



Types of Design Patterns

This lesson lists the most used and familiar design patterns

Introduction

Design patterns for object orientated programs are divided into three broad categories listed below. These are the same categories used by GoF in their seminal work on design patterns.

- Creational
- Structural
- Behavioural

Each of these are explained below

Creational


Creational design patterns relate to how objects are constructed from classes. New-ing up objects may sound trivial but unthoughtfully littering code with object instance creations can lead to headaches down the road. The creational design pattern come with powerful suggestions on how best to encapsulate the object creation process in a program.

- Builder Pattern
- Prototype Pattern
- Singleton Pattern
- Abstract Factory Pattern



Structural

Structural patterns are concerned with the composition of classes i.e. how the classes are made up or constructed. These include:

- Adapter Pattern
- Bridge Pattern
- Composite Pattern
-  (/learn) Decorator Pattern

- Facade Pattern
- Flyweight Pattern
- Proxy Pattern

Behavioral

Behavioral design patterns dictate the interaction of classes and objects amongst eachother and the delegation of responsibility. These include:

- Interpreter Pattern
- Template Pattern
- Chain of Responsibility Pattern
- Command Pattern
- Iterator Pattern
- Mediator Pattern
- Memento Pattern
- Observer Pattern

- State Pattern
- Strategy Pattern
- Visitor Pattern



Examples

For most of the patterns, we borrow concepts from the aviation industry to create our examples. You'll find the course regularly talking about F-16s and Boeings to elaborate aspects of the pattern under discussion.

For Interview Prep

For folks, who are rushing through the course for an upcoming interview, I would suggest going through all the **creational design patterns**, **decorator**, **proxy**, **iterator**, **observer** and **visitor** patterns. As you read through them, be sure to look at the Java framework's api examples pointed out in each lesson.

[← Back](#)[Introduction](#)[Next →](#)[Builder Pattern](#)[Mark as Completed](#)[Report an Issue](#)