

プログラム言語論

櫻井彰人

ポイント

- ◆ プログラム言語は、解くべき問題、問題の存する世界、問題の解き方を記述するもの
- ◆ プログラム言語には、問題、世界、解き方をどう観るかという思想が反映している
 - 勿論、技術的な観点(記述の効率性・容易性、間違いにくさ、検証容易性、実行効率 etc.)も非常に大切

プログラム言語とは

- ◆ コンピュータ計算機と人間との間に介在する言語
- ◆ 人間がコンピュータに行わせたいことを表現したもの
 - すなわち、ある種の命令・指令を表したもの

コンピュータに行わせたいこと？

- ◆ 計算か？
- ◆ 情報処理か？
 - 「情報」も「処理」も曖昧ではあるが、「計算」との違いは分かる？

「計算」とは

- ◆ 計算機誕生のきっかけは大量の計算を行わせたかったこと
 - 弾道ミサイルの開発(第二次世界大戦後)
 - 暗号解読(第二次世界大戦中)
- ◆ 計算 = 大量の四則演算

「情報処理」とは

- ◆ 例を挙げよう
 - ✓ 銀行の様々な取り引きの記録と実行
 - ✓ POSデータの蓄積と利用
 - ✓ JR/航空機/ホテルの予約
- ◆ 特徴: 計算は単純、量は膨大、関係が複雑
 - 計算するより、記憶しておこう、引き出そう
 - データ間関係を整合的に処理

もっと身近な例

- ◆ 文書作成: 計算は単純、データ量も少ない
「計算」というより「情報処理」
- ◆ イラスト作成: 同上
- ◆ 表計算ソフトは？
- ◆ ゲーム機:
 - 実は計算の山: 画像処理
 - 最近では、GPUの高機能化・低価格化
 - GPUがゲーム以外の計算に用いられるようになった
 - スーパーコンピュータもGPUを使用する

脱線

「計算」から「情報処理」へ

- ◆ 初期(1960年代)は計算が主体であったが
- ◆ 1970年代から様子が変わってきて、
- ◆ 1980年代以降は、情報処理が中心となる
 - 勿論、巨大計算の需要は存在する
 - シミュレーション: 気象予測、自動車開発、建造物の設計
 - 画像処理: CG、VR、ゲーム
 - ビッグデータ、深層学習、機械学習全般
- ◆ この変化に伴い、プログラム言語に対する考え方が変化

「プログラム言語」の見方

- ◆ 主たる記述対象の変化:
 - ✓ 計算の指令 → データ(間関係)の記述
 - 原因: 解くべき問題分野の変化
- ◆ 記述方法の洗練化
 - ✓ 単純な列 → 構造化、隠蔽化、自律化
 - 原因: 規模の変化、副次的目的の変化
 - 誤り難さ、可読性、保守容易性、再利用性、、、

発展は複合している

- ◆ 前記の発展は、設計方法論の発展とからんで、相互に影響しあっている
- ◆ 各言語、方法論の系統樹を描くのは非常に難しい

大切な背景: いわずもがなですが

- ◆ Niklaus Wirth:
Algorithm+Data Structure = Program
- ◆ 「計算」(解法)と「データ」は計算機で解く問題の2大要素

プログラム言語を構成するもの

- ◆ 構文論(syntax)と意味論(semantic)
それだけでなく
- ◆ 表現論・設計論
 - プログラミング・パラダイム(井田哲雄)
 - 問題の存在する世界(データ、データ間関係)、世界の変化をどのようにモデル化し記述するか、解法をどのように記述するか

「プログラム言語」の発展の見方

- ◆ 二つをみないといけない
- ✓ 計算のモデル化と記述方法の発展
 - syntax と semantics がより深く関与
 - 解を得るアルゴリズムの記述
 - プログラミング作法
- ✓ データのモデル化と記述方法の発展
 - 表現論・設計論がより深く関与
- ◆ この2つの上にたつのが型理論

アルゴリズム

- ◆ 計算機で具体的に解を構成する必要がある
 - 品目別月次売り上げ推移、明日の天気、、、
- ◆ 計算機は解の求め方を知らないゆえ、
 - ある範囲の問題に対して解を求める方法をプログラムしておくこともできるが、本質は同一
- ◆ 解の作り方を手順を追って記す必要がある
 - この「手順」がアルゴリズムであり、それをある言語で記述したものがプログラム

アルゴリズム(補足)

- ◆ 解は構成的に記述されないと(多くの場合)困る。
 - 例外は、研究目的。人間が介在して試行錯誤が許される時
- ◆ 従って、通常はプログラムは原始帰納的に記述される(後述)

プログラム言語への観点

- ◆ 計算のモデル
- ◆ データのモデル
- ◆ 記述方法
 - 多くのものと関連している
 - 計算のモデル, データのモデル
 - コンパイラの作り方
 - プログラミング規律
 - 制御: 構造化
 - 宣言: scope規則

言語概念: 時が移れば、、、

- ◆ 寓話
 - 私が最初にプログラムをしたのは1970年代
 - Fortran全盛時代; 関数の再帰呼び出しはなし
 - アルゴリズムやデータ構造に対する常識:
 - 再帰呼び出しはアイデアとしてはよいが、非効率
 - 同じことは、配列でスタックを真似れば可能
 - 今日: 再帰呼び出しは普通
- ◆ 教訓
 - 世界は変わる; 多くのアイデアを理解することが肝要
- ◆ 現在の事例: 関数の引数渡し
 - Cで関数を渡すのに、STLの“function objects”のように自分で関数閉包を作る

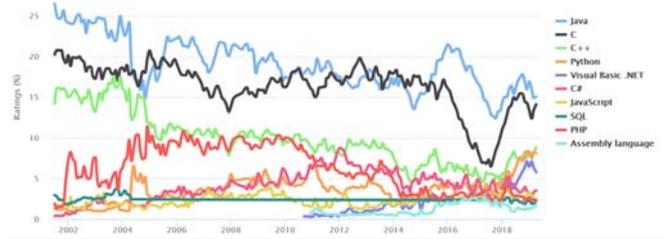
よく使われている言語 (II)

順位	1年間の変動	プログラム言語	比率	1年間の変動
1	≡	Java	15.035%	-0.74%
2	≡	C	14.076%	+0.49%
3	≡	C++	8.838%	+1.62%
4	≡	Python	8.166%	+2.36%
5	↑	Visual Basic .NET	5.795%	+0.85%
6	↓	C#	3.515%	-1.75%
7	↑	JavaScript	2.507%	-0.99%
8	↑	SQL	2.272%	-0.38%
9	↓	PHP	2.239%	-1.98%
10	↑↑	Assembly Languages	1.710%	+0.05%

TPC index: based on world-wide availability of skilled engineers, courses, and third party vendors, determined by using Google, MSN and Yahoo! search engines
2019年3月現在 <http://www.tiobe.com/tpci.htm>

Apr-19	Apr-18	Change	Programming Language	Ratings	Change
1	1		Java	15.04%	-0.74%
2	2		C	14.08%	0.49%
3	3		C++	8.84%	1.62%
4	4		Python	8.17%	2.36%
5	6	▲	Visual Basic .NET	5.80%	0.85%
6	5	▼	C#	3.52%	-1.75%
7	8	▲	JavaScript	2.51%	-0.99%
8	9	▲	SQL	2.27%	-0.38%
9	7	▼	PHP	2.24%	-1.98%
10	14	▲	Assembly language	1.71%	0.05%
11	18	▲	Objective-C	1.51%	0.25%
12	17	▲	MATLAB	1.29%	-0.17%
13	10	▼	Ruby	1.28%	-0.74%
14	16	▲	Perl	1.27%	-0.26%
15	11	▼	Delphi/Object Pascal	1.26%	-0.70%
16	12	▼	R	1.18%	-0.63%
17	13	▼	Visual Basic	1.06%	-0.74%
18	19	▲	Go	1.01%	-0.17%
19	15	▼	Swift	0.98%	-0.56%
20	68	▲	Groovy	0.93%	0.82%

よく使われている言語 (III)



Hacker news poll (2012/4/5)

Top 20 Most Popular Programming Languages Among Hacker News Readers

1	Python	3240 points
2	Ruby	1805 points
3	JavaScript	1482 points
4	C	1032 points
5	C#	886 points
6	PHP	711 points
7	Java	590 points
8	C++	570 points
9	Haskell	567 points
10	Clojure	476 points
11	CoffeeScript	373 points
12	Lisp	341 points
13	Objective C	335 points
14	Perl	335 points
15	Scala	253 points
16	Scheme	199 points
17	Other	193 points
18	Erlang	169 points
19	Lua	150 points
20	Smalltalk	128 points
21	Assembly	110 points
22	SQL	110 points
23	Actionscript	106 points
24	OCaml	88 points
25	Groovy	80 points
26	D	78 points
27	Shell	75 points
28	ColdFusion	50 points
29	Visual Basic	46 points
30	Delphi	45 points
31	Forth	41 points
32	Tcl	34 points
33	Ada	28 points
34	Pascal	28 points
35	Fortran	25 points
36	Rexx	13 points
37	Cobol	11 points

<http://siliconangle.com/blog/2012/04/06/top-20-most-popular-programming-languages-among-hacker-news-readers/>
<https://news.ycombinator.com/item?id=3746692>

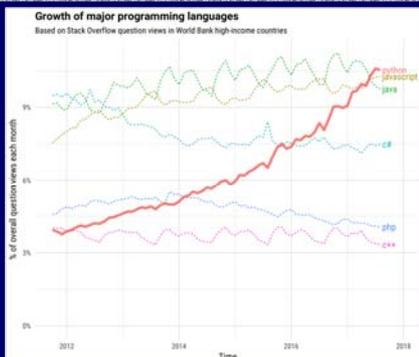
Hacker news poll (2018)

Top 10 Most Popular Programming Languages
July 2018 Hacker News Hiring Trends

1	Python
2	JavaScript
3	Go
4	Ruby
5	Java
6	C++
7	PHP
8	Scala
9	Swift
10	Kotlin

<https://www.hntrends.com/2018/jul-top-ten-programming-languages.html?compare=&compare=&compare=&compare=>

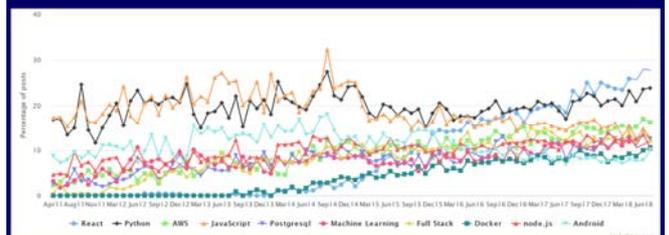
Ovreflow questions



<https://news.codecademy.com/why-learn-python/>

Hacker News Hiring Trends

Jun 2018 - Tracking the most popular programming languages, databases and software development tools in whoishiring discussion threads.



プログラム言語の流れ

- ◆ 多様性の増大
- ◆ 過去20年ほどの商用分野での流れ
 - Java, C#, ... など型安全な言語の普及
 - スクリプト言語, webアプリ目的の他の言語
 - Ruby が延びた(今は停滞)。Ruby on Rails
 - Python の伸びが著しい
- ◆ 教育面での流れ
 - 入門用言語としては、Java が C に代わってきた
 - 計算機内部のデータや制御の表現を強調しなくなった
- ◆ 研究・開発の流れ
 - モジュール性
 - Java, C++: モジュール機能が標準的に
 - プログラム解析
 - エラーの自動検出、プログラミング環境、コンパイル
 - 分離と安全性
 - サンドボックス, language-based security, ...

何を学ぶか? -- 基礎からのアプローチ

- ◆ プログラミング言語の基礎概念
- ◆ 有力なパラダイムの基礎
 - 命令型と関数型、オブジェクト指向等
 - 歴史的変遷
 - 失敗例と成功例
- ◆ 言語の記述とその意義と意味
- ◆ 設計上のトレードオフ