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**.

Praktikum Sprachimplementierung mit Werkzeugen WS 1999/ 2000 / Folie 801

Ziele:

Projektaufgabe vorstellen

im Vorlesungsteil:

Beantwortung von Fragen dazu

Übungsaufgaben:

- Machen Sie sich mit der Aufgabenstellung vertraut.
- Erledigen Sie die ersten Entwurfsschritte bevor Sie an den Rechner gehen.
- Lösen Sie die Aufgabe erst vollständig, ohne sie mit zusätzlichem Komfort zu erweitern.

Verständnisfragen:

Wenn Sie Zeit und Interesse haben können Sie die Aufgabenstellung erweitern, und feststellen, ob Ihre Spezifikation einfach erweiterbar ist:

- Mehrere Verbunde in einer Datei beschreiben und implementieren.
- In der Implementierung obstack statt malloc verwenden.
- Weitere Typklassen implementieren: Vereinigungstypen (union), Folgen, Arrays.
- Auch E/A-Funktionen erzeugen. Damit haben Sie dann schon den Kern eines Werkzeuges wie IDL (Interface Definition Language).

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));
}
```

Praktikum Sprachimplementierung mit Werkzeugen WS 1999/ 2000 / Folie 802

Ziele:

Hilfe zum Zieltext für das Projekt

im Vorlesungsteil:

- auf Probleme der C-Implementierung hinweisen
- gemeinsame Diskussion von Lösungsmöglichkeiten