

Warum betreiben wir modularen Klassenentwurf?

An dieser Stelle kann man sich mit zwei Fragestellungen befassen:

1. **Warum macht man das überhaupt?** Könnte man nicht einfach alles Funktionalität in einer Klasse unterbringen, anstatt das Programm auf viele einzelne Klassen(dateien) zu verteilen?
2. Wenn die OO-Modellierung eine Problems nicht eindeutig ist - **woran erkennt man dann, ob man es „gut“ gemacht hat?**

Warum verteilt man die Funktionalität und den Code auf mehrere Klassen?

Wenn man ein Problem sinnvoll modularisiert und modelliert, hat das viele Vorteile:

- Lesbarkeit des Quellcodes → Etwas stimmt mit dem Tor nicht? Also muss man in der „tor“-Klasse schauen und nicht 5000Zeilen Code durchscrollen, bis man zu dem Teil kommt, der das Tor erzeugt. Es erleichtert auch Änderungen an der Funktionalität, wenn stets klar ist, wo bestimmte Eigenschaften und Fähigkeiten festgelegt sind.
- Wenn man Klassen geschickt modelliert, kann man Sie in anderen Programmen wiederverwenden - nicht umsonst spricht man von „Klassenbibliotheken“.
- Neue Objekte können durch neue Klassen ein ein Modell eingefügt werden - du willst Hindernisse auf dem Spielfeld? Kein Problem mit der zusätzlichen „hindernis“-Klasse.

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:modellierung:warum:start?rev=1635168812>

Last update: 25.10.2021 15:33

