

HOW DOES COMPUTER
PROGRAMMING WORK?

MAGIC.



VORLESUNG PARADIGMEN UND KONZEPTE VON PROGRAMMIERSPRACHEN



ibas

departement mathematik & informatik

informatik.unibas.ch/lehre/fs20/prog#thorsten.moeller

Team

Dozent: Dr. Thorsten Möller thorsten.moeller@unibas.ch
Chief Technical Officer (CTO)
SBI AG, Reinach, www.sbi.ch

Tutoren: Salome Müller salo.mueller@stud.unibas.ch
Maarten Schenk maarten.schenk@unibas.ch

Koordinaten

- **Vorlesung** 2.5 SWS
 - Freitag 08.15 – 10.{00,30} Uhr Spiegelgasse 5,
Seminarraum 05.002

- **Übung** 1 SWS
 - Freitag 11.15 – 12.00 Uhr Spiegelgasse 5,
Seminarraum 05.002
 - Freitag 12.15 – 13.00 Uhr Spiegelgasse 5,
Seminarraum 05.002
 - Start Freitag 16.03.2019 (KW12)

- **Anmeldung/Belegen – Frist 16.03.2020**
 - Vorlesung: services.unibas.ch
 - Übung: courses.cs.unibas.ch

Vorlesungstermine

#	Datum	Dauer
1	Freitag 21.02.2020	08.15-10.00 Uhr
2	Freitag 28.02.2020	08.15-10.30 Uhr
3	Freitag 13.03.2020	08.15-10.30 Uhr
4	Freitag 20.03.2020	08.15-10.30 Uhr
5	Freitag 27.03.2020	08.15-10.30 Uhr
6	Freitag 03.04.2020	08.15-10.30 Uhr
7	Freitag 17.04.2020	08.15-10.30 Uhr
8	Freitag 24.04.2020	08.15-10.30 Uhr
9	Freitag 08.05.2020	08.15-10.30 Uhr
10	Freitag 15.05.2020	08.15-10.30 Uhr
11	Freitag 29.05.2020	08.15-10.00 Uhr

Vorlesungsfrei

Freitag 06.03.2020 (Fasnacht)

Freitag 10.04.2020 (Ostern)

Freitag 01.05.2020

Freitag 22.05.2020 (Auffahrt)

2x 45min + 15min Pause

2x 60min + 15min Pause

Übungsablauf

- 5 Übungen – im Rhythmus von drei, zwei, bzw. einer Woche.
 - Kleinere Programmieraufgaben
 - Auf Englisch formuliert
 - Rechner zum Bearbeiten der Übungen in Räumen U1.001 und U1075
 - Tutorial zum Aufsetzen einer eigenen Umgebung am **13.03.2020**
- Bearbeitung in Gruppen von max. 2 Personen
- Lösungen via courses.cs.unibas.ch abgeben (Abgabetermin auf jeweiligen Übungsblatt) und Vorzeigen der Lösung während Übungstermin; ansonsten keine Anwesenheitspflicht.
- **Optionale** Zusatzaufgabe(n) pro Übung

Planung Übungstermine

#	Datum	Bemerkung	#	Datum	Bemerkung
0	13.03.20	Tutorial			
1	16.03.20	Ausgabe	4	04.05.20*	Ausgabe
	29.03.20	Abgabe		17.05.20	Abgabe
	27.03., 03.04.	Demonstration		-	ohne Demonstration
2	30.03.20	Ausgabe	5	11.05.20* **	Ausgabe
	19.04.20	Abgabe		24.05.20	Abgabe
	17.04., 24.04.	Demonstration		ohne Demonstration	
3	20.04.20	Ausgabe			
	03.05.20	Abgabe			
	24.04., 08.05.	Demonstration			

- * Geringerer Umfang als 1.-3. Übung und deshalb auch geringere max. Punkteanzahl.
- ** Um eine Woche Überlappend zur 4. Übung!

Wie bekommt man die Kreditpunkte?

- 6 ECTS
- Übungen durch Punkte bewertet
 - $\frac{2}{3}$ der Punkte – Summe über alle Übungen – sind Voraussetzung für Zulassung zur Klausur.
 - Bei erneutem Belegen der Vorlesung neu zu erwerben.
- Schriftliche Klausur
 - Termin: Freitag, 05.06.2020
09:15-10:45 (90min), Kollegienhaus HS118

Vorlesungsmaterialien

- Alle Folien werden im Netz als Skript bereitgestellt.
- **Beachte!**
 - Folien ändern sich von Jahr zu Jahr (wenn auch nur minimal), denn eine Vorlesung ist etwas lebendiges; etwas, was niemals ein Endstadium erreicht.
 - Es gibt prüfungsrelevante Themen, die auf keiner Folie enthalten sind.

Literatur

Peter A. Henning, Holger Vogelsang

Taschenbuch Programmiersprachen

Hanser, 2. Auflage 2007, ISBN 978-3-446-40744-2



Oder der „grosse Bruder“ des Taschenbuchs

Handbuch Programmiersprachen

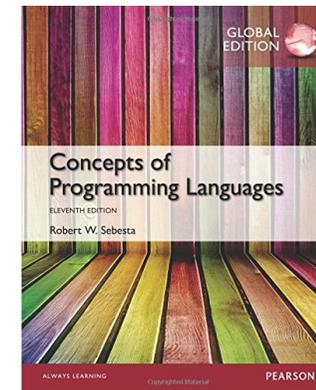
Hanser, 1. Auflage 2006, ISBN 978-3-446-40558-5



Robert W. Sebesta

Concepts of Programming Languages

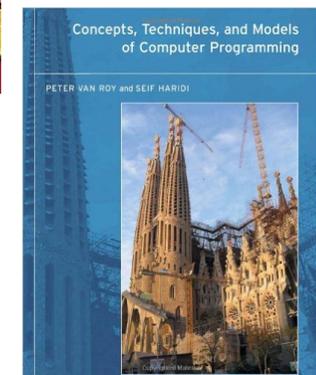
Prentice Hall, 11th ed. 2016, ISBN 978-1292100555



Peter Van Roy, Seif Haridi

Concepts, Techniques, and Models of Computer Programming

MIT Press; 1st edition 2004, ISBN 978-0262220699



Organisation – Fragen?



Vorlesungsinhalt

- Einführung
- Systemnahe Programmierung (Ass., C++)
- Funktionale Programmierung (Haskell)
- Logikprogrammierung (Prolog)
- Dynamische Sprachen
- Domänenspezifische Sprachen

(2x2h)

(6x2h)

(4x2h)

(1x2h)

(1x2h)

(1x2h)

Schätzwerte

Benötigte Vorkenntnisse

- Themen die als bekannt angenommen werden:
 - **Vorlesung Algorithmen und Datenstrukturen**
 - Rekursion
 - Datenstrukturen: Listen, Assoziative Listen, Mengen, Arrays
 - **Vorlesung Grundlagen der Programmierung**
 - Variable im Sinne der imperativen Programmierung
 - Klasse versus Objekt versus Instanz
 - Objektgleichheit versus Objektidentität
 - „einfache“ Vererbung, Polymorphie
 - Prozeduren bzw. Unterprogramme
 - Kontrollstrukturen
 - Kopf- bzw. fussabweisende Schleifen
 - Bedingte Verzweigung

Abgrenzung

(i)

- Welche Themen sind nicht Inhalt dieser Vorlesung?
 - **Markupsprachen**: „Programmierung“ von Inhalt und Layout von (Bildschirm-)Seiten
 - Meist keine Möglichkeiten zur Problemlösung durch Berechnung (d.h. meist nicht Turing-vollständig)
 - **Visuelle (im Sinne von graphischer) Programmierung**
 - Realisierung klassischer Programmierkonzepte mit visuellen/graphischen (Hilfs-)Mitteln
 - **Datenbankprogrammierung**
 - Anfrage- und Datendefinitionssprachen in Vorlesung *Datenbanken*
 - **Parallele Programmierung**
 - Ausgiebig behandelt in Vorlesung: *High Performance Computing*

Abgrenzung

(ii)

- Welche Themen sind nicht Inhalt dieser Vorlesung?
 - „Programming in the large“
 - (verteilte) Programmierung zur Abbildung von (verteilten) Workflows auf Basis von (Micro-)Services
 - Implementierung von Programmiersprachen
 - Compiler- bzw. Interpreterbau
 - Automatisierte Verifikation, Analyse
 - (Programmier-)Sprachentheorie
 - Formale Modelle zur Beschreibung von Syntax und Semantik
 - Klassifizierung von Sprachen nach Komplexität, Ausdrucksstärke, formale Verifizierbarkeit ...
 - Probabilistische Programmiersprachen
 - Anwendungsgebiete in Statistik und maschinellem Lernen

Folienkonventionen

23

Zeiger und ihre „dunkle Seite“

```
int a = 42;
int* p;
p = &a;

std::cout << "an der Adresse " << p
           << "steht (als integer) "
           << *p << std::endl;
```

p enthält eine Adresse. Der Speicherinhalt dort soll als Integer interpretiert werden

p enthält jetzt die Adresse, an der die Variable a im Speicher liegt.

20

... noch ein Zeigerarithmetik-Beispiel

```
sizeof(int) = 4
int* p = &a;          *p = 200;          *(p+1) = 300;
```

p (aa8200)	aa8192	p (aa8200)	aa8192	p (aa8200)	aa8192
h		h		h	

Oranger Folienkopf: Standardfolie

Grüner Folienkopf: Zusatzfolie zum Selbststudium. Wird in Vorlesung nur besprochen falls genügend Zeit zur Verfügung steht.

- Farben, die innerhalb der Folien benutzt werden, haben keine feste Bedeutung. Farben werden (in inkonsistenter Weise) nur zur Hervorhebung benutzt.