

Einführung in die Modula-3 Programmierumgebung

- SRC Modula-3
- Kommandos und Dateien
- Ein einfaches Beispiel
- Importieren von Modulen
- Hinweise zu den Übungsprogrammen
- Probleme - Mögliche Ursachen

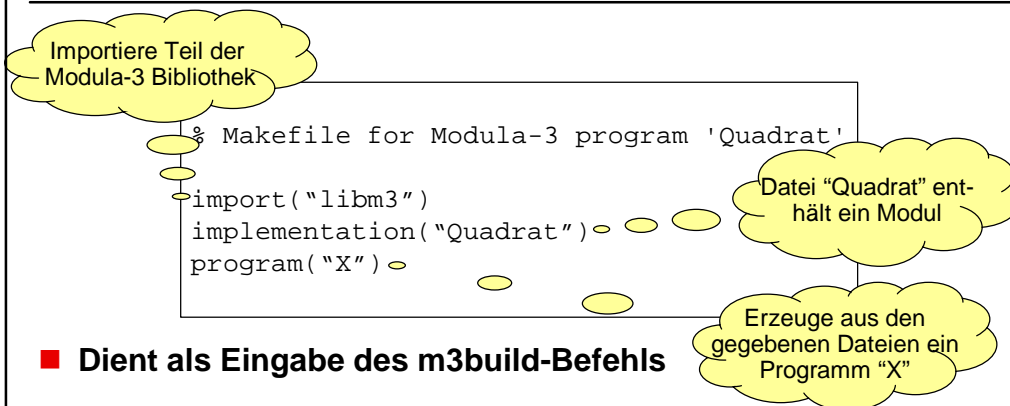
SRC Modula-3

- Entwickelt vom DEC Systems Research Center (SRC)
- Umfaßt den Übersetzer (Compiler)
 - prüft die Syntax des Programmtexts
 - übersetzt ihn in durch den Rechner ausführbaren Code
- Und eine ganze Reihe vorgefertigter Bibliotheken
 - für die Ein-/Ausgabe von Daten (z.B. das Modul IO)
 - um Zeichenketten zu manipulieren (das Modul Text)
 - ...
- Zu dem ist eine Laufzeitumgebung enthalten
 - die den Rahmen für ein ausführbares Programm stellt
 - und beispielsweise die Speicherverwaltung übernimmt.
- SRC Modula-3 ist Referenz für andere Modula-3 Umgebungen

m3build

- **Der Befehl, um ein Modula-3 Programm zu erzeugen.**
- **Theoretisch kann mit “m3”-Befehl der Modula-3 Übersetzer direkt gestartet werden:**
 - Erfordert umständliche Eingabe von Pfaden
 - Sowie genaue Kenntnis der Übersetzer-Optionen
- **m3build $\hat{=}$ m3makefile**
 - m3build sucht Datei mit Namen “m3makefile”
 - Versucht darin enthaltene Informationen auszuwerten
 - Und damit die Übersetzung vorzunehmen.
- **Vergleichbar dem “Make” aus der C-Programmierung:**
 - jedoch verbesserte Analyse der Abhängigkeiten (zwischen den Dateien)
 - erlaubt kleinere, einfachere und besser portierbare “makefiles”

m3makefile



- **Dient als Eingabe des m3build-Befehls**
- **Hat immer den Namen “m3makefile”**
- **Gibt Informationen**
 - wo welche Dateien abgelegt sind
 - und was damit zu tun ist

Organisation in Packages

- SRC Modula-3 ist in "Packages" organisiert.
- Verzeichnis "src" (Source) enthält Programmtext und "m3makefile"
- Plattformabhängiges Verzeichnis (NT386) umfaßt ausführbares Programm und neu erzeugten Objektcode
- Erlaubt sinnvolle Organisation sehr großer Projekte

```

graph TD
    Quadrat[Quadrat] --> src[src]
    Quadrat --> NT386[NT386]
    src --> Qm3[Quadrat.m3]
    src --> m3makefile[m3makefile]
    NT386 --> Qexe[Quadrat.exe]
    NT386 --> Qmo[Quadrat.mo]
    
```

Moritz Schnizler 1998, RWTH Aachen Einführung in die M3- Programmierumgebung - 7 -

Ein kleines Beispiel - Package

Problem: Programm, das eine Zahl quadriert (ohne Ausgabe)

- Erzeuge Verzeichnis mit Name des Packages:

```
... > mkdir Quadrat
```

- Und Unterverzeichnis mit Namen "src":

```
... > cd Quadrat
Quadrat > mkdir src
Quadrat > cd src
```

Moritz Schnizler 1998, RWTH Aachen Einführung in die M3- Programmierumgebung - 8 -

Ein kleines Beispiel - Programmtext

- **Editiere den Programmtext:** `src > emacs Quadrat.m3`
- **Speichere ihn im Verzeichnis "src"**

```

MODULE Quadrat EXPORTS Main;
(* Dieses Programm berechnet das Quadrat einer Zahl.

Autor      : Moritz Schnizler, RWTH Aachen
Umgebung   : SRC-Modula-3 rel. 3.6, Windows NT 4.0
Erstellt   : 30.10.98 Letzte Aenderung: 30.10.98 *)

PROCEDURE Quadriere(x: INTEGER): INTEGER =
BEGIN
  RETURN x*x; (* Gib das Quadrat der Zahl x zurueck *)
END Quadriere;

BEGIN
  IF (Quadriere(5) = 25) THEN
    (* Ist das Ergebnis richtig! *)
  ELSE
    (* Ist das Ergebnis falsch! *)
  END;
END Quadrat.
        
```

Implementiert
Hauptprogramm

Moritz Schnizler 1998, RWTH Aachen Einführung in die M3- Programmierumgebung - 9 -

Ein kleines Beispiel - m3makefile

- **Editiere die Datei "m3makefile":** `src > emacs m3makefile`
- **Speichere sie im Verzeichnis "src"**

```

% Makefile for Modula-3 program 'Quadrat'

import("libm3")
implementation("Quadrat")
program("Quadrat")
        
```

Importiere Basisfunktionalität aus Modula-3 Bibliothek

Name des erzeugten Programms

Name der zu übersetzenden Datei

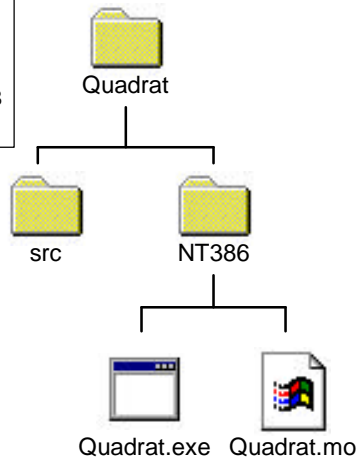
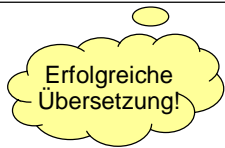
Moritz Schnizler 1998, RWTH Aachen Einführung in die M3- Programmierumgebung - 10 -

Ein kleines Beispiel - m3build

■ **Übersetze Programmtext in das ausführbare Programm:**

```
src > m3build
--- building in ..\NT386 ---
m3 -wl -why -g -o Quadrat -FC:\TEMP\qk3

new source -> compiling ..\src\Quadrat.m3
new "Quadrat.mo" -> linking Quadrat.exe
```

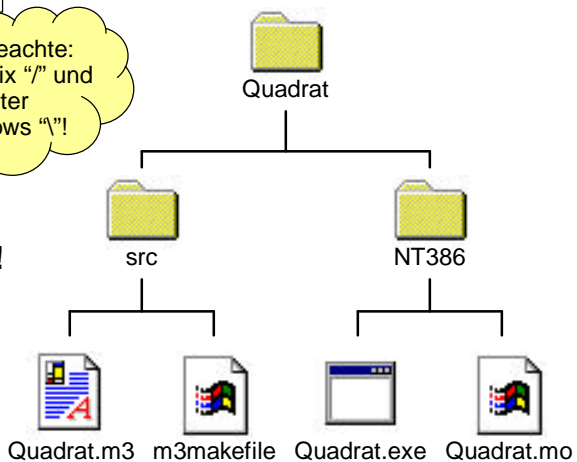
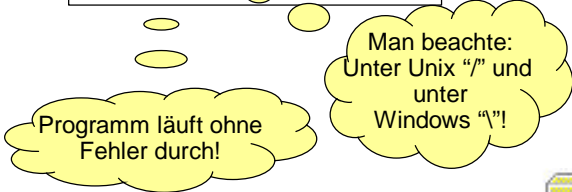


■ **Plattformabhängiges Verzeichnis (NT386 od. SOLgnu) mit ausführbarem Programm wird erzeugt.**

Ein kleines Beispiel - Test

■ **Versuche Programm auszuführen:**

```
src > ..\NT386\Quadrat
src >
```



■ **Achte auf Laufzeitfehler!**

Ein Fehler

```
PROCEDURE Quadriere(x: INTEGER): INTEGER =
BEGIN
...
END Quadrier;
```

```
--- building in ..\NT386 ---
m3 -wl -why -g -o Quadrat -FC:\TEMP\qk3

new source -> compiling ..\src\Quadrat.m3
"..\src\Quadrat.m3" line 14: Initial name
doesn't match final name (Quadriere)
1 error encountered
compilation failed => not building program "Quadrat"
```

Datei und Zeile,
wo das Problem
entdeckt wurde!

■ **Der Übersetzer gibt einen Fehler aus:**

- wenn eine Unstimmigkeit entdeckt wird
- und die vollständige Übersetzung daher nicht mehr möglich ist

■ **Ein guter Übersetzer lokalisiert Fehler sehr genau.**

Eine Warnung

Datei und Zeile,
wo das Problem
entdeckt wurde!

```
--- building in ..\NT386 ---
m3 -wl -why -g -o Quadrat -FC:\TEMP\qk3

new source -> compiling ..\src\Quadrat.m3
"..\src\Quadrat.m3", line 23: warning: file name
(Quadrat.m3) doesn't match module name (Quadrata)
1 warning encountered
new "Quadrat.mo" -> linking Quadrat.exe
_m3main.c
```

■ **Der Compiler gibt eine Warnung aus:**

- wenn eine Unstimmigkeit entdeckt wird
- die vollständige Übersetzung jedoch noch möglich ist

■ **Warnungen sollten geprüft und nie ignoriert werden**

⇒ **Zeigen an, daß irgendetwas nicht stimmt!**

Beispiel Import eines Moduls - Programm

Problem: Lagere Prozedure "Quadriere" in Modul A aus

```

MODULE Quadrat EXPORTS Main;
(* Dieses Programm berechnet das Quadrat einer Zahl.
Autor           : Moritz Schnizler, RWTH Aachen
Umgebung       : SRC-Modula-3 rel. 3.6, Windows NT 4.0
Erstellt       : 30.10.98 Letzte Aenderung: 30.10.98 *)

IMPORT A; (* Importiere Funktion Quadriere aus Modul A *)

BEGIN
  IF (A.Quadriere(5) = 25) THEN
    (* Ist das Ergebnis richtig! *)
  END;
END Quadrat.
    
```

Importiere
Inhalt von
Modul A

Benutze
Funktion aus
Modul A

Beispiel Import eines Moduls - Modul

■ Schnittstellendefinition für das Modul A:

```

INTERFACE A;
  PROCEDURE Quadriere(x: INTEGER): INTEGER;
  (* Quadriere x und gib das Resultat zurück *)
END A.
    
```

■ Implementierung des Moduls A:

```

MODULE A;

PROCEDURE Quadriere(x: INTEGER): INTEGER =
BEGIN
  RETURN x*x; (* Gib das Quadrat der Zahl x zurueck *)
END Quadriere;

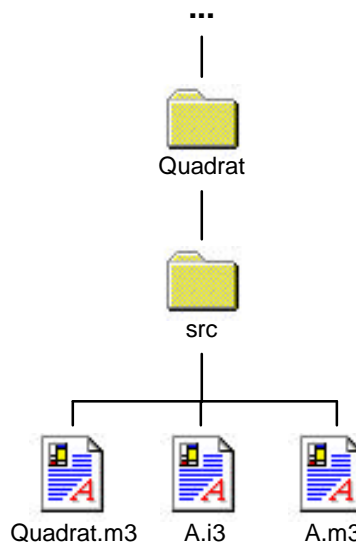
BEGIN
END A.
    
```


Beispiel Import eines Moduls - Package

- **Editiere die Dateien „Quadrat.m3“, „A.i3“ und „A.m3“:**

```
src > emacs Quadrat.m3
src > emacs A.i3
src > emacs A.m3
```

- **Speichere alle im Verzeichnis „src“**



Beispiel Import eines Moduls - m3build

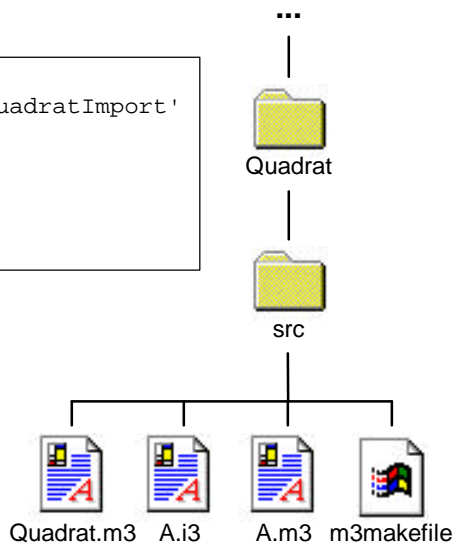
- **Editiere „m3makefile“: src > emacs m3makefile**

```
% Makefile for Modula-3 program 'QuadratImport'

import("libm3")
module("A")
implementation("Quadrat")
program("Quadrat")
```

Deklariert
Schnittstelle A.i3
und Modul A.m3

- **Übersetze mit m3build**



Import eines Bibliothekmoduls

■ **Importiere das Modul im Programm:**

```
IMPORT Text; (* Importiere Funktionen und Prozeduren
              zur Textmanipulation *)
```

■ **Importiere passenden Teil der Bibliothek im "m3makefile":**

```
% Makefile for Modula-3 program 'QuadratImport'

import("libm3")
module("A")
implementation("Quadrat")
program("Quadrat")
```

M3-Standard-
bibliothek enthält auch
Modul Text

Programme mit Ein-/Ausgabe

■ **Benutzen Sie das Modul SIO (Standard Input/Output) von unseren WWW-Seiten:**

<http://www-lufgi3.informatik.rwth-aachen.de/Programmierung>

■ **Importieren Sie das Modul SIO analog dem Modul A:**

- Speichern Sie die Dateien (SIO.i3, SIO.m3) bei Ihrem Programm im Verzeichnis "src"
- Ergänzen Sie das m3makefile um den Eintrag "module("SIO")":

```
% Makefile for Modula-3 program 'QuadratSIO'

import("libm3")
module("SIO")
module("A")
implementation("QuadratSIO")
program("QuadratSIO")
```

- Importieren Sie das Modul SIO mit der Anweisung "IMPORT SIO" in Ihrem Programm

Beispiel mit Ausgabe

■ Das Beispielprogramm mit Ausgabe:

```

MODULE QuadratSIO EXPORTS Main;
(* Dieses Programm berechnet das Quadrat einer Zahl mit Ausgabe.
  Autor      : Moritz Schnizler, RWTH Aachen
  Umgebung   : SRC-Modula-3 rel. 3.6, Windows NT 4.0
  Erstellt  : 30.10.98  Letzte Aenderung: 30.10.98 *)

IMPORT SIO; (* Importiere Ein-/Ausgabefunktionen und -prozeduren *)
IMPORT A;   (* Importiere Funktion Quadriere aus Modul A *)

BEGIN
  SIO.PutText("5*5 = ");
  SIO.PutInt(A.Quadriere(5)); (* Gib berechnetes Ergebnis aus *)
  SIO.Nl();                  (* Gehe in neue Zeile (new line) *)
END QuadratSIO.
    
```

■ Probelauf:

```

src > ..\NT386\QuadratSIO
5*5 = 25
src >
    
```

Ausgabeprotokoll

■ Ausgabe einer Programmausführung protokollieren:

```

src > ..\NT386\FormatDatum > Ausgabe.txt
Geben Sie das heutige Datum ein: 13.04.97
Heute ist der 13.04.97!
src >
    
```

Kursiver Text
wird nicht am
Bildschirm
ausgegeben

Schreibe
Ausgabe in Datei
Ausgabe.txt

■ Inhalt der Datei Ausgabe.txt:

```

Geben Sie das heutige Datum ein:
Heute ist der 13.04.97!
    
```

Man beachte:
Benutzereingaben
erscheinen hier
nicht!

Probleme - Mögliche Ursachen 1

```
--- building in NT386 ---  
m3build: cannot locate an m3makefile
```

- Datei "m3makefile" existiert nicht oder hat einen falschen Namen
- Zu übersetzende Dateien sind nicht im "src"-Verzeichnis.

```
--- building in ..\NT386 ---  
m3 -wl -why -g -o Quadrat -FC:\TEMP\qk3  
  
unable to open file for reading: ..\src\Nschild.m3: ErrorCode=2  
compilation failed => not building program "Quadrat"
```

- Datei "Nschild.m3" existiert nicht
- Befindet sich am falschen Ort (z.B. nicht im "src"-Verzeichnis)
- Hat einen falschen Namen

Probleme - Mögliche Ursachen 2

```
--- building in ..\NT386 ---  
m3 -wl -why -g -o Quadrat -FC:\TEMP\qk3  
  
new source -> compiling ..\src\Quadrat.m3  
"..\src\Quadrat.m3", line 11: unable to find interface (SIO)  
1 error encountered  
compilation failed => not building program "Quadrat"
```

- Die Dateien "SIO.m3" und "SIO.i3" sind nicht im "src"-Verzeichnis
- In der Datei "m3makefile" fehlt der Eintrag "module ("SIO")"

Zusammenfassung

- **Verwenden Sie m3build und einen passend konfigurierten “m3makefile” für die Übersetzung.**
- **Achten Sie auf die Organisation in Packages, insbesondere das Verzeichnis “src”.**
- **Benutzen Sie in den Übungen das Modul SIO für Ein- und Ausgaben.**
- **Protokollieren Sie (soweit möglich) die Ausgabe Ihrer Programme und geben diese mit ab.**
- **Achten Sie auf ordentlichen Programmierstil:**
 - Kommentare!!
 - Einrückungen
 - einheitliche Bezeichnerwahl
 - ...