再到底层的成例(idiom)。

在软件架构这个更大的背景下利用模式。

讨论模式系统，即不仅将模式收集到一个无所不包的容器中，还根据适合的标准对它们进行分组。

GoF将模式分为创建型、结构型、行为型，

这里分成交互性系统、适应性系统、工作组织、通信和访问控制等。

设计问题词汇表，让模式中更有效的识别、指出和讨论问题和解决方案。

提高设计系统的速度是使用模式的一大重要原因。

刊印前将内容呈现出来，对各方都有益。

模式的概念，描述模式的原则。

模式的编目。

架构模式是最高层的模式，旨在提供系统架构的整体骨架。介绍8个架构模式。

设计模式，确定软件系统的整体结构后，设计模式可用于解决常见的问题，比如组织组件以应对复杂性、将工作负荷分配给多个组件以及组织组件间通信。

成例，语言特定的模式。适用于C++和Smalltalk的成例，出处来源。

将模式组织成模式系统，对于作者和使用者都将受益于模式系统，理解，交流，实际运用。

讨论我们对模式架构及其基本原则的认识，并演示模式如何支持这些原则。

背景，问题，解决方案。

（描述性，解释性，描述性是用描述性语言将问题描述，解释性是用解释性语言将问题解释。描述性语言包括复杂，难度大，简单，步骤少，容易，好做，对的，错的，不好。解释性语言包含一步一步地解释某一层面上执行的步骤，以及这些步骤与其他步骤的多少，繁简，难易的对比。）