

Gründe für OO:

- Idee: Bauplan/Schaltplan/Klasse und Haus/Platine/Objekt
- Orientierung an der menschlichen Wahrnehmung, Wunsch, "Objekte" der realen Welt möglichst 1:1 in Programmiersprache zu beschreiben
- Was zusammengehört ist zusammen: Attribute und Methoden (keine Verschmierung verschiedener Ideen übers ganze Programm)
- Test pro Klasse einfach möglich
- „globale“ Variablen pro Klasse (keine anstrengende Übergabe in Methoden der Klasse), aber globale Variablen für alles gibt es zum Glück nicht
- Kein doppelter Code
- Zugriffsrechteverteilung (z.B. private)
- Polymorphie: einfachere Variablenbenennung u.a.
- ich möchte verstanden werden, Beispiel (aus <http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/implementation/articles/289606/>) für nicht-objektorientierte Kommunikation wäre: "Schau mal die schöne Kombination aus Zellulose, Lignin, Pektin, Stärke, Chlorophyll und Wasser.", OO: "Schau mal, der schöne Baum!"
- ich möchte verstehen
- Ich bin faul: Ich möchte schnell verstehen und nicht so viel Code eintippen. Also erst mal in Ruhe nachdenken, ehe die Finger bewegt werden.
- Ich mag Abwechslung: Ziel ist es, Probleme einmal zu lösen und nicht immer wieder.
- Ich habe lieber kleine Schmerzen als richtig große: am Anfang des Projektes ein wenig unangenehme Arbeit zu erledigen (z.B. nachdenken) schützt vor sehr viel unangenehmer Arbeit gegen Ende des Projektes, Beispiel Zahnarzt
- Ich habe gern Erfolgserlebnisse: fertigprogrammierte Klassen sind Teilerfolge, sind gut separat testbar und können damit abgehakt werden.
- Emotionale Gründe gegen OOP:
 - In Software-Entwicklung wird manchmal absichtlich gegen das Verstehen gearbeitet, um sich unabhkömmlich zu machen
 - Der eine ist zu faul zum Tippen und denkt lieber länger, der andere ist zu faul zum Denken und tippt lieber mehr
 - Angst vor Veränderung
- Natürlich: innerhalb von Objekten gibt es immer noch sequentielle Programmierung und auch genug Bit-Pfrieemelei
- Noch mal die OO-Vorteile: leichtere Verständlichkeit, einfachere Testbarkeit, Senkung des Programmieraufwandes und der Codezeilen, Erhöhung der Sicherheit der Software, Verringerung des Wartungsaufwand (bei der Entwicklung über zukünftige Erweiterungen oder Änderungen nachdenken)

Die Arbeit mit OO macht einfach mehr Spaß!