

MODULA – 2

Grundlagen Wissenschaftlichen Arbeitens

Hilal Tekoglu



Modula - 2

- Historische Entwicklung
- Die Sprache Modula-2
- Syntax und Semantik
- Beispiel: PrimZahlen + Stack
- Zusammenfassung
- Referenzen

Historische Entwicklung

- Nikolaus Wirth – 1978
- N.Wirth: Algol, Pascal, Modula, Oberon, Lilith Computer
- PDP 11 entwickelt – 16 Bit Computer (Geissmann, Knudsen, Jacobi)
- Weiterentwicklung von Pascal
- Nachfolger Oberon
- ETH – Eidgenössische Technische Hochschule
- John von Neumann – Computerpionier

Die Sprache Modula – 2

- **Mängel von Pascal**
 - schwaches File I/O Konzept
 - Einfache Laufzeitbibliothek
- **Modul Struktur**
 - Objekt kontrollieren
 - Separate Programmabschnitte
- **Information Hiding Prinzip**
 - Frühen 70er erwähnt
 - Durch Modula – 2 umgesetzt
 - Erstmals von David Parnas

Die Sprache Modula – 2

- Maschinenunabhängig
- System Entwicklungen
 - Allgemeine, effiziente Systemprogrammiersprache für Microcomputer
- Leicht lesbar, lernbar, verständlich
- Lehrmittel in Schulen

Syntax und Semantik

- case – sensitive
- Bezeichner und Schlüsselwörter – reservierte Wörter
 - ELSE, DEFINITION, EXIT, IMPORT
- Modula – 2:
 - Wertzuweisungen
 - IF-, CASE-, WHILE-, REPEAT-, FOR-, WITH-, RETURN Anweisungen
 - Prozeduraufruf
- Kommentar – (*ich bin ein Kommentar*)

Syntax und Semantik

- Module
 1. Programm Modul
 2. Definitionsmodul
 3. Implementierungsmodul
- Programm Modul – Hauptmodul für andere Module
- Definitionsmodul – Deklarationen
- Implementierungsmodul – Beginnt mit dem Schlüsselwort *IMPLEMENTATION MODULE*

Beispiel 1: PrimZahlen

- ```
MODULE PrimZahlen;
FROM InOut IMPORT WriteString, WriteInt, WriteLn;
CONST Max = 100; (* 2 < Primzahlen < 2*Max+1 *)
VAR Zahlen : ARRAY [1..Max] OF BOOLEAN;
 I,J : INTEGER;
BEGIN
 WriteString(,Primzahlen',0);
 WriteLn;
 FOR I:= 1 TO Max DO Zahlen[I]:= TRUE END;
 FOR I:= 3 TO (2*Max+1) DIV 3 BY 2 DO
 FOR J:= 3 TO (2*Max+1) DIV I BY 2 DO
 Zahlen[I*J DIV 2] := FALSE
 END
 END;
 J:= -1;
 FOR I:=1 TO Max DO
 IF Zahlen[I] THEN
 J:= J+1;
 WriteInt(2*I+1,6);
 IF J MOD 10 = 9 THEN WriteLn END
 END
 END; WriteLn
END Primzahlen.
```



# Beispiel 2: Stack

- DEFINITION MODULE Stack;  
TYPE  
    Stack = RECORD  
    top: INTEGER;  
    value: ARRAY [0..999] OF INTEGER;  
END;  
PROCEDURE Clear (VAR s: Stack);  
PROCEDURE Push (VAR s: Stack; i: INTEGER);  
PROCEDURE Pop (VAR s: Stack; VAR i: INTEGER);  
END Stack.  
IMPLEMENTATION MODULE Stack;  
    PROCEDURE Clear (VAR s: Stack);  
    BEGIN  
        s.top := 0;  
    END Clear;  
    PROCEDURE Push (VAR s: Stack; i: INTEGER);  
    BEGIN  
        IF s.top <= 999 THEN  
            s.value[s.top] := i; INC(s.top);  
        END;  
    END Push;  
    PROCEDURE Pop (VAR s: Stack; VAR i: INTEGER);  
    BEGIN  
        IF s.top > 0 THEN  
            DEC(s.top);  
            i := s.value[s.top];  
        END; END Pop;  
    END Stack.

# Zusammenfassung

- Einführung des Modul Konzepts
- System Entwicklungssprache
- Problemzerlegung und Verfeinerung
- Realisierung grosser Projekte und Wiederverwendbarkeit
- Source Code leicht lesbar
- Programmiersprache leicht lernbar

# Referenzen

1. Nikolaus Wirth , History and Goals of Modula-2, Byte , 1984
2. Nikolaus Wirth, Modula-2, ETH Zürich, 1980
3. R.J.Paul, An Introduction to Modula-2, Byte, 1984
4. J.Gutknecht , Tutorial on Modula-2, Byte, 1984
5. J.Gutknecht, Programming Languages and System Architectures, Springer, 1994
6. <http://www.modula2.org>

# Modula – 2

Danke für Ihre Aufmerksamkeit!

