

# Entwurfsmuster - Übersicht

	Erzeugungsmuster	Strukturmuster	Verhaltensmuster
klassenbasiert	<p><b>Fabrikmethode</b> Definiert eine Klassenschnittstelle mit Operationen zum Erzeugen eines Objektes aber lässt Unterklassen entscheiden, von welcher Klasse das zu erzeugende Objekt ist.</p>	<p><b>Adapter (klassenbasiert)</b> Passt die Schnittstelle einer Klasse an eine andere von ihren Klienten erwartete Schnittstelle an.</p>	<p><b>Interpreter</b> Definiert für eine gegebene Sprache eine Repräsentation der Grammatik sowie einen Interpreter, der die Repräsentation nutzt, um Sätze in der Sprache zu interpretieren.</p> <p><b>Schablonenmethode</b> Definiert das Skelett eines Algorithmus in einer Operation und delegiert einzelne Schritte an Unterklassen.</p>
objektbasiert	<p><b>Abstrakte Fabrik</b> Bietet eine Schnittstelle zum Erzeugen von Familien verwandter oder voneinander abhängiger Objekte an, ohne ihre konkreten Klassen zu benennen.</p> <p><b>Erbauer</b> Trennt die Konstruktion eines komplexen Objekts von seiner Repräsentation, so dass derselbe Konstruktionsprozess unterschiedliche Repräsentationen erzeugen kann.</p> <p><b>Prototyp</b> Bestimmt die Arten zu erzeugender Objekte durch die Verwendung eines prototypischen Exemplars, und erzeugt neue Objekte durch Kopieren dieses Prototypen.</p> <p><b>Singleton</b> Sichert ab, dass eine Klasse genau ein Exemplar besitzt, und stellt einen globalen Zugriffspunkt darauf bereit.</p>	<p><b>Adapter (objektbasiert)</b> Passt die Schnittstelle einer Klasse an eine andere von ihren Klienten erwartete Schnittstelle an.</p> <p><b>Brücke</b> Entkoppelt eine Abstraktion von ihrer Implementierung, so dass beide unabhängig voneinander variiert werden können.</p> <p><b>Dekorierer</b> Erweitert ein Objekt dynamisch um Zuständigkeiten.</p> <p><b>Fassade</b> Bietet eine einheitliche Schnittstelle zu einer Menge von Schnittstellen eines Subsystems.</p> <p><b>Fliegengewicht</b> Nutzt Objekte kleinster Granularität gemeinsam, um große Mengen von ihnen effizient verwenden zu können.</p> <p><b>Kompositum</b> Fügt Objekte zu Baumstrukturen zusammen, um Teil-Ganzes-Hierarchien zu repräsentieren und Kompositionen von Objekten einheitlich zu behandeln.</p> <p><b>Proxy</b> Kontrolliert den Zugriff auf ein Objekt mit Hilfe eines vorgelagerten Stellvertreterobjekts.</p>	<p><b>Befehl</b> Kapselt einen Befehl als Objekt.</p> <p><b>Beobachter</b> Definiert eine 1-zu-n-Abhängigkeit zwischen Objekten, so dass die Änderung des Zustands eines Objekts dazu führt, dass alle abhängigen Objekte benachrichtigt werden.</p> <p><b>Besucher</b> Kapselt eine auf den Elementen einer Objektstruktur auszuführende Operation als ein Objekt.</p> <p><b>Iterator</b> Ermöglicht den sequentiellen Zugriff auf die Elemente eines zusammengesetzten Objekts, ohne seine zugrundeliegende Repräsentation offenzulegen.</p> <p><b>Memento</b> Erfasst und externalisiert den internen Zustand eines Objektes ohne seine Kapselung zu verletzen, so dass das Objekt später in diesen Zustand zurückversetzt werden kann.</p> <p><b>Strategie</b> Definiert eine Familie von Algorithmen, kapselt jeden einzelnen und macht sie austauschbar.</p> <p><b>Vermittler</b> Definiert ein Objekt, welches das Zusammenspiel einer Menge von Objekten in sich kapselt.</p> <p><b>Zustand</b> Ermöglicht es einem Objekt, sein Verhalten zu ändern, wenn sein interner Zustand sich ändert.</p> <p><b>Zuständigkeitskette</b> Vermeidet die Kopplung eines Auslösers einer Anfrage an seinen Empfänger, indem mehr als ein Objekt die Möglichkeit erhält, die Anfrage zu erledigen. Verkettet die empfangenen Objekte, und leitet die Anfrage an der Kette entlang, bis ein Objekt sie erledigt.</p>