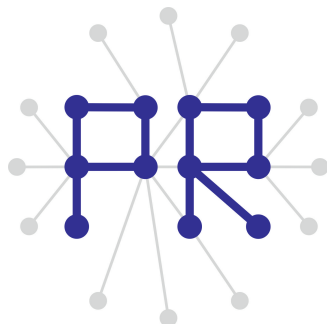
Einführung in die Informatik I

Kapitel I.7: Fehlersuche in Programmen

Prof. Dr. Marcin Grzegorzek[1]

Research Group for Pattern Recognition
www.pr.informatik.uni-siegen.de

Institute for Vision and Graphics
University of Siegen, Germany



[1] Die im Rahmen dieser Lehrveranstaltung verwendeten Lernmaterialien wurden uns zum Großteil von Herrn Prof. Dr. Wolfgang Wiechert und Herrn Prof. Dr. Roland Reichardt zur Verfügung gestellt.

Inhaltsverzeichnis

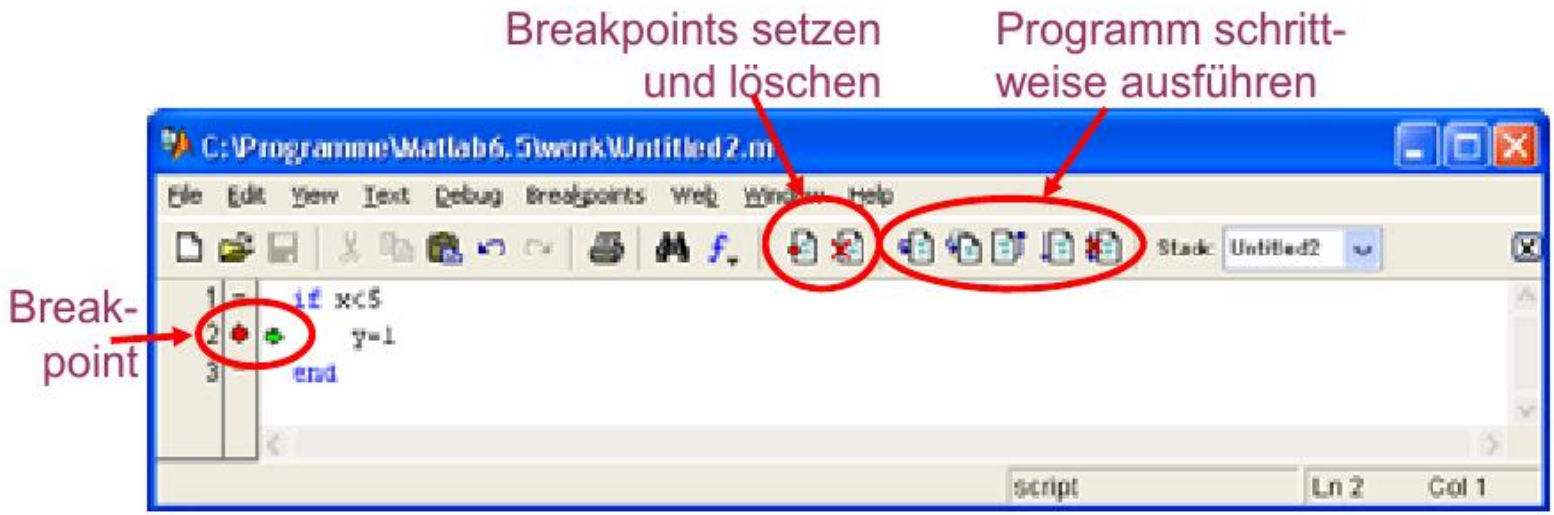
Inhaltsverzeichnis II

Fehlerraten in einem Programm

- **Syntaktischer Fehler**: Ein Befehl ist falsch geschrieben (z.B. durch einen Tippfehler).
  Wird von MATLAB sofort erkannt (Fehlermeldung).
  - `x=5/*y;`
    `Expected a variable, function, or constant, found "*".`
  - Fehler in Skripten werden mit Programmzeile und –spalte gemeldet
- **Semantischer Fehler**:
  Das Programm läuft, tut aber nicht was es soll.
  - Schwieriger zu finden, da nicht automatisch erkennbar.
  - Fehlersuche erfordert systematisches Testen eines Programms.

Debugger

- Ein **Bug** ist ein semantischer Fehler in einem Programm. Das Suchen und Entfernen von Bugs heißt **Debugging**.
- Ein **Debugger**, d.h. ein Werkzeug zum Debugging gehört zu jeder Programmierumgebung.
- In MATLAB findet man die Debugger-Funktionalität im Text-Editor unter „Debug“, „Breakpoints“ und der Menüleiste.

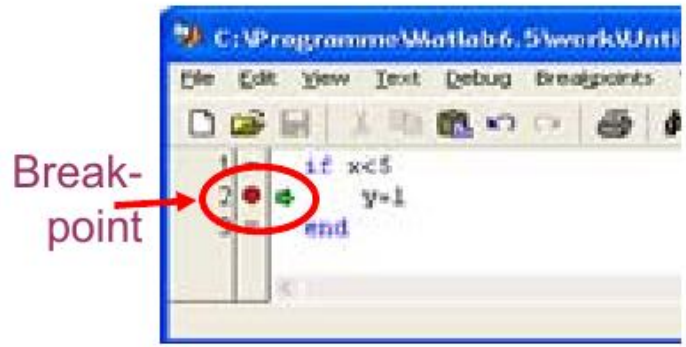


Schrittweises Ausführen

- Der Debugger kann ein Programm schrittweise ausführen:
  - Step : führt den nächsten Befehl aus
  - Step In : wird später erklärt
  - Save &Run: führt das ganze Programm aus
  - Exit Debug: Debugging beenden
- Nach jedem Schritt kann man die Werte aller Variablen inspizieren (Watch-Funktion):
  - Maus auf dem Variablennamen positionieren. Kurz abwarten.
  - Funktioniert nur, wenn die Variablen an dieser Stelle bereits definiert sind.

Breakpoints

- Breakpoints dienen dem Eingrenzen von Fehlern
- Das Programm wird angehalten, wenn ein Breakpoint erreicht wird.
- Danach kann mit Einzelschrittausführung fortgefahren werden.
- Ein Breakpoint kann auf jeder Programmzeile gesetzt werden.
  - Set Breakpoint:
    Setzt den BP auf die aktuelle Zeile.
    Zweimal Set entfernt den BP wieder.
  - Clear all Breakpoints:
    Entfernt alle BPs
    aus dem Programm



Beispiel für ein Debugging (1)

- Das Programm soll die Zahlen von 1 bis 5 addieren:
  - ```
    1: i=1;
    2: Summe=0;
    3: while i<5
    4:     i=i+1;
    5:     Summe=Summe+i;
    6: end;
    ```
- Methode: Schrittweise ausführen und Inspizieren der wichtigen Variablen
  - ```
    1: i=1;
    2: Summe=0;
    3: while i<5
    4:     i=i+1;
    5:     Summe=Summe+i;
    6: end;
    ```

Speicherauszug

i	1
Summe	0

Beispiel für ein Debugging (2)

- ```
  1: i=1;
  2: Summe=0;
  3: while i<5
  4:     i=i+1;
  5:     Summe=Summe+i;
  6: end;
  ```

Speicherauszug

i	2
Summe	2

1. Fehler: Zeile 4+5 vertauschen

- Programmkorrektur und Neustart:

```
1: i=1;
2: Summe=0;
3: while i<5
4:     Summe=Summe+i;
5:     i=i+1;
6: end;
```

Speicherauszug

i	2
Summe	1
```

Beispiel für ein Debugging (3)

- Einige Durchläufe später …

```
1: i=1;
2: Summe=0;
3: while i<5
4:     Summe=Summe+i;
5:     i=i+1;
6: end;
```

Speicherauszug

i	5
Summe	10

-

```
1: i=1;
2: Summe=0;
3: while i<5
4:     Summe=Summe+i;
5:     i=i+1;
6: end;
```

Speicherauszug

i	5
Summe	10

2. Fehler:
< durch
<= ersetzen

- Empfehlung: Programme immer erst mit Papier und Bleistift testen !

Systematische Fehlersuche (Testen)

- Kommerzieller Code enthält statistisch etwa 1 Bug pro 50 Zeilen
- Der systematische Test kritischer Codeteile (z.B. sicherheits-relevante Maschinensteuerungen) kostet extrem viel Zeit: Codierung : Test etwa 1 : 10
- Häufige Fehler bei Verzweigungen:
  - Ein Ast der Verzweigung liefert das falsche Resultat.
  - Es sind nicht alle Fälle vorgesehen.
  - Der falsche Ast wird betreten, weil mehrere Fälle zutreffen.
- Häufige Fehler bei Schleifen:
  - Der erreichte Endwert ist nicht gewollt.
  - Der gewollte Endwert wird nicht erreicht.
  - Wenn nichts zu tun ist darf die Schleife kein einziges Mal durchlaufen werden.
  - Endlosschleife durch unveränderte oder falsch veränderte Zählvariablen.