



DON'T MAKE ME THINK

DAS USABILITY-PRINZIP GILT AUCH BEIM CODEN

Best Practice / Konventionen

Eine Funktion sollte...

- **ABSTRAKT** (=Kontext-unabhängig) sein
- möglichst nur **EINEN** Zweck erfüllen
- nach ihrem **Zweck benannt** sein:
«kann ich sie nicht benennen, will ich zu viel auf einmal»
- sich ans **Naming** des Projekts halten
(**camelCase** oder **snake_case**)

A blue ribbon graphic with a folded end on the right, containing the text "DON'T MAKE ME THINK" in white capital letters.

DON'T MAKE ME THINK

Funktions-Pitch

Ablauf

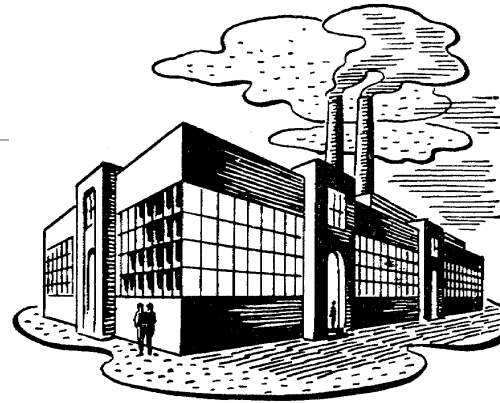
- Jeweils Gruppe A und B
- Gruppen sind gegenseitig Auftraggeber und Auftragnehmer



Auftrag

Zeit: 15min

Input:
Rohmaterial



Output:
Fertiges Produkt

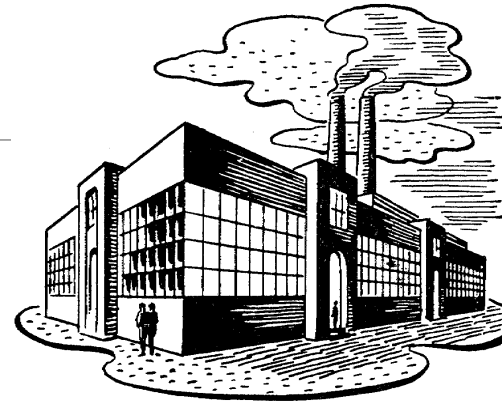
eine Funktion in Auftrag geben:

- **Was** soll die Funktion erledigen, lösen, produzieren? Kurze Beschreibung
- Welche Ausgangslage besteht (gibt es einen **Daten-Input**)?
- Welcher **Output (Daten und Datentyp)** wird erwartet?

Auftrag

Zeit: 30min

Input:
Rohmaterial



Output:
Fertiges Produkt

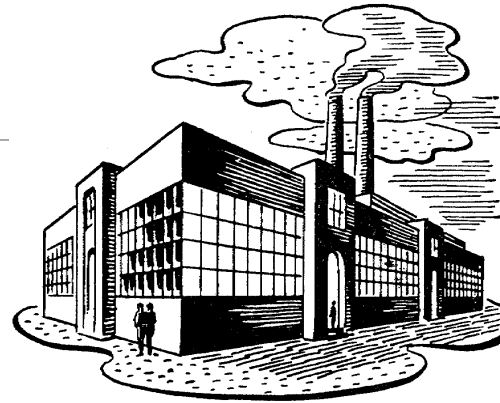
eine Funktion erstellen:

- Entwickelt die Funktion
- Kommentiert sie nach den Doxygen Konventionen zu Funktionskommentaren (<https://en.wikipedia.org/wiki/Doxygen>)

Auftrag

Zeit: 10min

Input:
Rohmaterial



Output:
Fertiges Produkt

Funktion prüfen / Feedback geben

- Mach die Funktion das, was ihr wolltet?
- Ist die Funktion verständlich dokumentiert? / Kann sie angewendet werden?
- Ist die Funktion abstrakt genug? Wenn nicht, warum?