

0.1 Abstraktion des DMF

Das DMF basiert auf einer Abstraktion der Datenstrukturen aus mehreren Sprachen. Dafür wurden die Sprachen Java, Typescript, Python, Golang, Rust und C analysiert. //TODO Auswahl der Sprachen

0.1.1 Analyse

Analyse der Typen

Es wurde analysiert, welche Typen als Referenz oder als Wert als Variablentyp genutzt werden können.

Typen	Java	Typescript	Python	Golang	Rust	C
Wert	Primitive Typen	Primitive Typen	Primitive Typen	Alle Typen	Alle Typen	Alle Typen
Referenz	Objekte	Objekte, Arrays, Funktionen, Klassen	Alles außer primitive Typen	Explizit	Explizit	Explizit

TODO: Welche Primitive Typen Warum Typen die Keine Primitive Typen sind als Argumente. (Datetime)

0.1.2 Analyse von Nullwerten

Nullwerte sind besonders aus Java bekannt und stellen das Fehlen eines Wertes dar. Es zählt zu der Definition eines Types dazu, zu definieren, ob der Typ Nullwerte erlaubt. Dies muss auch für Werte und Referenzen evaluiert werden.

Nullwerte	Java	Typescript	Python	Golang	Rust	C
Wert	nein	nein	ja	nein	Explizit	nein
Referenz	ja	Explizit	ja	ja	Explizit	ja

TODO: Referenz nullable und Argumente nicht nullable

0.1.3 Collectionstypen

Um 1:n- oder n:m-Beziehungen im Datenmodell modellieren zu können wurden drei Collection-Typen aus Java ausgewählt und passende Äquivalente zu finden.

Collectiontypes	Java	Typescript	Python	Golang	Rust	C
List	ja	ja (Array)	ja	ja (slice)	ja	ja (Array)
Set	ja	ja	ja	nein	ja	nein
Map	ja	ja	ja (dictionary)	ja	ja	nein