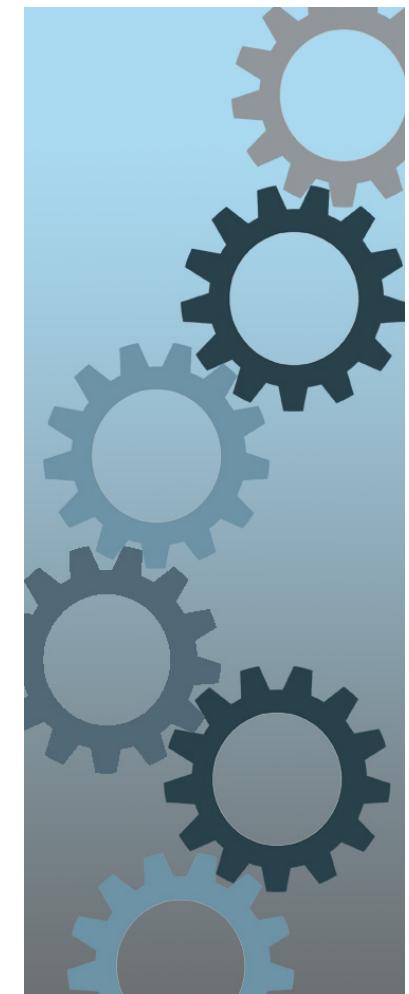


Programming Concepts & Paradigms

Team-Projekt: Aufgabenstellung

Prof. Dr. Ruedi Arnold

ruedi.arnold@hslu.ch



Semesterplan im Bild

SW 1	SW 2	SW 3	SW 4	SW 5	SW 6	SW 7
Einführung Paradigmen	Logische Programmierung I	Logische Programmierung II	Logische Programmierung III	Funktionale Programmierung I	Funktionale Programmierung II	Funktionale Programmierung III
Imperative vs. prozedurale Programmierung, Einführung: Paradigmen (Ruedi Arnold)	Einführung Prolog, Fakten, Regeln, Abfragen, Beweisstrategien, Backtracking, Operatoren	Optimierungen (Endrekursion, Aufbau, Anwendungen), Autoren	Schwache Negation, Finite Domänen, Sudoku, usw., Datenstrukturen (HTTP, JSON)	Einführung Scheme, Sprachelemente, Ausdrücke, funktionale Modellierung, strukturierte Daten und Datenabstraktion	Lambda-Kalkül, rekursive Datentypen, Rekursionsfunktionen, Funktionen höherer Ordnung	lokale Definitionen, lexikalisches Scoping, Spezialitäten und Anwendungen
	logische Programmierung: Prolog (Ruedi Arnold)			funktionale Programmierung: Scheme (Roger Diehl)		
	LF1, LF2, LF3					
SW 8	SW 9	SW 10	SW 11	SW 12	SW 13	SW 14
Java 8 & „functional Thinking“ I	Java 8 & „functional Thinking“ II	Compiler I & Kickoff Team-Projekt	Compiler II	Team-Projekt I	Team-Projekt II	Abschluss Team-Projekte
Lambdas, funktionale Interfaces, Methodenreferenzen, default und statische Interface-Methode, Streams	Lambdas & Streams im Einsatz, weitere Java „functional Thinking“, Bausteine funktionaler	Compiler, Interpreter, formale Sprache, Grammatik, BNF, virtuelle Maschinen, Parser-Generatoren	Lexer / Parser, Abstract Syntax Tree (AST), Anwendung eigener Compiler	Coaching, Besprechung Abschlusspräsentation	Coaching, Besprechung Abschlusspräsentation	Abschlusspräsentation inkl. Demos
Java 8 & "Functional Thinking" (Ruedi Arnold)		Compiler & Grammatiken (Roger Diehl)		individuelle Team-Projekte (Ruedi Arnold & Roger Diehl)		
HS 2018 V1.0		LF1, LF2, LF3		PCP: Team-Projekt - Aufgabenstellung		2

Weitere Vorträge ☺

- **30.11.: Compiler Bau**

- Roger Diehl

- **06.12.: Swift-Highlights**

- Ruedi Arnold

- **ggf. weiterere Gastvorträge ???**

- tbd.



Kein Prüfungsstoff

Es gibt nicht nur eine (Programmier-)Sprache...

→ Was kennt die Welt sonst noch ausser Java?

- Blick über den berühmten Tellerrand ☺
 - Bzw.: Warum es gut ist, andere Sprachen zu kennen

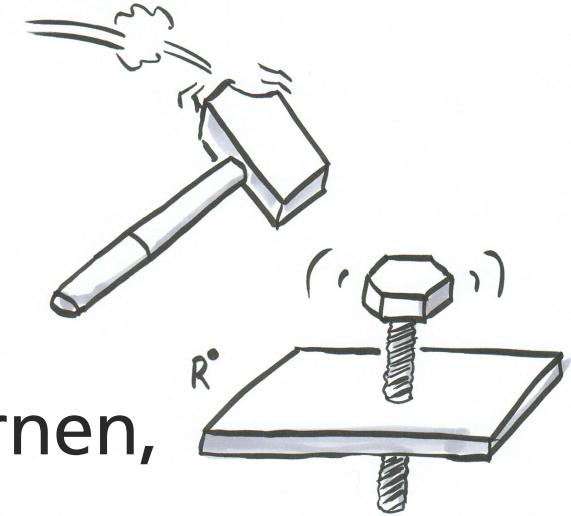


Grundidee vom Team-Projekt

- **Bisher in PCP:** Behandlung diverser Programmiersprachen, primär C, Java, Prolog & Scheme
 - Fokus auf Imperativ-objektorientierter, logischer & funktionaler Programmierung
- **Ab jetzt beschäftigen sie sich vertieft mit einer weiteren, individuellen Programmiersprache und präsentieren uns anhand von dieser weitere interessante Sprachkonzepte und –konstrukte**
 - Nach dem Motto: "Was gibt's in anderen Sprachen sonst noch, was Prolog/Scheme/Java so nicht kann?"
 - ...und wir alle lernen dabei hoffentlich viele spannende neue Dinge über x Programmiersprachen ☺

Ziele vom Team-Projekt

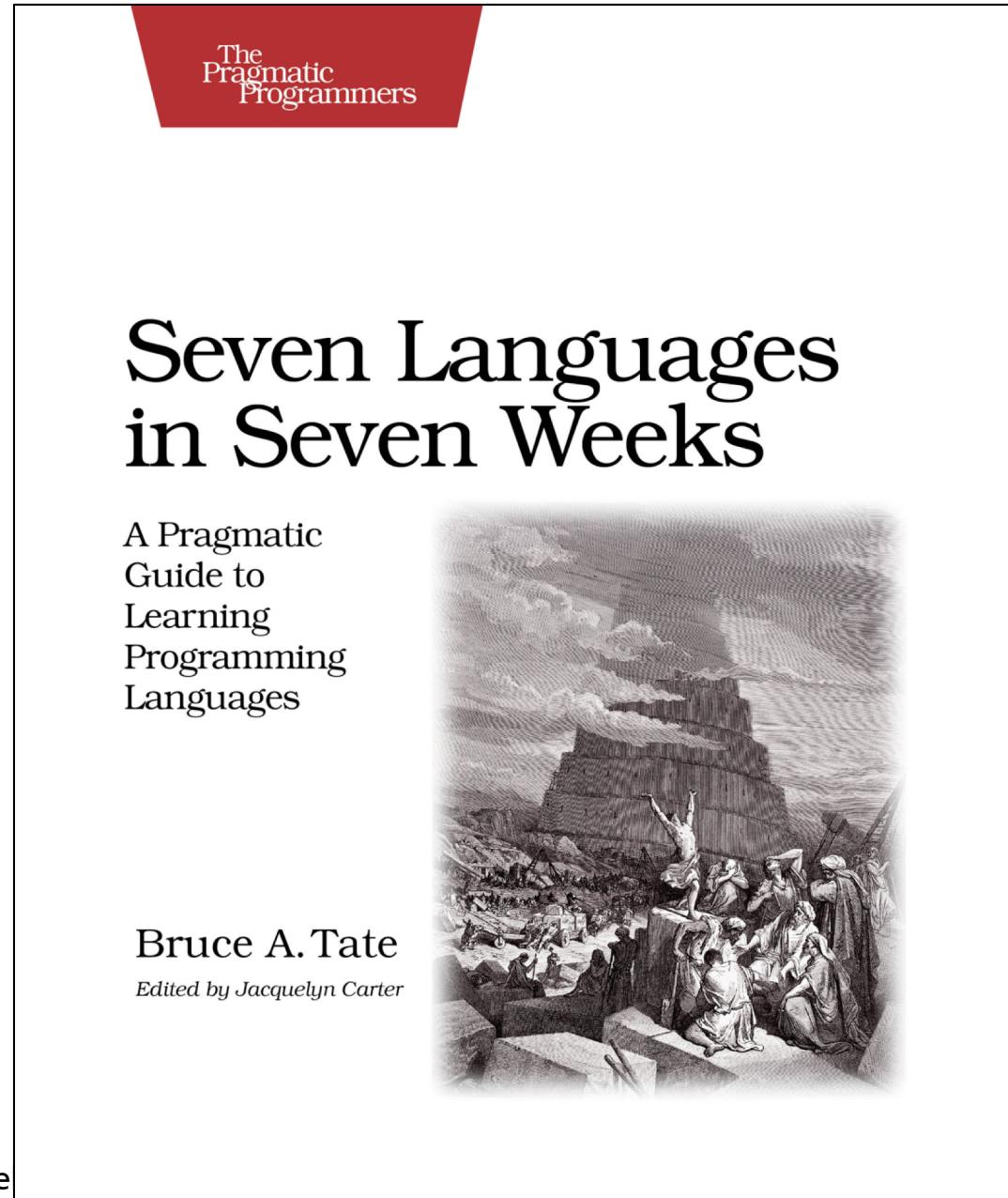
- Selbstständig im Zweier-Team eine neue Programmiersprache kennenlernen, analysieren und vorstellen
- Das bisher Gelernte anwenden
- Ihr Programmiersprachen-Wissen in selber gewählte Richtungen vertiefen und ausbauen
- Erfahrungen sammeln: viel über andere / neue Programmiersprache lernen, sich in diese einarbeiten, Demo-Code schreiben
- "Ihre Sprache" am Schluss kompakt aufbereiten und im Plenum präsentieren ☺



Buch-Tipp: Seven Languages in Seven Weeks

- Bietet kompakten, praktischen und pragmatischen Einstieg (oder eher Überblick) in folgende Sprachen:
Clojure, Haskell, Io, Prolog, Scala, Erlang, Ruby

→ Gute Weiterbildungslektüre für ProgrammierInnen 😎



Seven Languages in Seven Weeks

A Pragmatic
Guide to
Learning
Programming
Languages



Bruce A. Tate
Edited by Jacqueline Carter

...Vorgehen neue Sprache lernen?!

■ Pragmatische Fragen aus dem Buch :

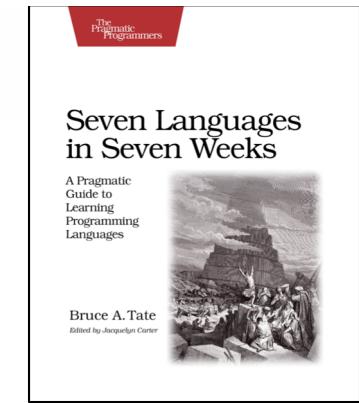
- Was ist das Typenmodell?
 - stark (z.B. Java) oder schwach (z.B. C), statisch (d.h. zur Kompilierzeit wie z.B. Java) oder dynamisch (d.h. zur Laufzeit wie z.B. Ruby), Gibt's Typinferenz (wie z.B. bei Swift)? Gibt's primitive Datentypen (wie z.B. bei Java) oder nur Referenztypen (wie z.B. bei Ruby)? Properties (C#)?
- Was ist das Haupt-Programmiermodell?
 - ...objektorientiert, funktional, prozedural? Oder etwas hybrides, d.h. ist es eine Multi-Paradigmen-Sprache? Welche Paradigmen?
- Wie laufen Programme ab?
 - Kompiliert? Interpretiert? Auf virtueller Maschine?
- Was sind wichtigsten Entscheidungskonstrukte?
 - if & while (imperativ), Pattern Matching (Erlang), Matching & Backtracking (Prolog)

Pragamatische Fragen, Teil II

- Was sind die wichtigsten Datenstrukturen?
 - Collections (Listen) sind wie integriert in die Sprache? Sind Listen zentrales Element (Scheme)? Array[]?
- Weitere Hauptmerkmale, welche diese Sprache einzigartig machen?
 - Spezielle Unterstützung für nebenläufige Programmierung (z.B. Aktoren in Erlang)? Einmalige Konstrukte wie Makros in Clojure oder Nachrichten-Interpretation in Io? Oder wie Constraint-Solving in Prolog? Wie werden Objekte erzeugt? Gibt's Klassen? Oder Objektorientierung durch Prototypen (wie JavaScript)

→ Gute Einstiegsfragen, um "Ihre" Sprache kennen zu lernen

→ **Aufgabe: Fragen durchgehen und so "Ihre" Sprache "entdecken" 😊**



Seven Languages
in Seven Weeks

A Pragmatic
Guide to
Learning
Programming
Languages



Bruce A. Tate
Edited by Jacqueline Carter

...Filmfigur zur Sprache? ;-)

- Was mir an diesem Buch weiter gefällt:
Jeder Programmiersprache wird eine Filmfigur zugewiesen, um die Charakteristik dieser Sprache kompakt mit einer Analogie zu vermitteln, resp. auf den Punkt zu bringen. z.B.:
 - Prolog: Raymond aus "Rain Man" (Kennt Telefonbuch auswendig, nachdem er's durchgelesen hat und fragt sich aber nicht, ob er das tun soll. Ein bisschen wie Prolog: Super für bestimmte Spezialaufgaben aber manchmal etwas unpraktisch und zerstreut...)
 - Scala: "Eduard mit den Scherenhänden" (Alles ein bisschen, eine Art Frankenstein aber kein Monster...)
 - Ruby: Marry Poppins...
 - usw.

...witzige Idee ☺

Inhaltlicher Fokus von Ihrem Team-Projekt

- Arbeiten Sie sich in "Ihre" Sprache ein (Doku, Bücher, Tutorials, weitere online Quellen, ...)
 - z.B. mittel "Seven Languages in Seven Weeks" (Es hat 6 Kopien in Bibliothek HSLU I, Erdgeschoss S41)
- Identifizieren Sie "spannende" (d.h. neue, ungewohnte, praktische, exotische, usw.) Sprachkonzepte und –konstrukte, z.B.:
 - Spezielle Syntax
 - Ungewohntes Konzept
 - Spezielles Sprachkonstrukt oder Sprachmittel
 - ...

...welche Sprachen schauen wir an?

- 3 Anforderungen an wählbare Programmiersprache:
 - Sollte eine "General Purpose Language" sein
 - Sollte frei verfügbar sowie +- relevant sein
 - konkret in den Top-50 (verhandelbar!) vom TIOBE-Index:
<https://www.tiobe.com/tiobe-index>
 - Haben wir in PCP noch nicht (vertieft) behandelt, d.h.
NICHT: C, Java, Prolog, Scheme
- Zur Auswahl stehen u.a.: Ceylon, Clojure, Cobol,
Dart, Eiffel, Erlang, Fantom, F#, Frege, Go, Groovy,
Haskell, Io, JavaScript, Kotlin, Lua, ML, Python, Ruby,
Rust, Scala, Scratch, TypeScript, Xtend, ...



TIOBE Index for November 2018

November Headline: the programming language of 2018 will be one of the top languages

There are only 2 months left before TIOBE will announce the programming language of the year 2018. There appear to be 5 candidates and these are all from the top 5: Java, C, C++, Python and Visual Basic .NET. Every year we hope that some shiny new programming language wins the award but for 2018 it will most probably be a well established language. There are new programming languages, but they just gain too little traction to deserve the title "programming language of the year". For instance, Apex jumped from position 46 to 36 last month, whereas Alice went from position 62 to 44 in the same time period. But these are all background battles, because both languages gained less than 0.5% in 2018. Let's see what language is going to win.

IMPORTANT NOTE. SQL has been added again to the TIOBE index since February 2018. The reason for this is that SQL appears to be Turing complete. As a consequence, there is no recent history for the language and thus it might seem that the SQL language is rising very fast. This is not the case.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

<https://www.tiobe.com/tiobe-index/>

Nov 2018	Nov 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.746%	+3.51%
2	2		C	14.396%	+5.10%
3	3		C++	8.282%	+2.94%
4	4		Python	7.683%	+3.20%
5	7	▲	Visual Basic .NET	6.490%	+3.58%
6	5	▼	C#	3.952%	+0.94%
7	6	▼	JavaScript	2.655%	-0.32%
8	8		PHP	2.376%	+0.48%
9	-	▲	SQL	1.844%	+1.84%
10	14	▲	Go	1.495%	-0.07%
11	19	▲	Objective-C	1.476%	+0.06%
12	20	▲	Swift	1.455%	+0.07%
13	9	▼	Delphi/Object Pascal	1.423%	-0.32%
14	11	▼	R	1.407%	-0.20%

<https://www.tiobe.com/tiobe-index/>

Rahmenbedingungen Team-Projekt

- Zweier-Teams
- Dozenten = Coach/Berater
- Abschluss: Präsentation & Bericht (inkl. Demo-Code)
- Termine einhalten / Abgaben beachten!
 - Siehe nächste Folien
 - Testatbedingung ("Aktive Teilnahme...")

Feste Projekt-Termine

- 29.11.: Gruppe & Sprache fixiert
 - 30.11.: Grober Themenfokus fixiert
 - 14.12.: Code-Demo
 - 18.12.: Abgabe Bericht, Folien & Code (Ilias)
 - 20. & 21.12.: Präsentationen & Code-Demo
 - Details siehe nächste Folien...
- Termine einhalten! ☺

Gruppe & Sprache (bis 29.11.)

- Gruppe & Sprache fixiert bis 29.11.
 - Welche zwei Studierenden beschäftigen sich mit welcher Sprache?
- Zwei Optionen:
 - a) Persönlich in der Vorlesung am 22. oder 23.11. im Gespräch mit Ruedi Arnold
 - b) Per Email an ruedi.arnold@hslu.ch bis 29.11.
 - Inhalt Mail: <Sprache>, <Studi1>, <Studi2>
- Gewählte Sprache wird von Ruedi Arnold bestätigt
 - Wir "vergeben" jede Sprache maximal 2x, FCFS

Bemerkung: Optionen zur Sprachwahl...

Wählen Sie eine Sprache, die...

...Sie schon immer kennen lernen wollten!

...in (Ihrer?) Zukunft wichtig werden könnte!

...Ihnen kryptisch erscheint!

...Spass macht!

...Sie überhaupt nicht kennen!

...Ihnen ein Kollege, Dozent oder Chef empfohlen hat

...was-auch-immer! ;-)

Hier haben Sie die Möglichkeit, eine Programmiersprache nach Wahl kennen zu lernen - Nutzen Sie diese Chance! ☺

30.11.: Grober Themenfokus fixiert (1/2)

- Sie bearbeiten 3 bis 7 interessante Sprachkonstrukte oder -konzepte
 - Auf welche Konstrukte und Konzepte "Ihrer" Sprache gehen Sie im Detail ein?
 - Unser Vorschlag pro Sprache siehe Git-Repo:
<https://gitlab.enterpriselab.ch/PCP/PCP-public-Code>
- 2 Optionen:
 - Sie übernehmen unseren Vorschlag...
 - Sie liefern einen eigenen Vorschlag...
 - Komplett neu bzw. Modifikation von unserem Vorschlag

30.11.: Grober Themenfokus fixiert (2/2)

- **Abgabe:** Ihre gewählten 3 bis 7 Fokuspunkte, jeweils inkl. kurze Erklärung (stichwortartig, ggf. inkl. Pseudo-Codesequenz)
 - Tragen Sie Ihre Fokuspunkte in einem eigenen Beitrag im PCP-Ilias-Forum ein!
 - Titel vom Beitrag gemäss folgender Namenskonvention:
"C++ von Roger Diehl & Ruedi Arnold"
- Entweder akzeptieren wir Ihre Fokuspunkte ODER...
- ...wir besprechen diese mit Ihnen bzw. Sie erhalten von uns per Email einen Alternativ-Vorschlag
 - D.h. wir behalten uns das Recht vor, Ihnen ggf. einzelne Punkte explizit vorzugeben

Die Themenliste

-> <https://gitlab.enterpriselab.ch/PCP/PCP-public-Code/blob/master/Sprach-Konzepte.md>

PCP: Themenvorschläge pro Programmiersprache

Dieses Dokument enthält den Vorschlag der Coaches an pro Sprache zu behandelnden Themen. Bitte beachten Sie dazu die entsprechenden Infos auf dem Foliensatz "Team-Projekt-Aufgabenstellung".

Hinweise, Fragen und Anmerkungen zu dieser Liste bitte in's [Ilias-Forum](#).

Clojure

- immutability and representation of state
- concurrency & software transactional memory
- persistent data structures
- the REPL

Cobol

- Datentypen & aggregierte Daten
- Programm-Divisionen
- Kontrollstrukturen
- Prozeduren

Crystal

- Everything is an object
- Union types
- Default values & named arguments

Selbständige Absprache bei gleicher Sprache

- Wir vergeben jede Sprache max. 2x
- Falls "ihre" Sprache zweimal gewählt ist, sprechen Sie sich bitte selbständig mit dem anderen Team ab
 - Ziel: möglichst wenig inhaltliche Überlappung bei Präsentation & Bericht
 - D.h.: Fokuspunkte disjunkt zwischen Teams aufteilen
 - Falls Sie den Wunsch haben, dass Team 1 vor Team 2 präsentiert: Meldung bis zum 14.12. per Mail an ruedi.arnold@hslu.ch

Bis 14.12.: Code-Demo

- Demonstration von Ihrem bis dahin erstellen Code & Code-Sequenzen:
 - Lösen und zeigen Sie uns eine Auswahl von bisherigen Programmieraufgaben aus dem Modul: min. 3 Aufgaben
 - Code-Beispiele, welche Ihrer 3 bis 7 Fokuspunkte möglichst verständlich, kompakt & schön illustrieren
- Vorzeigen: live Code-Demo & Coaching
 - Coaching: Code wird dem Coach live präsentiert, Diskussion und Beantwortung von allfälligen Fragen
 - Keine schriftliche Abgabe

18.12.: Abgabe

- Ihre finale Abgabe:
 - Präsentationsfolien (pdf-Format)
 - Bericht (pdf-Format)
 - Lauffähige Code-Beispiele
- Abgabe im Ilias-Briefkasten bis 18.12.
 - Alles zusammen-gezipppt in 1 Datei mit Namensschema **[Sprache]-[Stud1]-[Stud2].zip** (also z.B. "Java-Diehl-Arnold.zip") und hochgeladen in's Ilias

Erwarteter Inhalt ihrer Präsentation

- Kurze Einleitung (1 Folie)
 - z.B. über Vision, Geschichte, Verbreitung, ...
- Hauptteil: Die Sprache vorstellen (Ihre 3 bis 7 Fokuspunkte, inkl. Code-Demos)
- Ihr technisches Team-Fazit (1 Folie)
 - "Gute" Sprache? Würden Sie sie regelmässig Einsetzen wollen? Falls Ja: für was? Falls Nein: Wieso nicht?
- Ihr persönliches Fazit (je 1 Folie pro Team-Mitglied)
 - Persönliche Erkenntnisse? Erfahrungen? Whatelse?..

Anforderungen Bericht

- Qualität vor Quantität!
- Fokus auf wichtige/ interessante/ spezielle Sprach-Eigenschaften! (Was anders als Java?!)...
- Ergänzend zu den Folien
 - Falls interessant/relevant kurze Infos zu Vision, Geschichte & Verbreitung
 - **Hauptteil: Die Sprache vorstellen (Ihre 3 bis 7 Fokuspunkte, inkl. Verweise auf Ihren Demo-Code)**
 - Ihr technisches Team-Fazit
 - Persönliches Fazit (je min. 1 Abschnitt pro Team-Mitglied)

Anforderungen Code

- **Lauffähig!**
- Inkl. `Readme.txt`
 - Download, Setup, ggf. Einstellungen, usw. – Damit wir Ihren Code problemlos zum Laufen bringen können
- Min. drei gelöste Aufgaben aus den PCP-Übungen
- Min. je ein Code-Beispiel zu Ihren 3 bis 7 Punkten
 - Möglichst verständlichen & sauberen Code ☺

20. & 21.12.: Präsentationen & Code-Demo

- Abschluss-Präsentation inkl. Code-Demo
 - Zeit pro Team: max. 10' inkl. Demo
 - Live Code-Demo ist Pflicht
 - Danach max. 5' für Fragen
- Inhalt wie erwähnt
 - Bringen Sie "Ihre" Sprache in 10' auf den Punkt!
 - Machen Sie uns die Sprache und deren Spezialität(en) "schmackhaft" ☺

```
\int
_;"\
char*I,
*0[]={"",
"jistu","t"
"fdpoe","uij"
"se","gpvsui",\
"gigu","t"
"jyui","tfwi"
"oui","fjhiui",
"ojouu","ufouu",\
"fmfwfou","uxfmgu"
"i","!qbsusjehf!jo!"
"b!qfbssusff!xb\xb",\
"uxp!uvsumf!epwf"
"t-\xb","uisffigs"
"odi!ifot-!","gpvs!d"
"bmmjoh!cjset-!","gjwf"
"ihpmeljsjoh<\xb","tjylh"
"fft!b.mbzjoh-!","tfwfo!t"
"xbotlb.txjnjjoh~\xb","fjhiu"
"!n!bjetlb.njmjoh-!","ojof!mbe"
"jft!ebodjoh-!","ufo!m"
"pset!b.mfbqjoh~\xb","fm"
"fwfo!qjqfst!ajqjoh-!","ux"
"fmwf!esvnfnst!esvnjjoh-!","","
"Po!uf!","!ebz!pg!Disjntbn!l"
"z!usvf!mpwf!hbwf!up!nf\xb","boe"
"!");int putchar(int);int main(void)
}while(l<(sizeof 0/sizeof+0-2)/2-1){
    I=0=_?sizeof 0/sizeof*0-
    3:_<(sizeof(0)/sizeof*0-2)/2?
        sizeof 0/sizeof*0-2:===(sizeof(
        0)/sizeof*0-2)/2++l,0:<(sizeof(
        0)/sizeof*0)-3?_1===(sizeof(0)/
        sizeof*0-2)/2?sizeof 0/sizeof*0-1:_-1
        :_<sizeof(0)/sizeof*0-2?+1:_<sizeof(0)/
        sizeof*0-1?1+(sizeof 0/sizeof*0-2)/2:(
        sizeof(0)/sizeof*0-2)/2];while(*I){putchar(
        *I++-1);}
    return 0;}
```

<http://digitrode.ru/images/2013/09/18/P2.png>

PCP Team-Projekt

- Alles klar? – Fragen, Anmerkungen?
 - Aufgabenstellung = diese Folien
- Wenden Sie jederzeit gerne an Coaches bei Fragen oder Unklarheiten
 - Direkt in der Vorlesung oder via Ilias-Forum
- Wir freuen uns auf spannende Projekte & interessante Erkenntnisse!

→ Happy Coding & Exploring... ☺

