



# 设计模式概述

张天

软件工程组

[ztluck@nju.edu.cn](mailto:ztluck@nju.edu.cn)

2025年秋季



# 本章教学内容



- 设计模式的诞生与发展
- 设计模式的定义与分类
- GoF设计模式简介
- 设计模式的优点





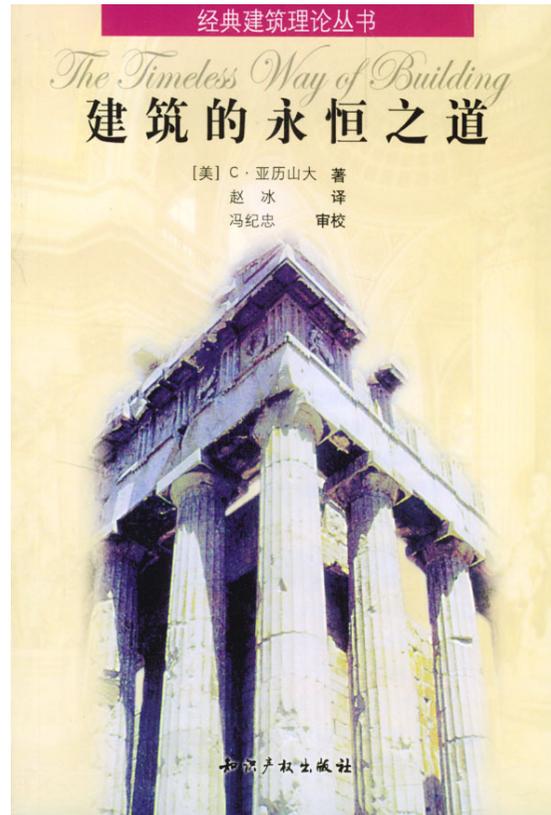
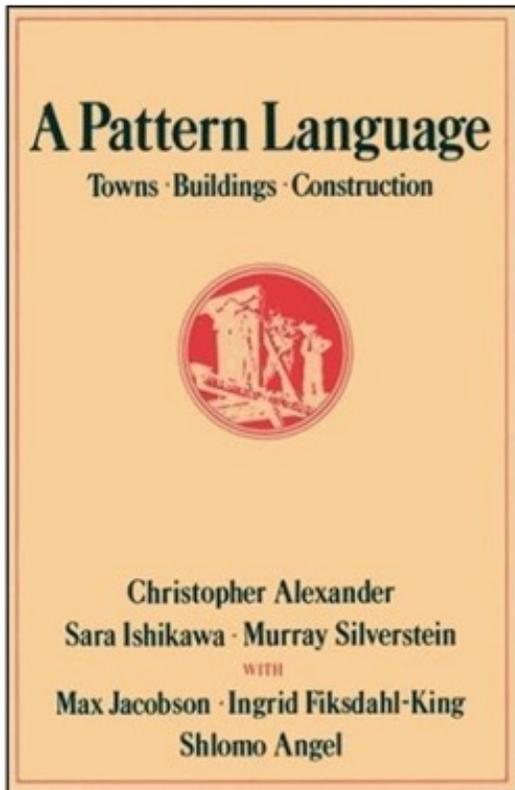
# 设计模式的诞生与发展

## ■ 模式的诞生与定义

- 模式起源于建筑业而非软件业
- 模式(Pattern)之父——美国加利福尼亚大学环境结构中心研究所所长**Christopher Alexander**博士
- 《A Pattern Language: Towns, Buildings, Construction》——  
253个建筑和城市规划模式
- 模式
  - **Context**（模式可适用的前提条件）
  - **Theme或Problem**（在特定条件下要解决的目标问题）
  - **Solution**（对目标问题求解过程中各种物理关系的记述）



# 设计模式的诞生与发展



**Christopher Alexander**



# 设计模式的诞生与发展



## ◆ 模式的诞生与定义

- **Alexander**给出了关于模式的经典定义：每个模式都描述了一个**在我们的环境中不断出现的问题**，然后描述了该问题的**解决方案**的核心，通过这种方式，我们可以无数次地重用那些已有的解决方案，无需再重复相同的工作。
- **A pattern is a solution to a problem in a context**
- 模式是在**特定环境中解决问题的一种方案**



# 设计模式的诞生与发展



## ◆ 软件模式

- **1990**年，软件工程界开始关注**Christopher Alexander**等在这一住宅、公共建筑与城市规划领域的重大突破，最早将该模式的思想引入软件工程方法学的是**1991-1992**年以“四人组(**Gang of Four, GoF, 分别是Erich Gamma, Richard Helm, Ralph Johnson和John Vlissides**)”自称的四位著名软件工程学者，他们在**1994**年归纳发表了**23**种在软件开发中使用频率较高的设计模式，旨在用模式来统一沟通面向对象方法在分析、设计和实现间的鸿沟。



# 设计模式的诞生与发展



**Gang of Four**



# Gang of Four



**Erich Gamma**

苏黎世大学计算机科学博士，是 Eclipse、JUnit 等项目主要技术负责人之一。



**Richard Helm**

墨尔本大学计算机科学博士，原IBM研究员，现在波士顿顾问集团供职。



**Ralph Johnson**

康奈尔大学计算机科学博士，伊利诺伊大学教授。



**John Vlissides**

斯坦福大学计算机科学博士，原IBM研究员，于2005年11月24日因脑瘤去世，享年44岁。



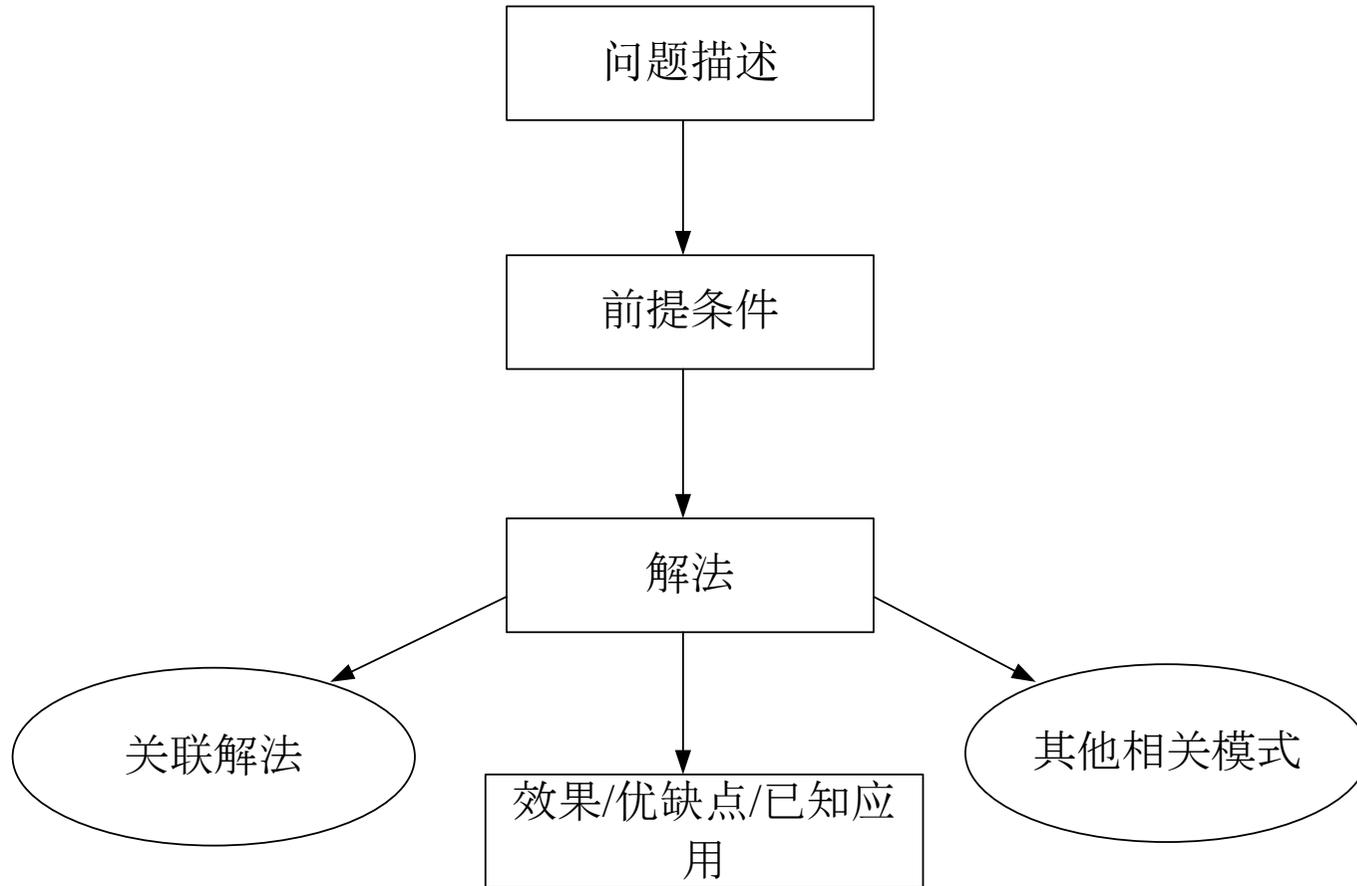
# 软件模式



- 软件模式是将模式的一般概念应用于软件开发领域，即软件开发的总体指导思路或参照样板。软件模式并非仅限于设计模式，还包括架构模式、分析模式和过程模式等，实际上，在软件生存期的每一个阶段都存在着一些被认同的模式。
- 软件模式可以认为是对软件开发这一特定“问题”的“解法”的某种统一表示，它和Alexander所描述的模式定义完全相同，即软件模式等于一定条件下的出现的问题以及解法。软件模式的基础结构由4个部分构成：问题描述、前提条件（环境或约束条件）、解法和效果。



# 软件模式





# 模式的提炼



## ■ 软件模式

- 软件模式与具体的应用领域无关，在模式发现过程中需要遵循大三律(Rule of Three)，即只有经过三个以上不同类型（或不同领域）的系统的校验，一个解决方案才能从候选模式升格为模式。



# 设计模式的发展



- **1987年**，Kent Beck和Ward Cunningham借鉴Alexander的模式思想在程序开发中开始应用一些模式，在OOPSLA会议上发表了他们的成果。
- **1990年**，OOPSLA与ECOOP联合举办，Erich Gamma和Richard Helm等人开始讨论有关模式的话题(Bruce Anderson主持)，“四人组”正式成立，并开始着手进行设计模式的分类整理工作。
- **1991年**，OOPSLA，Bruce Anderson主持了首次针对设计模式的研讨会。
- **1992年**，OOPSLA，Anderson再度主持研讨会，模式已经逐渐成为人们讨论的话题。
- **注: OOPSLA (Object-Oriented Programming, Systems, Languages & Applications, 面向对象编程、系统、语言和应用大会)**，编程语言及软件工程国际顶级会议，2010年改为**SPLASH --- Systems, Programming, Languages and Applications: Software for Humanity**



# 设计模式的发展



- **1993年**，Kent Beck 和 Grady Booch 赞助了第一次关于设计模式的会议，这个设计模式研究组织发展成为著名的Hillside Group研究组。
- **1994年**，由Hillside Group发起，在美国伊利诺伊州(Illinois)的Allerton Park召开了第1届关于面向对象模式的世界性会议，名为**PLoP(Pattern Languages of Programs, 编程语言模式会议)**，简称PLoP'94。
- **1995年**，PLoP'95 仍在伊利诺伊州的Allerton Park举行，“四人组”出版了《设计模式：可复用面向对象软件的基础》(**Design Patterns: Elements of Reusable Object-Oriented Software**)一书，本书成为1995年最抢手的面向对象书籍，也成为设计模式的经典书籍。



# 设计模式的发展



- 从1995年至今，设计模式在软件开发中得以广泛应用，在**Sun的Java SE/Java EE平台**和**Microsoft的.net平台**设计中就应用了大量的设计模式。
- 诞生了越来越多的与设计模式相关的书籍和网站，设计模式也作为一门独立的课程或作为软件体系结构等课程的重要组成部分出现在国内外研究生和大学教育的课堂上。



# 设计模式的定义



- 设计模式(Design Pattern)是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结，使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。



# 设计模式的定义与分类



- 设计模式一般有如下几个基本要素：模式名称、问题、目的、解决方案、效果、实例代码和相关设计模式，其中的关键元素包括以下四个方面：
  - 模式名称 (Pattern name)
  - 问题 (Problem)
  - 解决方案 (Solution)
  - 效果 (Consequences)



# 设计模式的定义与分类



## ■ 设计模式的分类

- 根据其目的（模式是用来做什么的）可分为创建型 (Creational)，结构型 (Structural) 和行为型 (Behavioral) 三种：
  - 创建型模式主要用于创建对象。
  - 结构型模式主要用于处理类或对象的组合。
  - 行为型模式主要用于描述对类或对象怎样交互和怎样分配职责。



# 设计模式的定义与分类



## ■ 设计模式的分类

- 根据**范围**，即模式主要是用于处理类之间关系还是处理对象之间的关系，可分为**类模式**和**对象模式**两种：
  - 类模式**处理类和子类之间的关系**，这些关系通过继承建立，在编译时刻就被确定下来，是属于**静态**的。
  - 对象模式**处理对象间的关系**，这些关系在运行时刻变化，更具**动态**性。



# GoF设计模式简介



范围\目的	创建型模式	结构型模式	行为型模式
类模式	工厂方法模式	(类) 适配器模式	解释器模式 模板方法模式
对象模式	抽象工厂模式 建造原型模式 建造者模式 单例模式	(对象) 适配器模式 桥接模式 组合模式 装饰模式 外观模式 享元模式 代理模式	职责链模式 命令模式 迭代器模式 中介者模式 备忘录模式 观察者模式 状态模式 策略模式 访问者模式



# GoF设计模式简介



## ◆ 创建型模式

- 抽象工厂模式(**Abstract Factory**)
- 建造者模式(**Builder**)
- 工厂方法模式(**Factory Method**)
- 原型模式(**Prototype**)
- 单例模式(**Singleton**)



# GoF设计模式简介



## ◆ 结构型模式

- 适配器模式(**Adapter**)
- 桥接模式(**Bridge**)
- 组合模式(**Composite**)
- 装饰模式(**Decorator**)
- 外观模式(**Facade**)
- 享元模式(**Flyweight**)
- 代理模式(**Proxy**)



# GoF设计模式简介



## ◆ 行为型模式

- 职责链模式(**Chain of Responsibility**)
- 命令模式(**Command**)
- 解释器模式(**Interpreter**)
- 迭代器模式(**Iterator**)
- 中介者模式(**Mediator**)
- 备忘录模式(**Memento**)
- 观察者模式(**Observer**)
- 状态模式(**State**)
- 策略模式(**Strategy**)
- 模板方法模式(**Template Method**)
- 访问者模式(**Visitor**)



# 设计模式的优点



- 设计模式是从许多优秀的软件系统中总结出的**成功的、能够实现可维护性复用的设计方案**，使用这些方案将避免我们做一些重复性的工作，而且可以设计出高质量的软件系统。
- 设计模式的主要优点如下：
  - 设计模式**融合了众多专家的经验**，并以一种**标准的形式**供广大开发人员所用，它提供了一套**通用的设计词汇和一种通用的语言**以方便开发人员之间沟通和交流，使得设计方案更加通俗易懂。对于使用不同编程语言的开发和设计人员可以通过设计模式来交流系统设计方案，每一个模式都对应一个标准的解决方案，设计模式**可以降低开发人员理解系统的复杂度**。



# 设计模式的优点



- 设计模式使人们可以更加简单方便地复用成功的设计和体系结构，将已证实的技术表述成设计模式也会使新系统开发者更加容易理解其设计思路。设计模式使得重用成功的设计更加容易，并避免那些导致不可重用的设计方案。
- 设计模式使得设计方案更加灵活，且易于修改。
- 设计模式的使用将提高软件系统的开发效率和软件质量，且在一定程度上节约设计成本。
- 设计模式有助于初学者更深入地理解面向对象思想，一方面可以帮助初学者更加方便地阅读和学习现有类库与其他系统中的源代码，另一方面还可以提高软件的设计水平和代码质量。



# 本章小结



- 模式是在特定环境中解决问题的一种方案。
- GoF (Erich Gamma, Richard Helm, Ralph Johnson和 John Vlissides)最先将模式的概念引入软件工程领域，他们归纳发表了**23**种在软件开发中使用频率较高的设计模式，旨在用模式来统一沟通面向对象方法在分析、设计和实现间的鸿沟。
- 软件模式是将模式的一般概念应用于软件开发领域，即软件开发的总体指导思路或参照样板。软件模式可以认为是对软件开发这一特定“问题”的“解法”的某种统一表示，即软件模式等于一定条件下的出现的问题以及解法。



# 本章小结



- 设计模式是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结，使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。
- 设计模式一般有如下几个基本要素：模式名称、问题、目的、解决方案、效果、实例代码和相关设计模式，其中的关键元素包括模式名称、问题、解决方案和效果。
- 设计模式根据其目的可分为创建型，结构型和行为型三种；根据范围可分为类模式和对象模式两种。
- 设计模式是从许多优秀的软件系统中总结出的成功的、能够实现可维护性复用的设计方案，使用这些方案将避免我们做一些重复性的工作，而且可以设计出高质量的软件系统。