Use Scenarios of Design Patterns

Design Pattern Name	Topic ID(s)	Summary	Online Materials
Abstract Factory	99	use abstract factory to implement dependency injection	I
Active Record	68	embedded in Ruby on Rails as ActiveRecord	I
	80	PostgreSQL specific usage of Active Record	I
Adapter	91	Android Adapter	I
	52	IBM MobileFirst/Worklight adapters	I, II
Builder	43	Ruby XML Builder	I, II
	7	Java StringBuilder	I
CQRS	115	use CQRS in Domain Driven Design	I
	118	Axon Framework	I
Command	100	command pattern in game programming	I, II
	1, 48, 95	command pattern in GUI design (WPF, GWT, WinForms)	I, II, III
Composite	9	composite in GUI design (SWT, GWT, WPF, JSF)	I, II, III, IV
	93	composite data entity	I
	71	composite in JavaScript frameworks (Marionette.js, ExtJS)	I, II
	43	composite for file systems	I
Content Negotiation	114	content negotiation in ASP.NET	I
	58	content negotiation in Spring MVC	I
CRTP	32	crtp in C++	I
DAO	89	DAO in Microsoft Access	I
	56	combine DAO with service layer	I
	62, 58	DAO in Spring framework	I
Data Mapper	37, 68, 27, 103, 85	DataMapper library	I
	56	data mapper in domain models	I
	93, 118, 115	DataMapper ORM library (involving Entity Framework, Hibernate, Doctrine)	I
	120, 64	PHP DataMapper	I
	118	iBATIS DataMapper framework	I
DTO	93	combine DTO with Entity Framework	I
	115	use DTO in domain riven design	I
Decorator	41	decorator in Python	I
	106	decorator for registering in Django	I
	60	decorator in Zend Framework	I
	73	decorator for login in Django	I
	57	decorator in AngularJS	I
	7	decorator in TypeScript	I
Dependency Injection	62	dependency injection in Spring Bean	I
	14	Google dependency injection framework (Guice, Dagger)	I, II
	57	dependency injection in Angular	I
	12	dependency injection in ASP.NET Core	I
	94	dependency injection in PHP framework (Symfony, Laravel)	I, II
	101	use dependency injection in WPF to decouple views	I, II
	28	dependency injection in AngularJS	I
	56	dependency injection in multi-layer application	I, II
Domain Model	115	Domain Model in Domain Driven Design	I
	93	domain model in ORM (Entity Framework, Hibernate)	I, II
	27	Grails Domain Class	I
Double Checked Locking	16	use double checked locking to implement thread-safe singleton	I
Event Sourcing	76	combine event sourcing with CQRS	I
	8	event sourcing in Event Store database	I
	44	event sourcing in Apache Kafka	I
Facade	94, 64	facade in Laravel framework	I
	56	facade for services	I, II
	62	EJB session bean facade	I
	114	facade for authentication	I
Factory Method	16	Class Factory Methods in Objective-C	I
	62	use factory method to create Spring Bean	I
Factory	57	factory in AngularJS	I
	41	Python Factory	I
	94	factory in Symfony	I
Federated Identity	73	federated identity in Microsoft Azure	I
	52	federated identity in Amazon Cognito	I
File Transfer	44	file transfer between message queues	I
Front Controller	64	front controller in PHP (Zend Framework, Symfony)	I, II
	58	Java implementation of front controller	I
Future	32	future in Scala	I
	78	future in C++	I
HMVC	64	implement HMVC in CodeIgniter	I
	118	Kohana framework	I
Interceptor	58	Spring MVC Interceptor (HandlerInterceptor, WebRequestInterceptor)	I
	106	Castle Windsor Interceptor	I
	93	Hibernate Interceptor	I
	73	login interceptor	I
	57	Angular http interceptor	I
	69	Struts Interceptor	I

	67	Flume Interceptor	I
Iterator	78	C++ iterator, vector/list iterator	I
	53	Java iterator, ListIterator	I
	43	read file by iterator	I
	75	traverse multidimensional data struct	I
	107	traverse tree struct, e.g., QTreeWidgetItemIterator	I
	97	C++ map iterator	I
	36	Java iterator	I
	64	PHP DirectoryIterator	I
Lazy Loading	93	entity lazy loading in ORM (Entity Framework, Nhibernate)	I, II
	46	images lazy loading in jQuery	I, II
	28	feature modules lazy loading in Angular	I
	91	images lazy loading in Android ListView	I
	69	DataTables lazy loading in PrimeFaces	I
Master/Slave	8	MongoDB master-slave replication (database)	I
	52	Jenkins master/slave architecture (project management)	I
	37	MySQL master-slave replication (database)	I
	44	using in ActiveMQ for high availabliity (message queue)	I
	15	Bluetooth master-slave model (communication)	I
	22	master-slave programming paradigm in parallel computing	I
Materialized View	89	materialized view in Oracle	I
	27	materialized view in PostgreSQL	I
	8	materialized view in Apache Cassandra	I
Mediator	114, 43, 52	mediators in WSO ESB	I
	48	event mediator	I
	1	mediator in MVVM	I
Message Broker	44	various message brokers (RabbitMQ, ActiveMQ, Kafka, WSO2 Message Broker)	I, II, III, IV
	89	WebSphere Message Broker	I
Messaging	44	use message queue to implement messaging	I
	76	event-driven messaging	I
	22	messaging in multithreading	I
MVC	120, 64	PHP MVC framework (CakePHP, Zend Framework, CodeIgniter)	I, II, III
	23	ASP.NET MVC (Routing)	I, II
	58	Spring MVC	I
	48	MVC in Java GUI design (JavaFX, Swing)	I, II
	71	JavaScript MVC architecture (Knockout.js, Node.js, AngularJS)	I, II, III
	85	Ruby on Rails MVC framework	I
MVP	39, 91	MVP in Android development	I
	14	combine MVP with Dagger 2 for dependency injection (Android)	I
	1, 92, 102	MVP in WinForms/WPF	I, II
MVVM	102	constructing architectures in WPF	I
	101	MVVM Light Toolkit	I
	71	MVVM in KnockoutJS	I
	39	MVVM in mobile development (Android, iOS)	I, II
	69	applying MVVM to Kendo UI	I
Object ID	8, 27, 71	ObjectId in MongoDB (Mongoose, Meteor)	I
	68	object_id in Ruby	I
	39	objectId in Parse	I
Object Pool	100	avoiding memory fragmentation in game programming	I
	91	reusing ListViews in Android development	I
Observer	39	observer in mobile development (Android, iOS)	I, II
	103	observer in Ruby on Rails	I
	90	observer in Magento	I
Page Objects	110	page object in Selenium	I
	53	page object in Watir	I
	28	page object in Protractor	I
	37	use page object with RubyGems	I
Pipeline	41, 43	shell script pipeline (Unix shell, Powershell, Bash)	I, II, III
	75	pipeline for machine learning (Scikit-Learn)	I
	116	pipeline in MIPS architecture	I
	52	pipeline for projects Continuous Integration/Delivery (Jenkins)	I
	42	graphics pipeline (OpenGL, DirectX)	I, II
	44	pipeline for web service (NServiceBus, Redis, BizTalk)	I, II, III
	8	pipeline for data processing (MongoDB, Hadoop)	I, II
	23	pipeline for JavaScript and CSS assets	I
Pooling	119	http connection pooling	I
	22	thread pooling	I
	100	object pooling in game design	I
	112	database connection pooling	I
	44	object pooling in communication	I
Post/Redirect/Get	58	Post-Redirect-Get in Spring MVC	I
	120	Post-Redirect-Get in ASP.NET MVC	I
Publish/Subscribe	71	use Publish/Subscribe in JavaScript	I
	15	use Publish/Subscribe in Android	I
	8	Redis Publish/Subscribe	I
	80	Meteor Publish/Subscribe	I

Reactor	62	Spring 5 Reactor	I
	48	Python Twisted Reactor	I
Record Set	89	Recordset in MS ADO (Access, Excel)	I, II, III
Reflection	7	use reflection to handle generic types	I
	36	Java Reflection	I
	6	C# Reflection	I
Repository	12	DbContext: combine unit Of work and repository (Entity Framework)	I
	56, 104, 93	use repository in Entity Framework with ASP.NET	I
	115	use repository in Domain-Driven Design (Aggregates)	I
	106	use repository with dependency injection	I
	94	repository in Laravel Doctrine	I
	20	generic repository	I
Service Layers	56, 12	combine service layer with data access layer (repository, entity)	I, II
	62	Spring bean as service layer	I, II
	73	service layer for MVC application	I, II
Service Locator	99	use service locator with dependency injection	I
	106	use service locator in Unity Container	I
	64	service locator in Zend Framework	I
Sharding	8	database sharding (MongoDB)	I
	27, 119	database sharding (MySQL, PostgreSQL)	I, II
	44	message queue sharding (RabbitMQ)	I
	76	Akka cluster sharding	I
STI	103, 86, 85, 27	single table inheritance in Ruby on Rails	I
	93	single table inheritance in ORM (Hibernate ORM, Entity Framework, Doctrine)	I, II, III
Singleton	119	singleton class for database connection (bad design)	I
	62	Spring Bean singleton scope	I
	54	singleton UIViewController in iOS development (bad design)	I, II
	41	singleton-decorator in Python	I
State	71	React component state	I
	100	using state pattern in game programming	I
	23	state pattern for routing in Angular UI.Router	I
Throttling	22	throttling on threads	I, II
	44	throttling on web services	I, II
	119	WCF throttling	I
	46	throttling on resize/scroll events	I, II
Unit of Work	12	unit of work with repository pattern	I
	93, 56, 104	unit of work in ORM (Entity Framework, Nhibernate)	I, II
	106	unit of work with Unity Container	I
	115	unit of work in Domain Driven Design	I
Value Object	115, 56, 99	value object in domain driven design	I, II
	93	value object in ORM (Nhibernate, Entity Framework)	I, II
Viewcontroller	54, 39, 42	UIViewController in iOS	I
Visitor	107	combining with the traversal strategies of tree structure	I, II
	36	parsing Abstract Syntax Trees	I, II
	43, 60	transforming structures into xml files	I

Related Design Pattern Pairs

Design Pattern Pair	Category	Summary	Online Materials
Abstract Factory - Dependency Injection	Co-operation	abstract factory can be used in dependency injection frameworks for creating stateful objects	I
Abstract Factory - Factory Method	Analogy	they are all for creating objects	I
Abstract Factory - Factory	Variation	abstract factory is a factory interface	I
Active Record - DAO	Analogy	both built a layer between the application and persistence layer	I
Active Record - Data Mapper	Analogy	they are for database operation in ORM	I
Active Record - Lazy Loading	Co-operation	lazy loading can be performed on active record instance	I
Active Record - MVC	Dependency	active record can be the Model in MVC	I
Active Record - Repository	Analogy	they are all patterns for data persistence	I
Active Record - Single Table Inheritance	Co-operation	active record usually allows single table inheritance	I
Adapter - Decorator	Analogy	they are similar in functions	I
Adapter - Facade	Analogy	they have similar intents	I
Bridge - Strategy	Analogy	both of them decouple an abstraction from its implementation	I
Builder - Decorator	Analogy	they are all patterns to add extensions	I
Builder - Factory	Analogy	they are different versions of constructors	I
CQRS - Event Sourcing	Co-operation	CQRS is often used along with event sourcing for efficient queries	I
CQRS - Messaging	Co-operation	messaging can be used to send commands to the domain in CQRS	I
Command - MVVM	Dependency	command is often used in MVVM architecture to coordinate the view with the viewmodel	I
Command - Strategy	Analogy	both patterns encapsulate an algorithm and decouple implementation details from their calling classes	I
Composite - Iterator	Co-operation	use iterator to recursive the structure of composite	I
Composite - Visitor	Co-operation	apply visitor to composite	I
DAO - DTO	Analogy	they all operate on data between classes or modules	I
DAO - Factory	Co-operation	factory can be built to produce DAOs	I
DAO - Record Set	Dependency	record set is often used in DAO to manipulate data	I
DAO - Repository	Analogy	they are all responsible for data access of a software system	I
Data Mapper - Domain Model	Co-operation	data mapper can be used for transferring data between the domain logic and the database	I
Data Mapper - Repository	Analogy	they are all for data access	I
DTO - MVC	Dependency	DTO as the Model of MVC	I
DTO - MVVM	Dependency	DTO as the Model of MVVM	I
DTO - Repository	Analogy	both data access relevant patterns	I
DTO - Service Layers	Dependency	DTO can be used in service layer for communication	I
DTO - Value Object	Analogy	they are all used as data containers	I
Decorator - Interceptor	Analogy	they achieve similar functions	I
Decorator - Strategy	Analogy	both patterns can be used to add behaviours to the base components	I
Dependency Injection - Factory	Analogy	they all have the purpose to separate the use of a certain component	I
Dependency Injection - Interceptor	Co-operation	interceptor classes may be targets of dependency injection, e.g., in Context and Dependency Injection (CDI)	I
Dependency Injection - MVVM	Dependency	dependency injection can be used to decouple the ViewModel in MVVM	I
Dependency Injection - Reflection	Dependency	dependency injection can be implemented by using reflection	I
Dependency Injection - Repository	Co-operation	repositories can be injected via dependency injection	I
Dependency Injection - Service Locator	Analogy	the class is still responsible for creating its dependencies in both patterns	I
Dependency Injection - Singleton	Analogy	they can all make dependencies for objects	I
Dependency Injection - Strategy	Analogy	they all allow us to set run-time behaviours of objects	I
Double Checked Locking - Singleton	Dependency	double checked locking can be used to make singleton thread-safe	I
Facade - Mediator	Analogy	they all encapsulate the functionalities of systems	I
Factory Method - Factory	Variation	factory is a simplified version of factory method	I
Factory - Service Locator	Analogy	both encapsulate the creation of the objects	I
Factory - Singleton	Analogy	they are all for creating objects	I
Factory - Strategy	Analogy	they are similar in implementations	I
HMVC - MVC	Analogy	HMVC is a variation of MVC	I
Iterator - Visitor	Analogy	both iterator and visitor can be used to visit structures of elements	I
Lazy Loading - Singleton	Dependency	lazy loading can be used to make singleton thread-safe	I
Master/Slave - Sharding	Analogy	they can be all database partitioning approaches	I
Mediator - MVVM	Dependency	use mediator to implement communication between View-Models in MVVM	I
Mediator - Observer	Analogy	they have similar functions	I
Mediator - Publish/Subscribe	Dependency	mediator can be used to implement the publish/subscribe model	I
Message Broker - Messaging	Variation	message broker is a way for messaging	I
Message Broker - Publish/Subscribe	Dependency	use message broker to implement publish-subscribe messaging	I

Messaging - Publish/Subscribe	Variation	publish/subscribe is a kind of messaging pattern	I
MVC - MVP	Analogy	MVC and MVP are all for building architectures of presentation	I
MVC - MVVM	Analogy	MVC and MVVM are all for building architectures of presentation	I
MVC - Observer	Dependency	observer can be used to synchronize the Model and the View in MVC	I
MVC - Repository	Co-operation	the repositories interact with the Controller in MVC	I
MVC - Service Layers	Co-operation	service layers interact with the controller in MVC	I
MVP - MVVM	Analogy	MVP and MVVM are all for building architectures of presentation	I
MVVM - Repository	Co-operation	repository can be used with the Model in MVVM	I
Observer - Publish/Subscribe	Variation	publish/subscribe pattern is a variation of observer pattern	I
Reflection - Singleton	Co-operation	singletons should be made reflection-proof	I
Repository - Service Layers	Co-operation	repository can be used in the data access layer to accompany with the service layer	I
Repository - Unit of Work	Co-operation	unit of work is often implemented on repositories	I
State - Strategy	Analogy	the two patterns are pretty similar in practice	I
Strategy - Visitor	Analogy	they have similar behaviours	I