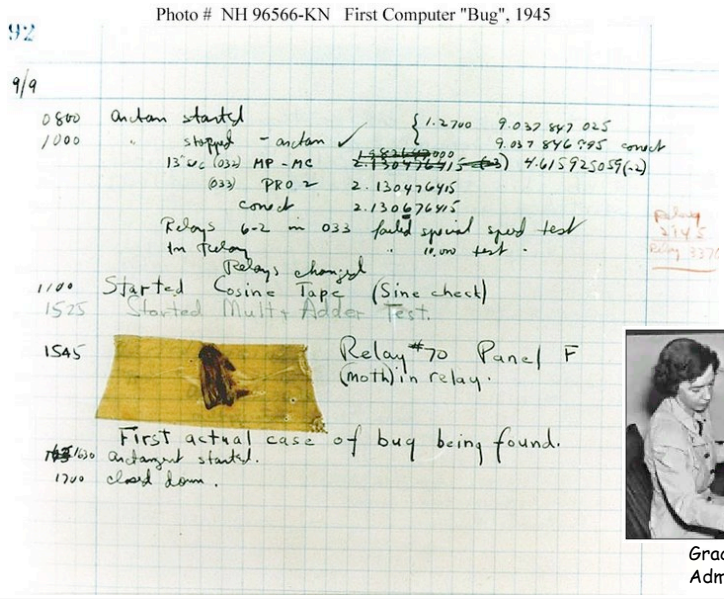
Informatik I
Debugging

G. Zachmann
Clausthal University, Germany
zach@in.tu-clausthal.de

Der erste Computer-Bug

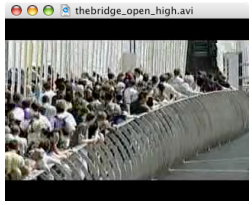


Grace Hopper
Admiral, US Navy

Folgen von Programmfehlern

- 1962 führte ein fehlender Bindestrich in einem Fortran-Programm zum Verlust der Venus-Sonde Mariner 1, welche über 80 Millionen US-Dollar gekostet hatte.
- Zwischen 1985 und 1987 gab es mehrere Unfälle mit dem medizinischen Bestrahlungsgerät Therac-25. Infolge einer Überdosis, die durch fehlerhafte Programmierung und fehlende Sicherungsmaßnahmen verursacht wurde, mussten Organe entfernt werden, ein Patient verstarb drei Wochen nach der Bestrahlung.
- 1999: NASA-Sonde *Climate Orbiter* verpasst den Landeanflug auf den Mars, weil die Programmierer das falsche Maßsystem verwendeten – Yard statt Meter. Die NASA verlor dadurch die mehrere hundert Millionen Dollar teure Sonde.

- 1996: Ariane-5 explodierte
  - Grund: Konvertierung einer zu großen Zahl von 64-bit float nach 16-bit int
- 1994: "Pentium-Bug"
  - Inkorrekte Division in manchen Fällen wegen unvollständiger Lookup-Tabellen
- Orkan "Lothar": keine Vorwarnung durch Wetterdienst
  - Grund: ignorierter Ausreiser
- Millenium-Bridge in London:
  - Falsche Anhnahmen über Fußgänger-Verhalten → fehlerhafte Simulation



Programmfehler (oder Bugs)

- Ursache:
  - Programmierfehler
  - Fehler in der Laufzeitumgebung
  - Unvollständigkeit, Mehrdeutigkeiten, Fehler in der Spezifikation
- Eine Regel besagt: Ab einer bestimmten Größe enthalten Computerprogramme auch immer Programmfehler
- Arten von Programmfehlern:
  - Syntaxfehler
  - Laufzeitfehler
  - Designfehler

Syntaxfehler

- Syntaxfehler sind Verstöße gegen grammatische Regeln der Programmiersprache

```
N = 100
for i in range( 0, N )
    while (N % i) == 0:
        print i
        N = N / i
```

Doppelpunkt vergessen

- Solche Fehlern werden beim Versuch des Programmstarts von Python erkannt und gemeldet.

```
Eingabeaufforderung
C:\Programmieren\Python\Python2.4.2>python debugging.py
  File "debugging.py", line 2
    for i in range( 0, N )
                          ^
SyntaxError: invalid syntax

C:\Programmieren\Python\Python2.4.2>_
```

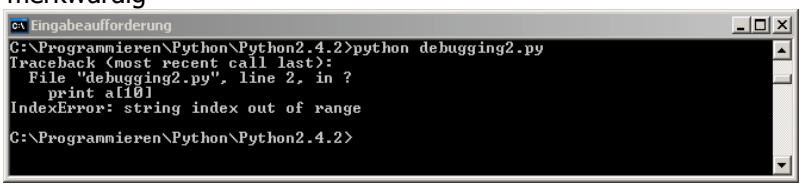
Laufzeitfehler (Semantische Fehler)

- Programme mit Syntaxfehlern lassen sich nicht starten.
- Semantische Fehler sind dagegen Fehler, die die Bedeutung des Programmcodes betreffen

```
a = "String"
print a[10]
```

a besteht nur aus 6 Zeichen

- Solche Fehler können das Programm zum Absturz bringen, das Programm friert ein (Endlosschleifen) oder verhält sich merkwürdig



Designfehler

- Designfehler sind Fehler im Grundkonzept einer Software
- Fehlerarten:
  - Fehler in der Anforderungsdefinition / Spezifikation
    - Das Programm tut nicht das, was der Kunde wollte
    - beruhen meist auf mangelnder Kenntnis des Fachgebietes, für das die Software entwickelt wird
  - Fehler im Softwaredesign
    - Das Programm läßt sich später nicht mehr verstehen / warten / verbessern
    - beruhen oft auf mangelnder Erfahrung des Softwareentwicklers

Vermeidung von Fehlern

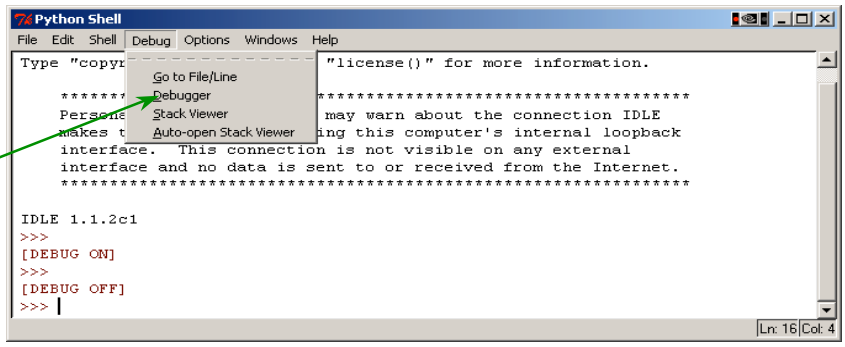
- Je früher in einem Entwicklungsprozeß ein Fehler auftritt und je später er entdeckt wird, desto aufwendiger ist es, ihn zu beheben.
- Designfehler lassen sich oft durch gute Planung und Wahl eines geeigneten Softwaredesigns weitestgehend vermeiden
- Syntaktische Fehler erkennt meist der Compiler oder der Interpreter
- Beim Auffinden semantischer Fehler helfen (für bestimmte Fälle / Klassen von Fehlern) spezielle Programme

Debugger

- Nicht immer sind semantische Fehler so einfach zu finden wie in dem Beispiel oben
- Deswegen hilft ein spezielles Programm, der sogenannte Debugger, dem Programmierer bei der Fehlersuche
- Ein Debugger ermöglicht die Ablaufverfolgung des Programms in einzelnen Schritten
- Mit Python wird ein Debugger mitgeliefert

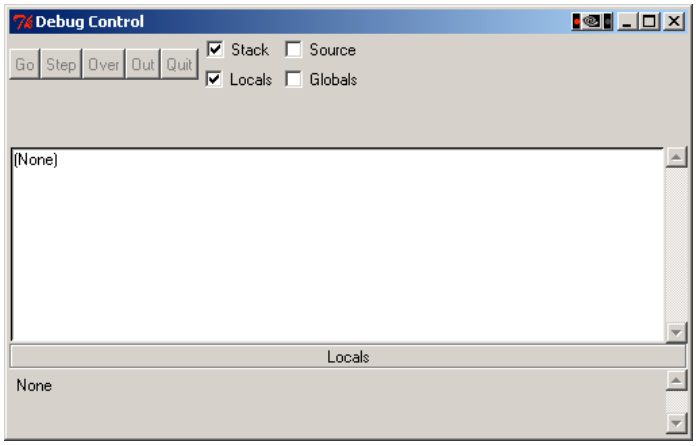
Python-Debugger

- Die Python-IDE Idle starten
- Das Menu „Debugger" wählen
- Auf Menüpunkt „Debugger" drücken



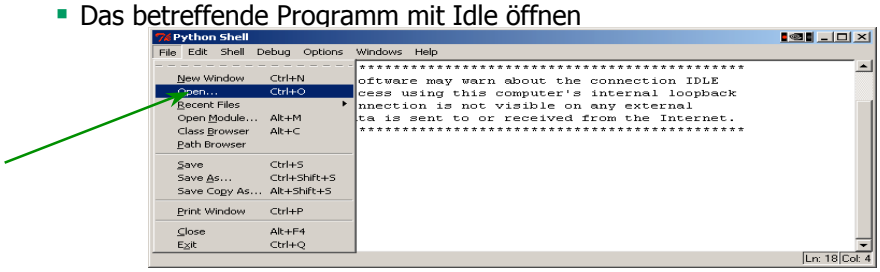
Python-Debugger

- Das Debugger-Fenster wird geöffnet

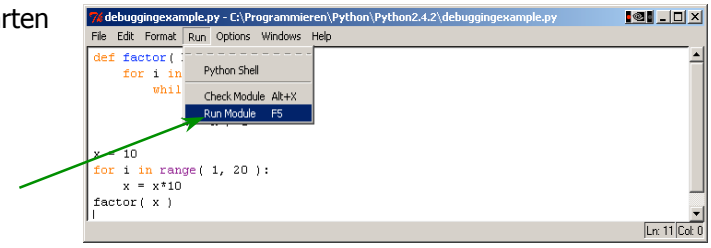


Python-Debugger

- Das betreffende Programm mit Idle öffnen

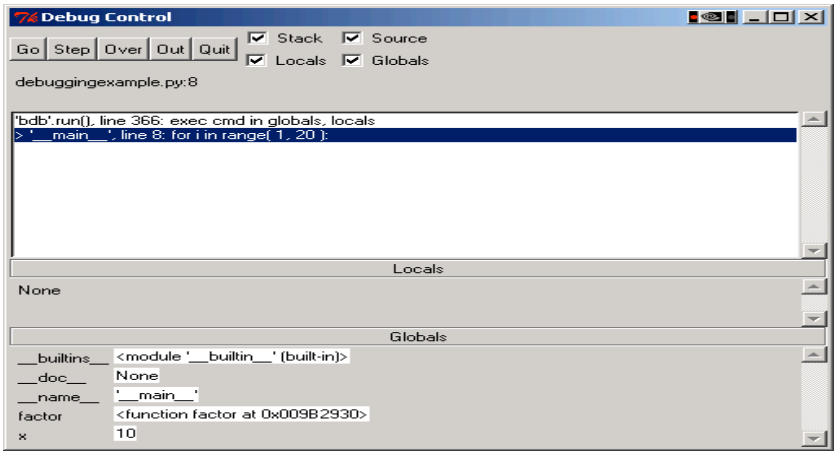


- Und starten



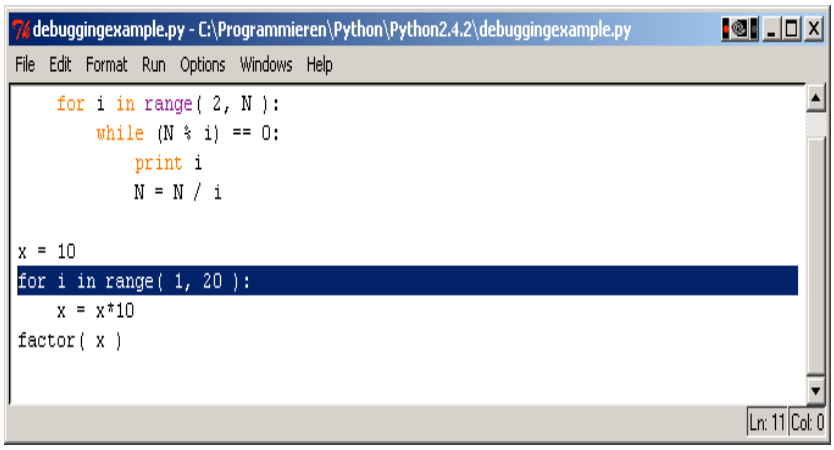
Python-Debugger

- Im Debugger-Fenster sieht man den Stack, sowie die Werte der lokalen und globalen Variablen



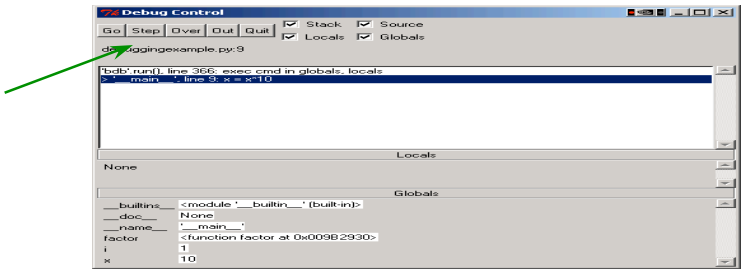
Python-Debugger

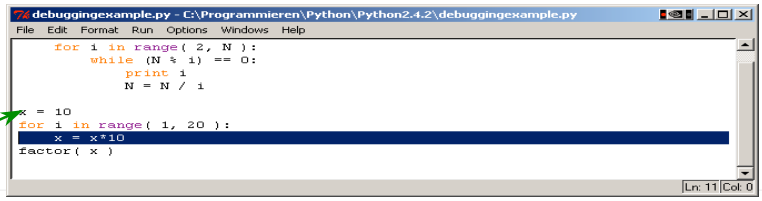
- Im Sourcecode-Fenster wird die aktuelle Position, an der der Debugger das Programm gerade hält, angezeigt



Python-Debugger

- Mit dem Step-Button kann man das Programm schrittweise weiter ausführen





Python-Debugger

- Wenn man nur an bestimmten Stellen des Source interessiert ist kann man per Rechtsklick im Sourcecode-Fenster einen Breakpoint setzen und mittles Go-Taste im Debugger-Fenster bis zu dieser Stelle laufen lassen