

Objektorientierung in Modula-3, „T-Notation“

INTERFACE-DATEI:

TYPE T <: OeffentlichesT

OeffentlichesT = OBJECT

```
METHODS
  Methode1 (Parameter1:TEXT, Pamameter2:TEXT):TEXT;
  Methode2 (Parameter1: T);
END;
```

Zuerst wird ein „T“ aus „Untertyp“ von einem Objekttyp definiert, den wir gleich definieren werden:
In dieser Definition werden dann die Methoden (und nur die!) inkl. aller Eingabe- und Ausgabeparameter definiert.

MODUL-DATEI:

REVEAL T = OeffentlichesT BRANDED OBJECT

```
Variable1:TEXT;
Variable2:TEXT;
Variable3:TEXT;
Variable4:TEXT;

OVERRIDES
  Methode1 := ProzedurFürMethode1;
  Methode2 := ProzedurFürMethode1;
END;
```

In der Modul-Datei wird T dann „revealed“, d.h. es wird genauer festgelegt, was T eigentlich ist:
In „Reveal“ werden

- 1.) alle Variablen des Objektes¹ deklariert
- 2.) es wird mit „OVERRIDE“ festgelegt, welche Prozeduren bei den entsprechenden Methoden aufgerufen werden.

.....

PROCEDURE ProzedurFürMethode1 (**self: T**, Parameter1:TEXT, Pamameter2:TEXT):TEXT = ...

Die Prozedur selbst muss dann die gleichen Parameter enthalten wie zugehörige die Methode im Interface!!!
Zusätzlich muss nur ein weiterer Parameter vom Typ T zugefügt werden: dies ist immer das Objekt selbst, welches der Prozedur übergeben wird.

¹ OBJEKT = RECORD mit Methoden

ADT's in Modula-3

INTERFACE-DATEI:

TYPE EigenerTyp <: REFANY

```
PROCEDURE Methode1 (o:EigenerTyp, Parameter1:TEXT, Parameter2:TEXT):TEXT;  
PROCEDURE Methode2 (o:EigenerTyp, Parameter1:TEXT, Parameter2:TEXT);
```

Hier definieren wir im Interface einen „eigenen Typ“ als Untertyp von Refany. Eine Variable dieses Typs wird dann an jede Prozedur übergeben.

MODUL-DATEI:

REVEAL EigenerTyp = BRANDED REF RECORD

```
Variable1:TEXT;  
Variable2:TEXT;  
Variable3:TEXT;  
Variable4:TEXT;  
END;
```

Daten zum Arbeiten haben muss.

In der Modul-Datei wird dieser „Eigene Typ“ dann revealed: Es wird festgelegt, welchen Datentyp er hat (hier RECORD...).

.....

```
PROCEDURE Methode1 (o:EigenerTyp, Parameter1:TEXT, Parameter2:TEXT)  
PROCEDURE Methode2 (o:EigenerTyp, Parameter1:TEXT, Parameter2:TEXT)
```

Danach stehen all die Prozeduren ausformuliert, mit denen man diesen abstrakten Datentyp bearbeiten kann. Hierbei ist darauf zu achten, dass immer ein Element dieses Datentyps als erster Parameter übergeben wird, da die Prozedur ja Daten zum Arbeiten haben muss.

DATENTYPEN:

```
INTEGER      -2^32 bis +2^31  
CARDINAL:   0 bis ...
```

+, -, *, DIV, MOD

= # < > <= >=

```
FIRST (type)  
LAST (type)  
INC (z)  
DEC (z)  
ABS (z)
```