Übungen zu Programmiersprachenkonzepte

Wintersemester 2012/2013

Dozent: Dr.Jutta Göers

HiWi: C. Wassmuth

Übungsblock 1:
eine einfache Zählschleife in verschiedenen Sprachen

Zählschleife konzeptuell:

- Spezifikation einer Schleifenvariablen
- einem Anfang- und Endwert für diese Variable
- einem Schleifenrumpf

In abstrakter Syntax:

```
for-statement(index,start,ende,rumpf)
```

Übungsblock 1

FORTRAN:

```
      do 1 i=1,10
1     a(i) = 0
```

- in do-Zeile Anweisungsnummer, die Ende des Schleifenrumpfs anzeigt
- Vektor a, die ersten 10 Elemente werden auf 0 gesetzt

Übungsblock 1

Basic:

```
for i=1 to 10
   let a(i)=0
next i
```

- Syntax besser lesbar
- Was wäre, wenn wir (ausversehen) `next j` geschrieben hätten?
- → fehleranfälliger, da Schleifenzähler selber hochzuzählen ist

Übungsblock 1

Pascal:

```
for i:=1 to 10 do a[i]:=0;
```

- Schleifenrumpf auf eine Anweisung begrenzt
- braucht man mehrere Anweisungen: klammern durch
  `begin … end`
- Anweisungsende `;`

Übungsblock 1

Modula-2:

```
for i:=1 to 10 do a[i]:=0 end
```

- Modula-2 eindeutig Abkömmling von Pascal
- hebt Einschränkung auf eine Schleifenrumpf-Anweisung von Pascal wieder auf

Übungsblock 1

COBOL:

```
PERFORM INIT VARYING i FROM 1 BY 1
UNTIL i>10.
…
INIT.
MOVE ZERO TO A(i).
```

- Trennung von Schleifenkopf und Schleifenrumpf entspricht Prinzip der schrittweisen Verfeinerung,
- eher unübersichtlich (vor allem bei einfachen Rümpfen),
- Spezifikation einer Schrittweite möglich

Übungsblock 1

C:

```
for (i=1;i<=10;i++) a[i]=0;
```

- statt Endwert für die Schleifenvariable wird hier Bedingung für *Verbleib* in Schleife angegeben
- neu: Aktion, die am Ende jedes Schleifendurchlaufs ausgeführt wird (hier: Inkrement)
- Schleifenvariable, Initialisierung, Bedingung und Aktion müssen nicht in direktem Zusammenhang stehen
- → keine Zählschleife, sondern initialisierte while-Schleife

Übungsblock 1

APL:

```
a ← 10 p 0
```

- Bedeutung: Lege Feld a mit 10 Elementen an und initialisiere diese mit 0
- Wirkung wie bei den anderen Sprachen, aber keine Schleife!
- gleichzeitig Feld a deklariert und Speicher allokiert

Übungsblock 1

Fazit:

Diskussion über Konzepte von Programmier-

sprachen und deren Umsetzung innerhalb einzelner

Sprachen

- Konzept auf höchstmöglicher Abstraktionsstufe beschreiben (abstrakte Syntax)
- entsprechende konkrete Syntax einer Sprache dazu in Beziehung setzen
- bewerten, in welchem Unfang Sprache das Konzept verwirklicht (Einschränkungen, Erweiterungen?)