8. Übung zur
Konstruktion fehlerfreier Software

Frank Pilhofer

Montag 13-14 Uhr

0. Umsetzung des Pascal-Programms

Original

```
1  program printsquares;
2  procedure square ( par1:integer;
3                     var par2:integer);
4  begin
5    par2 := par1 * par1;
6    par1 := -10
7  end;
8
9  var n,f:integer;
10
11 begin
12   writeln (n,' ', f,' Erste Zeile');
13
14
15   n:=1;
16   while n≤10 do
17   begin
18     square (n,f);
19
20
21
22
23     writeln (n, ' ', f);
24     n:=n+1;
25   end
26 end.
27
28
```

Umsetzung

```
procedure square

  par2 := par1 * par1;
  par1 := -10;
endprocedure

declare (n, Z[1], unbekannt[2])
declare (f, Z,  unbekannt)
program
  scr[3](zeile[4])=str[5](n)&' '&
                   str(f)&' Erste Zeile'
  zeile := zeile + 1
  n := 1
  while n≤10 do

    declare (par1, Z, n)
    declare (par2, Ref, &f[6])
    call square
    release par2
    release par1
    scr(zeile)=str(n)&' '&str(f)
    n:=n+1
  endwhile
endprogram
release f
release n
```

1. Auswirkungen der Änderungen

1.0. **procedure** square (**var** par1: **integer**; **var** par2 : **integer**);

Mit dieser Änderung würde auch die Variable *n* bzw. *par1* an die Unterfunktion *square* als Referenz übergeben (Zeile 18 würde zu "`declare (par1, Ref, &n)`". Das hätte zur Folge, daß bei jedem Aufruf die Variable des Hauptprogrammes auf den Wert -10 zurückgesetzt würde. Nach der "ersten Zeile" würde das Programm noch einmal die 1 und das Quadrat von 1 drucken, danach nur noch die Zeile "`-9 81`" in einer Endlosschleife.

---

1 Hier kann man sich natürlich streiten, ob der Zahlenbereich der Variablen ganz Z ist; Der Pascal-Datentyp "integer" ist schließlich auch beschränkt.

2 Sofern Variablen nicht explizit initialisiert werden, ist ihr Anfangswert undefiniert.

3 Die Variable *scr* hält die Textzeilen des Bildschirms

4 *zeile* bezeichnet die momentane Textzeile.

5 Die Funktion *str* wandle eine Zahl in einen String.

6 *&f* sei eine Referenz auf *f* für die by-reference übergebene Variable *par2*.

1.1. **procedure** square (par1: **integer**; par2 : **integer**);

Mit dieser Änderung wird die Variable *f* bzw. *par2* by-value an die Unterfunktion übergeben. Zeile 19 würde zu "**declare** (par2, Z, f)". Das hätte zur Folge, daß das Quadrat von *n* zwar noch richtig berechnet wurde, dieser Wert aber nicht im Hauptprogramm abgefragt werden kann. Für *f* wird immer der gleiche uninitialisierte Wert ausgegeben, während *n* korrekt hochzählt und die Schleife terminiert.

2. Basic-Programm

Original

```
1000 REM Programmanfang
1010 n=13
1020 s=1
1030 w=1
2000 if n≤0 then goto 4000
2010 gosub 5000
3000 n=n-w
3010 s=3-s
3090 goto 2000
4000 print"Das Spiel ist zu Ende.
          Spieler";s;"hat gewonnen."

4010 end



5000 ...
...
5080 w=...
5090 return
```

Umsetzung

```
program
  declare (n, Z[7], 13)
  declare (s, Z, 1)
  declare (w, Z, 1)
  while n>0 do
    call 5000
    n := n-w
    s := 3-s
  endwhile
  scr(z)='Das Spiel ist zu Ende.'
        &'Spieler'&str(s)
        &'hat gewonnen'
  release w
  release s
  release n
endprogram

procedure 5000
  ...
  w := ...
endprocedure
```

[7] In vielen Basic-Implementationen nehmen undeklarierte Variablen auch oft den Typ "Real" an, was teilweise daran liegt, daß dies der einzige verfügbare numerische Typ ist (z.B. MS Basic 2/3.5/7 auf Commodore-Computern).