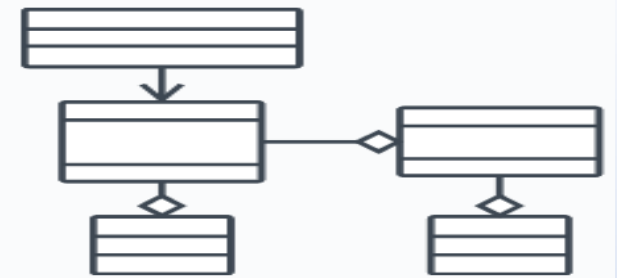
UML Diagramm



Was ist UML?

Die Unified Modeling Language (UML) ist eine visuelle Modellierungssprache für die Architektur, das Design und die Implementierung von komplexen Softwaresystemen.

Die UML besteht aus verschiedenen Diagrammarten. Insgesamt beschreiben UML-Diagramme die Grenzen, die Struktur und das Verhalten von Systemen und den darin enthaltenen Objekten.

Zwar ist UML keine Programmiersprache, jedoch gibt es Tools, die UML-Diagramme nutzen, um Code in verschiedenen Sprachen zu generieren. UML hat einen direkten Bezug zu objektorientierter Analyse und Design.

UML und seine Rolle beim objektorientierten Modellieren und Design

In der Informatik, also der Wissenschaft von Algorithmen und Daten, gibt es viele Paradigmen oder Modelle zur Lösung von Problemen. Dabei stehen vier Modellkategorien für die Problemlösung zur Verfügung: imperative, funktionale, deklarative und objektorientierte Sprachen (OOP).

In objektorientierten Sprachen werden Algorithmen durch die Definition von „Objekten“ und die Interaktion der Objekte untereinander ausgedrückt. Diese Objekte können manipuliert werden und sie existieren in der echten Welt. Dabei kann es sich um Gebäude, Widgets auf einem Desktop oder Menschen handeln.

Objektorientierte Sprachen dominieren die Welt der Programmierung, denn sie können reale Objekte modellieren. UML ist eine Kombination aus mehreren objektorientierten Notationen: objektorientiertes Design, Objektmodellierungstechnologie und objektorientierte Softwareentwicklung.

UML nutzt die Stärken dieser drei Ansätze für ein schlüssigeres Verfahren, das sich einfacher verwenden lässt. UML vereint die Best Practices für die Erstellung und Dokumentation unterschiedlicher Aspekte der Modellierung von Software- und Geschäftssystemen.

Von UML festgelegte Modellierungskonzepte

Die Systementwicklung konzentriert sich im Großen und Ganzen auf drei unterschiedliche Systemmodelle:

- **Funktional: Hierbei handelt es sich um Anwendungsfalldiagramme, welche die Systemfunktionalität aus Sicht des Benutzers beschreiben.**

- **Objekt: Hierbei handelt es sich um Klassendiagramme, welche die Systemstruktur im Hinblick auf Objekte, Attribute, Assoziationen und Vorgänge beschreiben.**

- **Dynamisch: Interaktionsdiagramme, Zustandsdiagramme und Aktivitätsdiagramme werden verwendet, um das interne Verhalten eines Systems zu beschreiben.**

Objektorientierte Konzepte in UML

Die Objekte in der UML sind Entitäten, die auch in der Realität existieren. In der Softwareentwicklung können Objekte dafür verwendet werden, um das zu erstellende System so zu beschreiben oder zu modellieren, dass es für die Domain relevant ist. Objekte ermöglichen außerdem die Zerlegung komplexer Systeme in verständliche Komponenten, sodass immer nur ein Teilbereich dargestellt wird.

Hier sind einige grundlegende Konzepte einer objektorientierten Welt:

- **Objekte Repräsentation** einer Entität und des jeweiligen Grundbausteins.

- **Klasse Plan** eines Objekts.

- **Abstraktion Verhalten** einer Entität aus der echten Welt.

- **Verkapselung Mechanismus**, bei dem Daten zusammengefügt werden, um diese vor der Außenwelt zu verbergen.

- **Vererbung** Der Mechanismus, bei dem neue Klassen aus einer vorhandenen erstellt werden.

- **Polymorphismus** Definiert den Mechanismus, der angewendet wird, um unterschiedliche Formen anzunehmen

Arten von UML-Diagrammen

UML verwendet Elemente und verknüpft diese auf unterschiedliche Art und Weise miteinander, um verschiedene Arten von Diagrammen zu erstellen: zum einen Diagramme, die statische oder strukturelle Aspekte eines Systems darstellen, zum anderen verhaltensbasierte Diagramme, die die dynamischen Aspekte eines Systems erfassen.

- **Strukturelle UML-Diagramme**
  - Klassendiagramm
  - Objektdiagramm
  - Paketdiagramm
  - Kompositionsdiagramm
  - Komponentendiagramm
  - Verteilungsdiagramm

  } Architekturdiagrammen

- **Verhaltensbasierte UML-Diagramme**
  - Use-Case-Diagram
  - Aktivitätsdiagramm
  - Zustandsdiagramm

- **Interaktionsdiagramme**
  - Sequenzdiagramm
  - Interaktionsdiagramm
  - Zeitdiagramm

- Strukturelle UML-Diagramme
- Verhaltensbasierte UML-Diagramme
- Interaktionsdiagramme

_____>

- Strukturelle UML-Diagramme

•Klassendiagramm Das am häufigsten verwendete UML-Diagramm und die wichtigste Grundlage für jede objektorientierte Lösung. Klassen in einem System, Attribute und Vorgänge sowie die Beziehung zwischen den einzelnen Klassen. Klassen werden gruppiert, um Klassendiagramme zu erstellen, wenn große Systeme als Diagramm dargestellt werden sollen.
•Komponentendiagramm Stellt die strukturelle Beziehung von Softwaresystemelementen dar. Wird am häufigsten für komplexe Systeme mit mehreren Komponenten eingesetzt. Komponenten kommunizieren über Schnittstellen.
•Kompositionsstrukturdiagramm Kompositionsstrukturdiagramme werden verwendet, um die interne Struktur einer Klasse darzustellen.
•Implementierungsdiagramm Illustriert die Systemhardware und die zugehörige Software. Nützlich, wenn eine Softwarelösung auf mehreren Maschinen mit individuellen Konfigurationen implementiert wird.
•Objektdiagramm Zeigt die Beziehung zwischen Objekten unter Verwendung von Beispielen aus der Realität. Hierbei wird illustriert, wie ein System zu einem bestimmten Zeitpunkt aussieht. Da Daten in Objekten zur Verfügung stehen, können diese ebenfalls dafür verwendet werden, um Beziehungen zwischen Objekten zu verdeutlichen.
•Paketdiagramm Es gibt zwei spezielle Arten von Abhängigkeiten, die zwischen Paketen definiert werden: Paketimporte und Paketverschmelzungen. Pakete können die unterschiedlichen Ebenen eines Systems darstellen, um die Architektur zu visualisieren. Paketabhängigkeiten können so dargestellt werden, dass die Kommunikationsmechanismen zwischen verschiedenen Schichten erkennbar sind.

- **Strukturelle UML-Diagramme**
- **Verhaltensbasierte UML-Diagramme**
- **Interaktionsdiagramme**

_____>

Verhaltensbasierte UML-Diagramme

- **Anwendungsfalldiagramm** Stellt eine bestimmte Funktionalität eines Systems dar und wurde entwickelt, um zu illustrieren, wie Funktionen zueinander in Beziehung stehen und welche internen/externen Akteure es gibt.

- **Aktivitätsdiagramme** Grafisch dargestellte Geschäfts- oder Betriebsabläufe, um die Aktivität eines Teils oder einer Komponente in einem System zu visualisieren. Aktivitätsdiagramme werden alternativ zu Zustandsdiagrammen verwendet.

Zustandsdiagramm Ähnlich wie Aktivitätsdiagramme beschreibt diese Art von Diagramm das Verhalten von Objekten, die in ihrem aktuellen Zustand unterschiedliche Verhaltensweisen an den Tag legen

- **Strukturelle UML-Diagramme**
- **Verhaltensbasierte UML-Diagramme**
- **Interaktionsdiagramme**

_____>

Interaktionsdiagramme

- **Sequenzdiagramm Zeigt, wie und in welcher Reihenfolge Objekte miteinander interagieren. Solche Diagramme repräsentieren Interaktionen für ein bestimmtes Szenario.**

- **Interaktionsübersichtsdiagramm Es gibt sieben Arten von Interaktionsdiagrammen und dieses Diagramm zeigt die Abfolgesequenz von Aktionen im modellierten System.**

Zeitverlaufsdiagramm Wie bei Sequenzdiagrammen wird hier das Verhalten von Objekten in einem bestimmten Zeitraum dargestellt. Wenn es nur ein einziges Objekt gibt, ist das Diagramm recht simpel. Bei mehr als einem Objekt werden die Interaktionen der Objekte während des festgelegten Zeitraums dargestellt