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 Pritimive Typen sind als Argumente. (Datetime)

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

Collectiontypen

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

TODO: Erklärung der Typen Besonderheiten z.B. slice

0.1.2 Elemente eines Modells

Um mit dem DMF Daten in Strukturen verschiedener Programmiersprachen darstellen zu können, müssen auch diese Abstrahiert werden.

Primitive Typen

Grundvoraussetzung sind die Primitiven Typen und Referenzen zu anderen Elementen. Mithilfe der Analyse der verschiedenen Variablentypen und Eigenschaften, wurden folgende Primitive Typen festgelegt:

// TODO : Tabelle mit Erklärung

Komplexe Datentypen

Als komplexe Datentypen werden Daten

Structured constructors. (+), types can be built up from these basic types by means of type. The type constructors in our language include function spaces Cartesian products (x), record types (also called labeled Cartesian products), and variant types (also called labeled disjoint sums).[?]

In nahezu allen Programmiersprachen gibt es die Möglichkeit, mit so genannten zusammengesetzten oder komplexen Datentypen zu arbeiten. Ihnen ist gemeinsam, dass wir mehrere Werte nebeneinander dort abspeichern können.[?]

Es gibt in jeder Programmiersprache ein Konstrukt mit dem mehrere Variablen zusammengefasst werden können. C nannte diese Konstrukte Structs. Diese Structs besitzen keine Identität.

Identität einer Instanz in der Datenbank

Ein Modell im DMF Framework soll in einer Datenbank gespeichert werden können. Dafür müssen Datenbankschlüssel definiert werden. Ein Schlüssel definiert die Identität eines Zeile in einer Tabelle. Diese Identität muss auch im Modell abgebildet werden. Das DMF fügt deshalb den Typen Ëntity"hinzu, welcher eine Identität besitzt. Er basiert auf dem Struct und kann somit Argumente, Referenzen und Funktionen beinhalten.

0.1.3 Zuweisungen der Abstraktionen

Damit diese Abstraktion genutzt werden kann, müssen für jeden abstrakten Typen im DMF eine Zuweisung in jeder Sprache festgelegt werden.

Java

// TODO: Tabelle mit Mapping // TODO: Erklärung von Datetimes