

Programmierkurs Python I

Michaela Regneri
2009-10-30

(Folien basieren auf dem gemeinsamen Kurs mit Stefan Thater,
der für diese Vorlesung auf Alexander Kollers Java-Kurs basierte)



Organisatorisches

- Wöchentliche Übungen (wichtig!) in Teams von max. drei Studenten. Jeder Student muss die Übung selbst kommentieren und erklären können
- Prüfungsvoraussetzung: 50% der Übungspunkte
- Größeres Programmierprojekt nach etwa der Hälfte des Kurses (50% der Endnote)
- Klausur am Ende des Semesters (50% der Endnote)

Was heißt „Programmieren“?

- Der Programmierer will ein **Problem** auf systematische Weise lösen.
- Ein **Algorithmus** ist eine abstrakte, detaillierte Berechnungsvorschrift, die das Problem löst.
- Ein **Programm** ist eine Darstellung des Algorithmus in einer konkreten Programmiersprache.
- Ein Programm kann mit verschiedenen **Eingaben** ausgeführt werden.

3

Einige Probleme

- Berechnung von arithmetischen Funktionen
- Was ist der kürzeste Weg in einem Graphen?
- Computerlinguistische Probleme: morphologische Analyse, Taggen, grammatische Analyse (Parsen), ...
- Alltägliches: Wie komme ich vom Bett zur Uni?
 - Teilproblem: Anziehen

4

Algorithmen ...

- „Kochrezepte“, die Schritt für Schritt beschreiben, wie das Problem gelöst wird
- Sollten für alle Eingaben, die das Problem vorsieht funktionieren
- Müssen in endlich vielen Schritten fertig werden
- Granularität der Einzelschritte hängt davon ab, was als offensichtliche Schritte akzeptiert wird

5

Ein Algorithmus für Anziehen

- Gehe zum Kleiderschrank.
- Nimm eine frische Unterhose.
- Stecke die Beine durch die Löcher.
- Nimm ein frisches T-Shirt.
- Stecke Kopf und Arme durch die passenden Löcher.
- ...
- Binde den rechten Schuh.

6

Algorithmus für „größte Zahl“

- gegeben eine Liste `list` von n natürlichen Zahlen; gesucht ist die größte Zahl in `list`
- möglicher Algorithmus:
 - merke dir die erste Zahl in `list` als aktuelles Maximum
 - betrachte nacheinander die 2-te bis letzte Zahl in `list`
 - vergleiche die betrachtete Zahl mit dem aktuellen Maximum
 - ist die Zahl größer, ändere das Maximum auf den Wert der Zahl
 - das gespeicherte Maximum ist nach Ansehen aller Zahlen in `list` die größte Zahl aus `list`

7

Programme ...

- sind konkrete Implementierungen eines Algorithmus in einer Programmiersprache
- verwenden Konstruktionen der Programmiersprache, um intuitive Begriffe präzise zu machen
 - Schleifen, Bedingungen, Variablen, ...
- Einzelne Schritte hängen von Programmiersprache und verfügbaren Funktionen ab.

8

„Größte Zahl“ in Python

- merke dir die erste Zahl in `list` als aktuelle größte Zahl
- gehe der Reihe nach durch die 2-te bis letzte Zahl in `list`
- ist die aktuelle Zahl größer als die bisherige größte (`max`), dann merke dir die aktuelle als die neue größte Zahl
- am Ende ist der gespeicherte Wert von `max` die größte Zahl in `list`

```
max = list[0]
i = 1
while i < len(list):
    if liste[i] > max:
        max = liste[i]
    i = i + 1
```

Vorsicht: Wir haben den Spezialfall missachtet, dass die Liste auch leer sein kann.

9

Effiziente Algorithmen

- Manche Algorithmen lösen ein Problem effizienter als andere (mit weniger Zeit- oder Speicherverbrauch)
- Beispiel: Finde die zwei größten Zahlen in `list`
 - Algorithmus #1: Berechne zunächst die größte Zahl; dann lösche sie aus `L` und berechne ein weiteres mal die größte Zahl
 - Algorithmus #2: Verwende zwei Variablen, um die aktuell beiden größten Zahlen zu speichern

10

Programmiersprachen

- Es gibt eine sehr breite Auswahl an verschiedenen Programmiersprachen
- CPU versteht sehr eingeschränkten Befehlssatz, der für Menschen unbequem ist
- Programmiersprachen verbergen Komplexität und stellen abstraktere Konstrukte zur Verfügung
 - Verfolgen verschiedene Ziele
 - unterstützen verschiedene Programmierstile
 - haben verschiedene Vor- und Nachteile.

11

Programmierparadigma

- **Imperativ:** Programme sind streng geordnete Sequenzen von Anweisungen. (Algol, Fortran, Pascal, Java, C++)
- **Deklarativ:** Programm-Code spezifiziert das Problem so, dass sich die eigentliche Berechnung der Lösung ergibt
 - **Funktional:** Programme sind Funktionen (vergleichbar mit math. Funktionen), die aus einer Eingabe eine Rückgabe berechnen und ggf. aus einfacheren Funktionen zusammengesetzt sind (SML, Lisp, Haskell)
 - **Logisch:** das Programm besteht aus einer Menge von Axiomen; ein Programmaufruf versucht, die „Eingabe“ zu beweisen (Prolog)

12

Prozedural vs. Objektorientiert

- **Prozedural:** Die wesentliche Gliederungseinheit des Programms ist die Prozedur, d.h. Gruppe von Anweisungen (C, SML)
- **Objektorientiert:** Die wesentliche Gliederungseinheit ist die Klasse, d.h. Repräsentation eines Objekts mit Sammlung von Methoden (Java, C++, Smalltalk)

13

Python ...

- Python ist eine objektorientierte Programmiersprache, d.h. objektorientierte Programmierung wird konsequent unterstützt.
- Allerdings wird man nicht gezwungen, objektorientiert zu programmieren: Man kann auch prozedural und (teilweise) funktional programmieren.

14

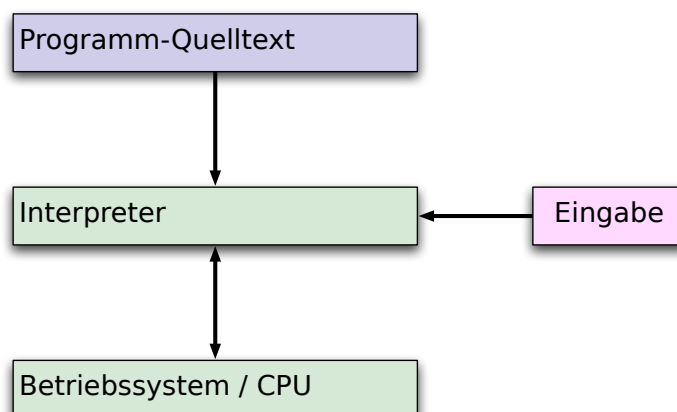
Interpretiert vs. kompiliert

- Programmiersprachen sind dafür da, dass Menschen Algorithmen bequem aufschreiben können.
- Programm-**Quelltexte** können nicht direkt vom Computer ausgeführt werden.
- Zu einer Programmiersprache gehört deshalb Software, die Programme für Computer ausführbar macht.
 - Interpreter oder Compiler

15

Interpretation

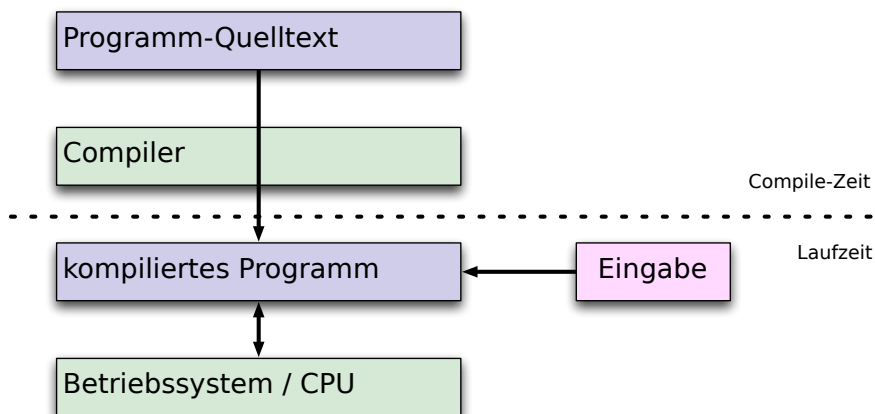
- Interpreter sind Programme, die den Quelltext Anweisung für Anweisung ausführen. Beliebte z.B. für Skriptsprachen wie Perl.



16

Kompilation

- Compiler sind Programme, die den Quelltext in ein direkt ausführbares Maschinenprogramm übersetzen. Beliebte z.B. für C, C++ usw.



17

Interpretation vs. Kompilation

- Interpretation ist (teilweise deutlich) langsamer, da Interpreter selbst Rechenzeit verbraucht.
- Ein Compiler wendet einmal Zeit bei der Kompilation auf. Danach wird das Programm direkt („nativ“) von der CPU sehr effizient ausgeführt.
- Ein kompiliertes Programm läuft nur auf einer einzigen Plattform (Betriebssystem + CPU). Der Quelltext muss daher auf neuen Plattformen neu kompiliert werden und ggf. angepasst werden (Portierung). Das ist oft problematisch.

18

Python-Ausführung

- Python-Programme werden zunächst in Byte-Code kompiliert, der dann von einer virtuellen Maschine interpretiert wird
- die Art der Virtuellen Maschine hängt von der Python-Implementierung ab (Standard: C, gibt auch Java, .NET...)
- Python kann auch als interpretierte Sprache benutzt werden

19

Python - allgemein

- Der Name kommt von *Monty Python*
- Python basierte ursprünglich auf der „Lehrprogrammiersprache“ ABC
- ABC war für ambitionierte Anwender ohne Programmierkenntnisse entworfen
- ABC wurde damit auch als übersichtliche, einfache Sprache entworfen
- Python war ursprünglich ein ABC-Ausschnitt mit ein paar praktischen (aber komplexeren) zusätzlichen Konstrukten
- Besonderer Fokus: Einfache Datei-Handhabung

20

Vorzüge von Python

- Einfache Syntax (⇒ gut lesbare Programme)
- Sehr flexibel, wenige “Verbote” (viel Konvention)
- Datei-Handhabung
- volle Unicode-Unterstützung
- beliebig große (!) Ganzzahlen
- bequeme Umsetzung regulärer Ausdrücke

21

Unser erstes Programm: Hello, World!

- Unser erstes Python-Programm (Datei: hello.py):

```
print("Hello, World!")
```
- Ausführen des Programms von der Kommandozeile:

```
> python hello.py
```

22

Python Interaktiv

```
> python3
```

```
Python 3.0.1+ (r301:69556, Apr 15 2009, 15:59:22)
```

```
[GCC 4.3.3] on linux2
```

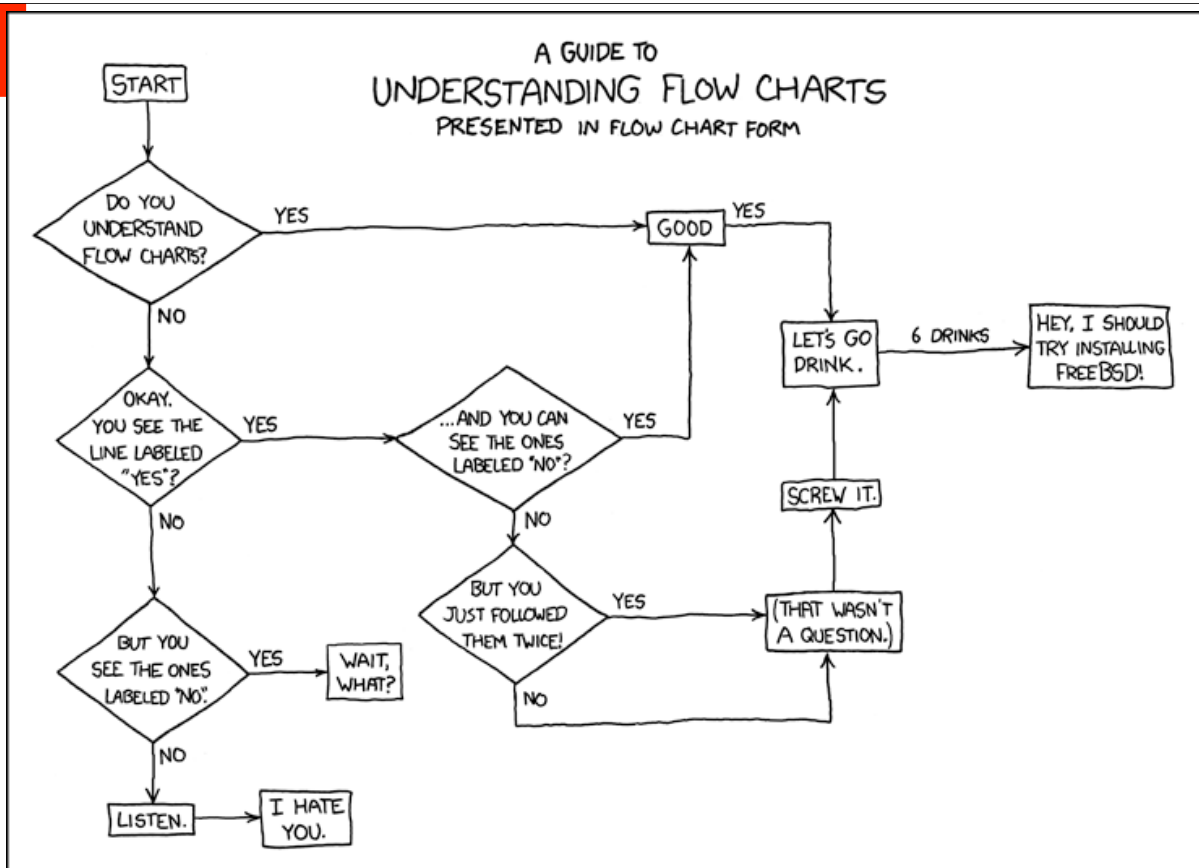
```
Type "help", "copyright", "credits" or "license" for  
more information.
```

```
>>> print("Hello World")
```

```
Hello World
```

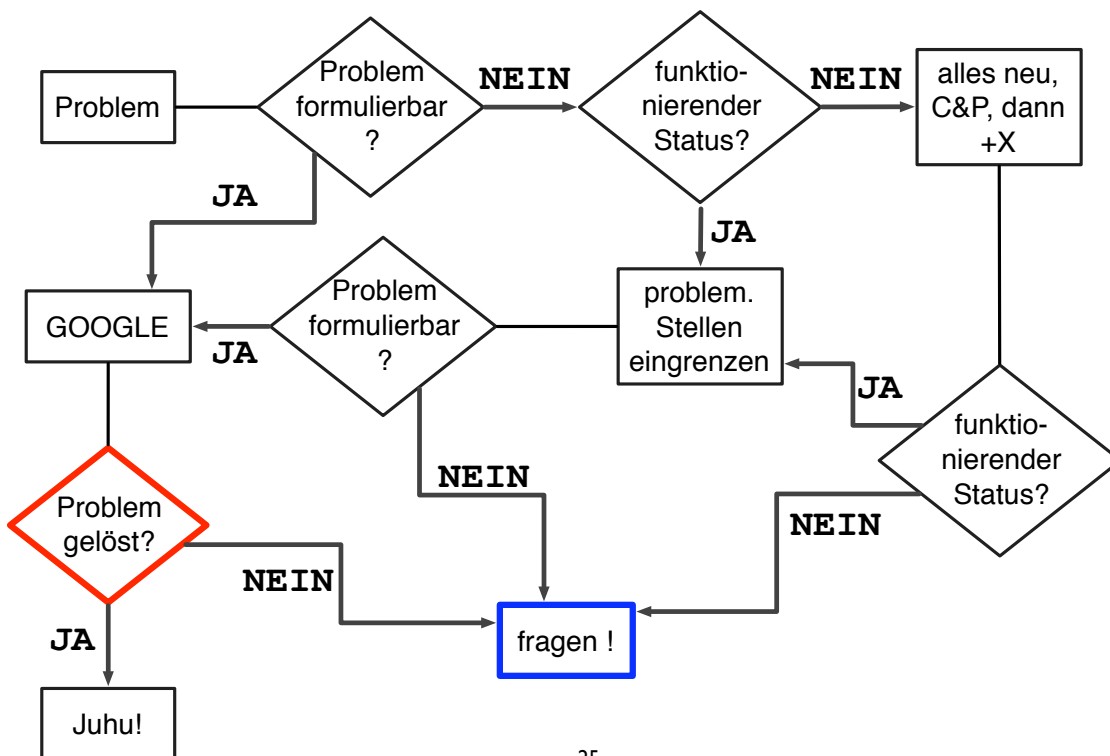
```
>>>
```

23



24 as taught by xkcd.com

Lösungswege



25

Literatur

- Viele Tutorials & komplette Doku unter www.python.org: (auf Deutsch z.,B. <http://python.net/~gherman/publications/tut-de/online/tut/>)
- Das Python-Tutorial: <http://tutorial.pocoo.org/>
- Video-Tutorial (ungetestet): <http://showmedo.com/videotutorials/series?name=HRCwU3mb5>
- Für Leute, die schon Programmieren können: Dive Into Python (<http://www.diveintopython.org/>)
- Viele Bücher; falls eins hinzugezogen wird, bitte auf Python 3 achten (SULB leider unterversorgt hier)

26