

## Teil2: Semantikanschluss und Symboltabelle

Das Ziel der zweiten Übung ist, den MIECCompiler so weit zu erweitern, dass für alle Deklarationen entsprechende Symbole und Typen erzeugt werden, die miteinander so verkettet sind, dass keine Informationen verloren gehen. Die Symboltabelle stellt die Basis für die Zwischencodeerzeugung dar. Dazu fügen Sie in die `MIEC.atg` Attribute und semantische Aktionen ein und führen folgende Implementierungsschritte durch:

1. Erzeugung von Symbol- und Typknoten für alle Variablen und Typen.
2. Abbildung von numerischen Konstanten.
3. Aufbau einer Symboltabelle, die alle deklarierten Variablen, Typen und Konstanten speichert.
4. Prüfung der nötigen Kontextbedingungen in Deklarationen und Anweisungen:
  - Doppeldeklarationen sind nicht erlaubt.
  - Alle verwendeten Namen (Variablen) müssen deklariert sein.
  - Zuweisungskompatibilität: Typprüfung bei Zuweisung oder Vergleich von Variablen und Konstanten
5. Führen Sie eine Offsetberechnung für die deklarierten Variablen durch. Jede Variable speichert ihren Offset den sie später im Datensegment einnehmen wird.

**Hinweis:** Die Symboltabelle wird in der ATG folgendermaßen inkludiert und deklariert:

```
1 #include "SymbolTable.h"
2
3 COMPILER MIEC
4
5     SymbolTable mSymTab;
6
7     // helper methods
8     // ...
9
10 CHARACTERS
11 ...
12 TOKENS
13 ...
```

Durch diese Deklaration wird die Symboltabelle als Attribut in der Klasse `Parser` erzeugt, und somit kann direkt in den semantischen Aktionen auf die Symboltabelle zugegriffen werden. Auch eine Implementierung als Singleton ist möglich.