

VI. Funktionen

3. Wert- und Referenzparameter

Felder werden als Referenzparameter übergeben: `fkt_feld_ref_1/2.cc`

Bsp: `void f(int (&x)[10])` Abkürzung: `void f(int x[])`

4. Überladen von Funktionen und Operatoren

`fkt_ueber_1/2/3/4/def.cc`, `op_ueber.cc`

Für **einen Funktionsnamen** kann es **mehrere Funktionsdefinitionen** geben. Anzahl und Typ der Argumente sind unterschiedlich.

Operatoren: Benutze "+", "-", "*", etc. für benutzerdefinierte Datentypen.

Bsp: "+" für Vektoren, "*" für Matrizen, komplexe Zahlen, etc.

5. Kommandozeilenparameter

`komm_zeil_par.cc`

Übergabe von Parametern an Funktion `main()` beim Programmstart

Syntax: `int main(int argc, char * argv[])`

`argc`: Anzahl der Parameter; `argv[i]`: i^{ter} Kommandozeilenparameter als String

VII. Klassen und Objekte

1. Motivation

Ziel: Mehr **Sicherheit** und **Flexibilität** durch Zugriffskontrolle auf Elemente eines Objekts

Idee: Zusammenfassung von Daten/Funktionen/Anweisungen in Klassen;

Zugriff über **Schnittstelle**, **Implementierung** für Benutzer der Klasse nicht "sichtbar"