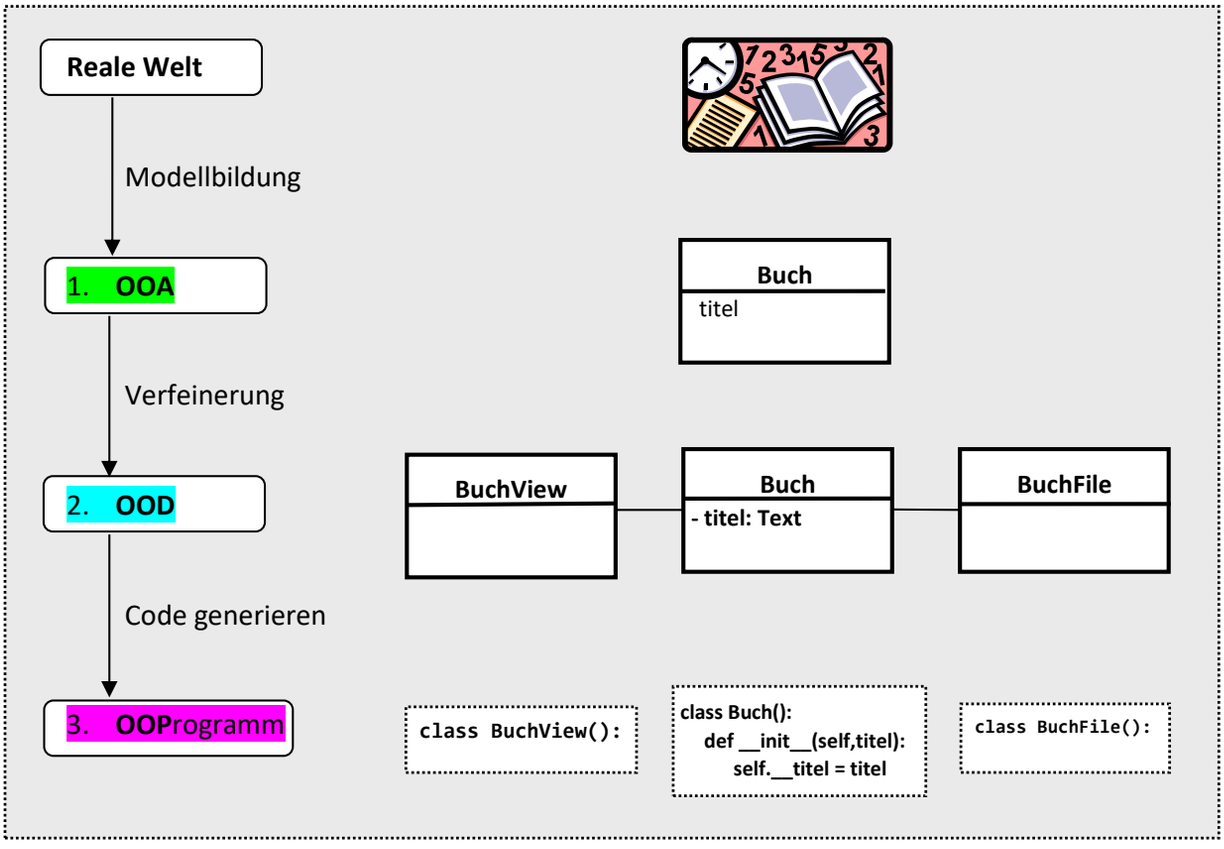
Software-Entwicklung: 3-Schichten-Architektur

Ziel ist die Entwicklung eines Modells zur Lösung einer Problemstellung.
Die Entwicklung des Modells zum fertigen Programm erfolgt in 3 Phasen:



1. OOA – Objektorientierte Analyse (Definitionsphase)

Aus Objekten (Gegenständen, Begriffen) der realen Welt sind Modell-**Objekte** und **Klassen** zu entwickeln.
Jede Klasse ist dabei für einen bestimmten Aspekt des Gesamtsystems verantwortlich. Das Zusammenwirken der Objekte wird durch **Assoziationen** (Beziehungen) beschrieben.
Die Daten werden in **Attributen**, die Funktionalität in **Operationen** der Klassen modelliert.

Wir unterscheiden zwischen:

statischem Modell

- Klassendiagramm / Objektdiagramm

und

dynamischen Modellen

- Sequenzdiagramm
  Zusammenarbeit der Objekte / Reihenfolge der Verarbeitungsschritte
- Zustandsdiagramm
  Objektzustände / Lebenszyklus von Objekten

2. OOD – Objektorientiertes Design (Entwurfsphase)

Die Klassen und Assoziationen des OOA-Modells werden ergänzt und in Details spezifiziert:

- Datentypen / Parameter der Operationen ...
- Bedingungen / Wiederholung bei Operationsaufrufen ...

3. OOP – objektorientierte Programmierung (Implementierungsphase)

Aus dem OOD-Modell kann direkt der Programmcode generiert werden.

3-Schichten-Architektur

Große Softwareprojekte müssen übersichtlich strukturiert sein. Dadurch werden die Wartung und Erweiterung der Software vereinfacht. Eine Möglichkeit bietet die 3-Schichten-Architektur.

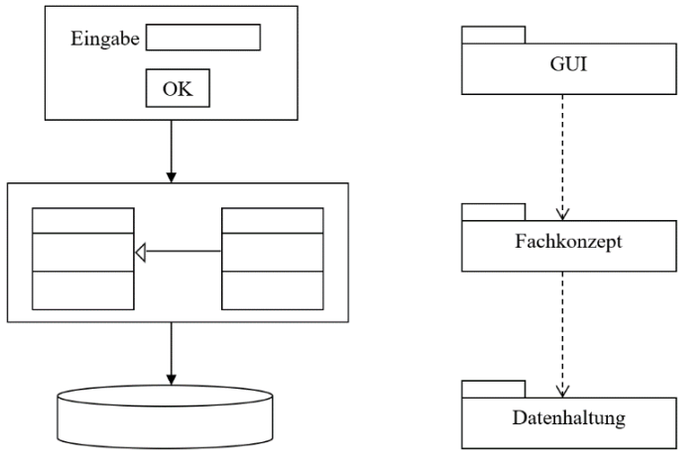
Frontend

Schicht 1: GUI (Graphical User Interface, Benutzerschnittstelle, Oberfläche)

Präsentationsschicht

Funktion:

- Ein- und Ausgabemasken (Formulare) bereitstellen
- Eingaben des Benutzers entgegennehmen und überprüfen (Plausibilitätskontrolle)
- Eingabewerte aufbereiten (z.B. Datentyp) und an Schicht 2 liefern
- Ausgabewerte aus Schicht 2 entgegennehmen, aufbereiten (z.B. formatieren) und ausgeben.



Frontend

Schicht 2: Fachkonzept (Steuerung; Verarbeitungslogik)

Logikschicht

Backend

Funktion:

- Steuerung des Programmablaufs (z.B. Eingabesteuerung)
- Verarbeitung der eingegangenen Daten zu Ausgangsdaten
- Logik (z.B. Berechnungen, Algorithmen)
- Ausgangsdaten ausgeben (➔ Schicht 1) oder speichern (➔ Schicht 3)

E-V-A

Schicht 3 – Datenhaltung

Datenzugriffsschicht

Funktion:

- Daten speichern und aufbereiten (z.B. sortieren)
- Kommunikation mit der Datenbank

In objektorientierten Sprachen wird jede Schicht durch mindestens eine Klasse repräsentiert. Diese Klassen können mit Hilfe öffentlicher Operationen (=Schnittstelle der Klasse) Botschaften (Nachrichten) austauschen.
Meistens ändern sich die Anforderungen an ein Software-System im Laufe der Zeit. Entsprechende Änderungen sollen jedoch mit möglichst wenig Aufwand erfolgen.

Vorteile der 3-Schichten Architektur:

- Bei Änderung der Benutzeroberfläche müssen weder Fachkonzept noch Datenhaltung angepasst werden. Das Portieren auf ein anderes Betriebssystem betrifft evtl. nur die GUI.
- Die GUI-Schicht ist unabhängig von der Datenhaltung. Die GUI weiß nicht, wie der Zugriff auf die Daten erfolgt. Dadurch ist die Datenkapselung einfacher umzusetzen.
- Eine Änderung der verwendeten Datenbank hat nur Auswirkungen auf die unterste Schicht

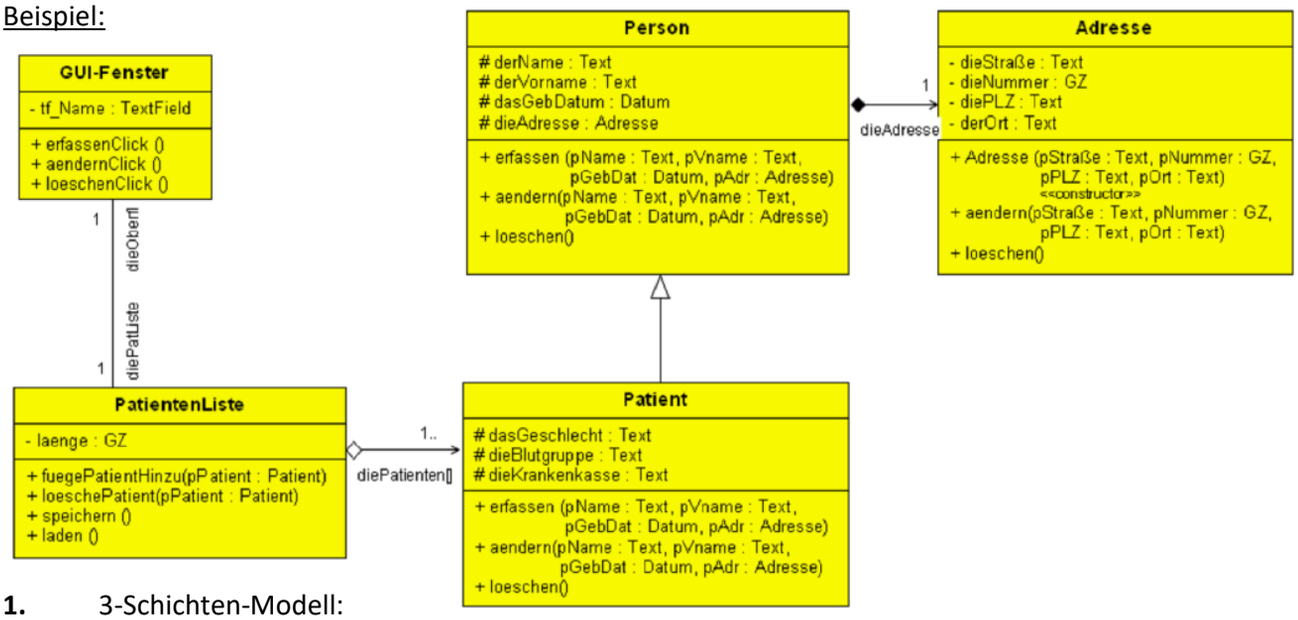
⇒ Jede Schicht kann, unabhängig von den anderen Schichten, ihre Aufgabe umsetzen.

Ein **Nachteil** der 3 Schichten Architektur stellt die Performance dar. Warum?

- Performance-Overhead
- Komplexität der Implementierung
- Erhöhter Wartungsaufwand
- Beanspruchung von (vielen) Ressourcen
- Zusätzliche Tests
- Speichern und Verwalten von Daten ohne Verlust bei der Übertragung

Übungsaufgaben

Beispiel:



1. 3-Schichten-Modell:
Warum ist es sinnvoll die fachliche Funktionalität einer Anwendung, deren Benutzerschnittstelle und die Datenhaltung zu trennen ?

Übersichtlichkeit, Unabhängigkeit, Schichten können leichter ausgetauscht werden.

2. Aus welchen Diagrammen besteht ein statisches UML-Modell ?

Klassendiagramm, Objektdiagramm

3. Aus welchen Diagrammen besteht ein dynamisches UML-Modell ?

Sequenzdiagramm, Zustandsdiagramm

4. Das Softwarehaus „UML-Soft“ führt in einer Bibliothek eine Analyse durch.
Folgende Informationen stehen zur Verfügung:

a) Für jedes Buch sind Titel, Autor und Jahr zu speichern.
b) Die erfassten Bücher sind nach Titeln aufsteigend sortiert in einer Datenbank zu speichern.
c) Jede Ausleihe von Büchern wird im System gespeichert.
d) Defekte Bücher werden aus der Bibliothek entfernt und in der Datenbank-Datei mit einem „D“ gekennzeichnet.
e) Das System soll jederzeit einen Überblick über die Ausleihhäufigkeit der einzelnen Bücher erlauben.
f) Für die Realisierung der Benutzeroberfläche soll die Bibliothek „GUI“ verwendet werden.
g) Da es sich um eine Bibliothek mit mehreren Filialen handelt, ist eine Client-Server-Anwendung notwendig; die zentralen Daten sollen auf dem Server liegen.

Unterscheiden Sie, welche Informationen in der Analyse (OOA) und welche im Entwurf / Design (OOD) erfasst werden müssen.

Analyse: a, (b), c, e

Design: b, d, f, g

"Mini-Projekt" 3-Schichten-Architektur

- 2er Teams
- Situation
- Recherche: GUI's in Python
- Klassendiagramm — diagrams.net, draw.io
  - Alle OOP-Konzepte
  - Leicht: Interfaces
- Modell in Python umsetzen

Stoff Datenbank: Konsolenausgaben ...

- Aaron, Aleksey, Dougie
- Jonathan, Maria
- Bastian, Daniel
- Luis Batt, Joel
- Nick, Jason
- Latizia, Tinja
- 2 mal Luis