Projektaufgabe

Generator für Implementierungen von Verbundtypen:

Verbundtypen sollen durch eine einfache Sprache beschrieben werden, z. B.

```
import "String.h"
Adresse
      (name:  String;
       plz:   int;
       Stadt: String;
      )
```

Daraus soll ein **ANSI-C-Modul** mit Schnittstellen- und Implementierungsdatei erzeugt werden.
Er definiert

- den **Verbundtyp und seinen Pointertyp**,
- eine **Erzeugerfunktion** `newAdresse`, die ein Verbundobjekt erzeugt und den Pointer darauf liefert,
- für jede Komponente ein **Zugriffsmakro**, z. B.
  `#define nameOfAdresse(x) ((x)->name_fld)`
- **`#include`-Anweisungen** für die benutzten Typen

Vorgehen:

- **Zieltext für das Beispiel** aufschreiben,
- **Konkrete Syntax** der Sprache entwerfen,
- Aufgaben der **Namensanalyse** aufschreiben, spezifizieren,
- notwendige **Eigenschaften von Objekten** feststellen, spezifizieren,
- **Zieltext-Erzeugung** spezifizieren,
- Generator und seine Produkte **validieren**.

Zieltextbeispiel zur Projektaufgabe

Eingabeprogramm `adr`:

```
import "String.h"
Adresse
      (name:  String;
       plz:   int;
       Stadt: String;
      )
```

Ausgabedatei `adr.h`:

```
#include "String.h"

typedef struct __Adresse *Adresse;
typedef struct __Adresse {
  String name_fld;
  int plz_fld;
  String stadt_fld;
} _Adresse;

#define nameOfAdresse(x) ((x)->name_fld)
#define plzOfAdresse(x) ((x)->plz_fld)
#define stadtOfAdresse(x) ((x)->stadt_fld)

extern Adresse newAdresse ();
```

Ausgabedatei `adr.c`:

```
#include "adr.h"
#include <stdlib.h>

Adresse newAdresse ()
{ return (Adresse)malloc(sizeof(_Adresse));
}
```