

Design Patterns

OP2 - Komplexere Anwendungen mit UML entwerfen und entwickeln



Inhalt

1	Allgemeine Informationen	2
---	--------------------------------	---



1 Allgemeine Informationen

- Die so genannte „Gang of Four“ (Erich Gamma, John Vlissides, Ralph Johnson und Richard Helm) haben sich in den 1990ern angeschaut und gesammelt auf welche Weise Entwickler wiederkehrende Probleme bei der Softwareentwicklung gelöst haben. Viele Entwickler haben dieselben oder ähnliche Lösungen gefunden, um häufig auftretende Probleme zu beheben. Die GoF (Gang of Four) haben dann diese Lösungen zusammengetragen und systematisiert und in dem 1994 erschienenen Buch "Design Patterns: Elements of Reusable Object-Oriented Software" zusammengefasst. Das Buch wurde ein Bestseller und gilt als absoluter Standard der Softwareentwicklung mit OOP.
- Design patterns (deutsch: Entwurfsmuster) sind typische Lösungen zu wiederkehrenden Problemen in der Softwareentwicklung.
- Sie sind wie vorgefertigte Blaupausen, die wir anpassen können, um auftauchende Design Probleme im Quellcode zu lösen.
- Entwurfsmuster werden eher entdeckt als bewusst erfunden.
- Seitdem wurden viele weitere Muster gefunden. Der Muster Ansatz wurde sehr beliebt und heute finden sich Muster in allen Programmierparadigmen auch außerhalb der OOP.
- Entwurfsmuster sind ein generelles Konzept, um ein spezifisches Problem zu lösen. Wir können zwar dem Entwurfsmuster grob folgen, müssen dieses aber entsprechend für unser konkretes Programm anpassen und implementieren. Wir können nicht einfach ein vorgegebenes Entwurfsmuster nehmen und kopieren.
- Entwurfsmuster dürfen nicht mit Algorithmen verwechselt werden. Beide stellen zwar Lösungen zu bekannten Problemen dar, aber ein Algorithmus gibt genaue Anweisungen und Handlungsschritte vor, während ein Entwurfsmuster eine allgemeinere Beschreibung einer Lösung ist.
- Die meisten Entwurfsmuster sind eine formale, allgemeine Beschreibung, damit Leute diese in vielen Kontexten anwenden können.
- Es gibt außerdem noch Architekturmuster und Idiome.

- Idiome sind grundlegende, low-level Muster, die nur auf eine konkrete Programmiersprache zutreffen.
 - Architekturmuster sind universelle high-level Muster. Diese können praktisch in vielen verschiedenen Sprachen umgesetzt werden. Im Gegensatz zu den Entwurfsmustern werden diese verwendet, um das Design einer ganzen Anwendung umzusetzen. (Beispiele: MVC, MVVM, MVP etc.)
 - Wir können alle Muster auch noch nach ihrer Absicht und Zweck unterscheiden. Die GoF unterscheiden drei Gruppen an Entwurfsmustern:
1. **Erstellungsmuster:** stellen Instanziierungsmechanismen für Klassen bereit, die Flexibilität und Wiederverwendbarkeit umsetzen.
 2. **Strukturmuster:** zeigen wie Objekte und Klassen in größere Strukturen eingebettet werden und dabei flexibel und effizient bleiben.
 3. **Verhaltensmuster:** kümmern sich um eine effektive Kommunikation zwischen Objekten und die Zuweisung von Verantwortlichkeiten von Objekten.