Struktogrammgenerator

Franz Fiala, N, TGM DSK-378, 379

Liebe EDV-geplagte Schüler- und LehrerInnen!

Ein übersichtliches Struktogramm ist ein Ausdrucksmittel mit dem der EDV-Fachmann auch mit Laien über Programmabläufe diskutieren kann. Dennoch sind sie bei den Schülern unbeliebt, weil das händische Zeichnen oft unflexibler ist als das Editieren des Codes selbst.

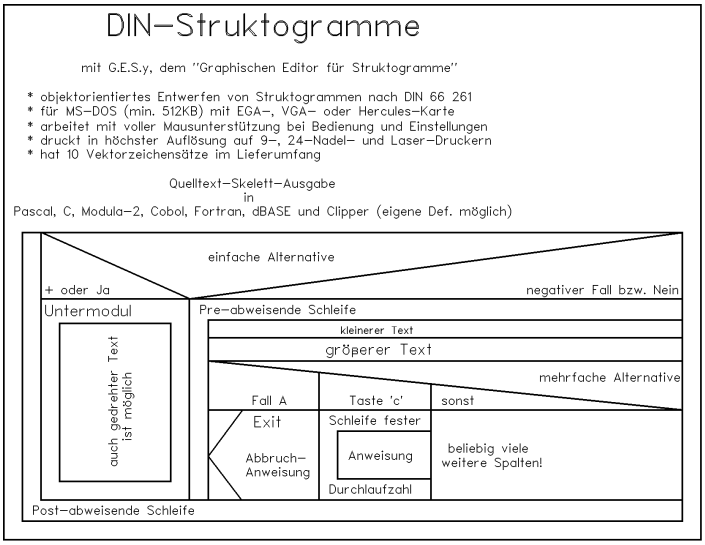
Auf der Suche nach geeigneten Hilfen stößt man zunächst auf den ABC-Flowcharter von Micrografix, der aber Flußdiagramme und nicht Struktogramme zeichnen kann; und Flußdiagramme sind ein Ausdrucksmittel, das man beim Programmentwurf vermeiden will.

Ich habe eingangs ein Struktogramm als Kommunikationsmittel zwischen Programmierer und Anwender erwähnt.

Struktogramme sind aber für den Lernenden auch ein Hilfsmittel zum Verständnis der Abläufe in Rechnerprogrammen. Wenn Struktogramme auch bei Programmierern weniger beliebt sind (wie mir einige kritische Leser dieses Beitrags berichteten), als didaktisches Hilfsmittel hat es den Charakter eines Bildschirmspiels bei dem unbemerkt Lerninhalte transportiert werden.

Neulich fand ich ein preiswertes DOS-Programm, über das ich hier berichten möchte und dessen Kauf sich für Schulen, Schüler und Lehrer, die mit Programmierung zu tun haben, lohnt:

G.E.S.y. V2.1 ist ein DOS-Programm mit Windows-ähnlicher Oberfläche, mit dem alle Stukurelemente mit drag-drop-Technik in den Entwurf mit entsprechenden Kommentaren eingebunden werden können. Dabei stellen sich die einzelnen Blöcke selbsttätig auf die Textlänge ein; das Einfügen einer Bedingung im letzten Moment wird zum Kinderspiel. Folgende Elemente kennt der Struktogrammgenerator (nicht abgebildet: Parallelverarbeitung):



Nur in DOS?

Wenn man es als Nachteil empfindet, daß das Programm nicht in einer Windows-Version vorliegt: man kann den Output in PCX, TIFF, PostScript oder HPGL exportieren, sodaß die Weiterverarbeitung der Grafik auch in Windows problemlos möglich ist.

Bezugsquelle:

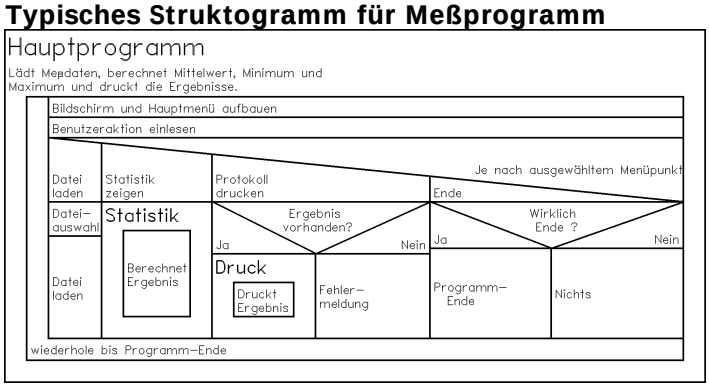
SIP-Software-Lösungen
Griesäckerstrasse 15
D-96117 Memmelsdorf
TEL:0951/43489, FAX:0951/420514
Preis: DM 200,- (VISA/EUROCARD)

Schutz

Das Programmpaket kann jeder benutzen, es installiert sich als DEMO-Version mit folgenden Einschränkungen: Maximal zwei-zeiliger Kommentar und nicht mehr als vier gleichartige Blöcke hintereinander. [5 Anweisungsblöcke hintereinander geht nicht, dagegen können ineinander verschachtelte Stukturen, wie Schleifen, Entscheidungen usw. gut geprobt werden]. Zum Lernen und für einfache Dinge ausreichend, zum Arbeiten zu wenig. Erst mit einer persönlichen Kennung, die man einfach als ASCII-Text in eine Datei schreibt, die aber ein daraus berechnetes Schutzwort enthält, wird es eine lizensierte Version ohne Einschränkungen. *Anmerkung: Die folgenden Struktogranne und alle Struktogramme des Beitrags "Hardwarenahe Programmierung mit C" wurden mit der Demo-Version erstellt.*

Daher: Wer das Programm kennenlernen möchte, der kann die Diskette über die Mailbox His Master's Voice bestellen oder downloaden. Die Kosten des Programm sind gering im Vergleich zum didaktischen Wert.

Typisches Struktogramm für Meßprogramm



Templates

Besonders hübsch ist, daß man aus dem Struktogramm ein Programmgerüst ableiten kann (Templates für C, PASCAL, COBOL, BASIC, REXX oder dBASE sind vorhanden). Die jeweiligen Schlüsselworter sind in reinen ASCII-Dateien abgelegt, man kann die entstehenden Programmsequenzen auch an andere Sprachen selbst anpassen. Ein komplettes Beispiel sehen Sie auf der folgenden Seite.

Fortsetzung

Und was ist mit den vielen, bis jetzt undokumentierten Programmen? Auch dafür gibt eine Lösung, allerdings kostet die DM 600,-: Dieses Programm generiert aus einem vorhandenen Listing ein Struktogramm. Einige Schüler planen schon einen Kauf!

Real programmers don't write specs. Users should consider themselves lucky to get any programs at all and take what they get.

Real programmers don't comment their code. If it was hard to write, it should be hard to read.

Real programmers don't write application programs, they pro- gram right down on the bare metal. Application programming is for feebs who can't do systems programming.

Real programmers don't eat quiche. **Real programmers** don't even know how to spell quiche. They eat Twinkies, Coke and palate-scorching Szechwan food.

Real programmers don't draw flowcharts. Flowcharts are, after all, the illiterate's form of documentation. Cavemen drew flowcharts; look how much it did for them.

Real programmers don't read manuals. Reliance on a reference is a hallmark of the novice and the coward.

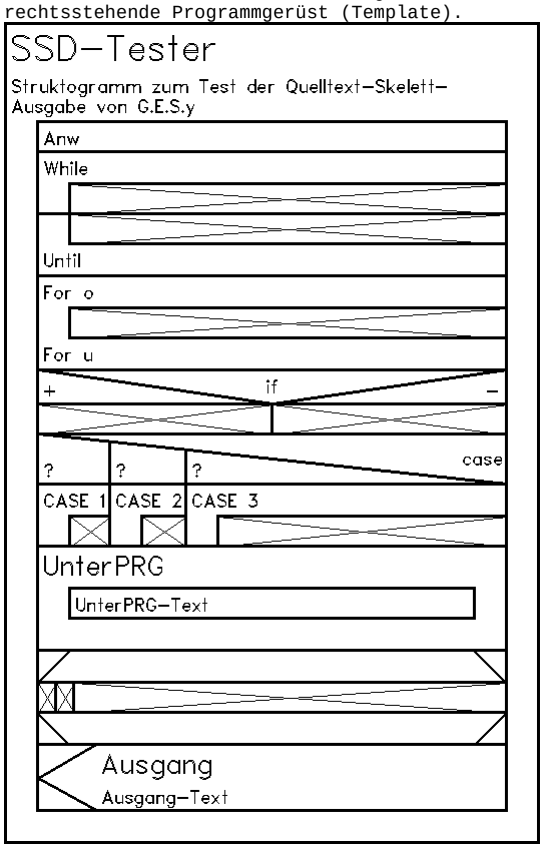
Templates

Ein Template ist ein Programmgerüst, das man aus einem Struktogramm für viele Sprachen ableiten kann.

Zunächst muß eine Anweisungsdatei (Beispiel für ANSI-C) vorhanden sein, die die Anpassung an jede Sprache und an jeden persönlichen Schreibstil erlaubt:

```
; Quelltextskelett-Übersetzungsdatei für ANSI-C
; Parameter :
;   \$      Objektnamen einfügen
;   \#      Objekttext einfügen
;   \+      oberen Text eines FOR-Objektes einfügen
;   \@      Sohnliste hier einfügen
;   \%      Sohnliste unter einem
;           Fallunterscheidungsobjekt einfügen
;   \iNAME Datei NAME einfügen (nur Großbuchstaben!)
;   \\      das \-Zeichen
;   \?      gibt eine Meldungsbox aus (für in einer Sprache
;           undefinierte Objekte
;   \n      neue Zeile
;   \&xxx   in der Spalte weitermachen (Kommentarbeginn)
;   \>      eine Tabulator-Position einrücken
;   \<      eine Tabulator-Position ausrücken
;   \0..\9 Label-Nummern - pro Objekt 10 Stück
;   \p      Parameter-Liste
;           (nur bei Procedur- und Root-Sinnbild)
*.C
 \i FILEANSI.HDR \n
 \i FUNCANS.HDR \n/* \# */\n\nvoid\n\$(\p)\n\n{\>\n\@\<\n} /*
end of function \$ */\n\n\n
/***** Unknown Statement: \# */\n
/* \# */\n
\n/* \# */\nwhile (0) {\>\n\@\<}  /* end while */\n
\n/* \# */\ndo {\>\n\@\<} while (0);       /* end do */\n
\n/* \#\n\+ */\nfor (;;) {\>\n\@\<}        /* end for */\n
\n/* \# */\nif (0) {\>\@\<} else {\>\@\<} /* end if */\n
\n/* \# */\nswitch (0) {\>\n
\n\<}    /* end switch */\n
\n
\ncase 0:        /* \# */\>\n\%break;\<\n
\n/* \# */\n\$(\p);\n
\n/* \# */\n\$;\n
{        /* begin PARALLEL */\>\n
\<}      /* end PARALLEL */\n
{\>\n/* \# */\<\n}\n
```

Danach formuliert man das Struktogramm und erhält daraus das rechtsstehende Programmgerüst (Template).




```
 /*
############################################################
  NAME        :
  SYNOPSIS    :
  DESCRIPTION :
  RETURN VALUE:
  EXAMPLE     :
  SEE ALSO    :
############################################################
*/
/* Struktogramm zum Test der Quelltext-Skelett-
   Ausgabe von G.E.S.y */

void
SSD-Tester(PARAMETER)
{
        /* Anw */

        /* While */
        while (0) {
                /***** Unknown Statement: * */
        }        /* end while */

        /* Until */
        do {
                /***** Unknown Statement: * */
        } while (0);        /* end do */

        /* For u */
        for (;;) {
                /***** Unknown Statement: * */
        }        /* end for */

        /* if */
        if (0) {
                /***** Unknown Statement: * */
        } else {
                /***** Unknown Statement: * */
        }        /* end if */

        /* case */
        switch (0) {

                case 0:        /* ? */

                        /* CASE 1 */
                        while (0) {
                                /***** Unknown Statement: * */
                        }        /* end while */
                        break;

                case 0:        /* ? */

                        /* CASE 2 */
                        while (0) {
                                /***** Unknown Statement: * */
                        }        /* end while */
                        break;

                case 0:        /* ? */

                        /* CASE 3 */
                        while (0) {
                                /***** Unknown Statement: * */
                        }        /* end while */
                        break;

        }        /* end switch */

        /* UnterPRG-Text */
        UnterPRG(PARAMETER);
        {        /* begin PARALLEL */
                {
                        /* * */
                }
                {
                        /* * */
                }
                {
                        /* * */
                }
        }        /* end PARALLEL */

        /* Ausgang-Text */
        Ausgang;

}        /* end of function SSD-Tester */
```