

Спор о первом языке программирования: окончательное решение

Виталий Брагилевский



Профессиональная
конференция
разработчиков мобильных
приложений



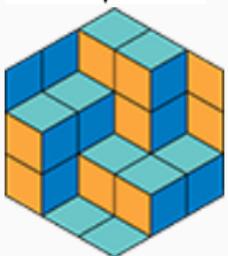
Eto ya



JetBrains
(Санкт-Петербург)



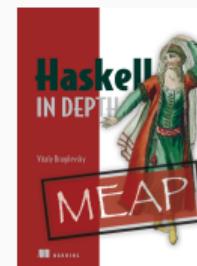
Санкт-Петербургский
государственный
университет



Факультет математики
и компьютерных наук



Glasgow Haskell Compiler
Steering Committee



Haskell in Depth

The most useless things of all time.



**Спор
о первом языке
программирования**

Дейкстра о важности языка программирования (1972)

«...the tools we are trying to use and the language or notation we are using to express or record our thoughts are the major factors determining what we can think or express at all!»

А вот вы лично встречали программистов...

- испорченных Фортраном
- пострадавших от BASIC
- раздавленных C++
- задушенных Java





**MORE PERSONAL
STORIES**

LET'S GO!

План

История вопроса

Глас народа

Окончательное решение

История вопроса

Проблема Дика Вексельблата

Richard L. Wexelblat. **The consequences of one's first programming language.** Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems, 1980.

- Влияет ли первый ЯП на то, как человек программирует в дальнейшем?

- Письмо в журнал (1978): «Ошибки молодости, или Действительно ли Бейсик вреден для вашего здоровья?»

- Письмо в журнал (1978): «Ошибки молодости, или Действительно ли Бейсик вреден для вашего здоровья?»

I received almost 50 replies broken down roughly as:

- 30 addressing my points
- 12 not relevant
- 4 pointer to other sources
- 2 attacking me for raising the issue in the first place

Некоторые результаты

- Есть языки, более подходящие для обучения (Pascal, PL/C) и менее подходящие (COBOL, FORTRAN).

Некоторые результаты

- Есть языки, более подходящие для обучения (Pascal, PL/C) и менее подходящие (COBOL, FORTRAN).
- Негативные эффекты от плохого обучения значительнее, чем от плохого языка.

Некоторые результаты

- Есть языки, более подходящие для обучения (Pascal, PL/C) и менее подходящие (COBOL, FORTRAN).
- Негативные эффекты от плохого обучения значительнее, чем от плохого языка.
- Один из респондентов отметил, что начинал с Бейсика, но его способности к программированию ничуть не пострадали.

**Исследований,
посвящённых выбору
ЯП, сотни!**

Есть даже работы, обобщающие работы по выбору ЯП!

Kevin Parker, Bill Davey. **The History of Computer Language Selection.** (In Arthur Tatnall. *Reflections on the History of Computing: Preserving Memories and Sharing Stories*, AICT-387, Springer, pp.166-179, 2012, IFIP Advances in Information and Communication Technology (SURVEY))

Ключевые факторы (1)

Прагматика

- распространение в промышленности
- популярность на рынке (влияние коммерческих субъектов)

Ключевые факторы (2)

Педагогика

- как избежать сложности промышленных окружений?
- ориентация на problem-solving (vs изучение ЯП)
- наличие учебных материалов и преподавателей

Ключевые факторы (3)

Язык и библиотеки

- реализация определённой парадигмы
- работа с графикой и интерактивность

Сравнением языков тоже занимаются очень часто

Linda Mannila and Michael de Raadt. **An Objective Comparison of Languages for Teaching Introductory Programming.** (In *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006.*)

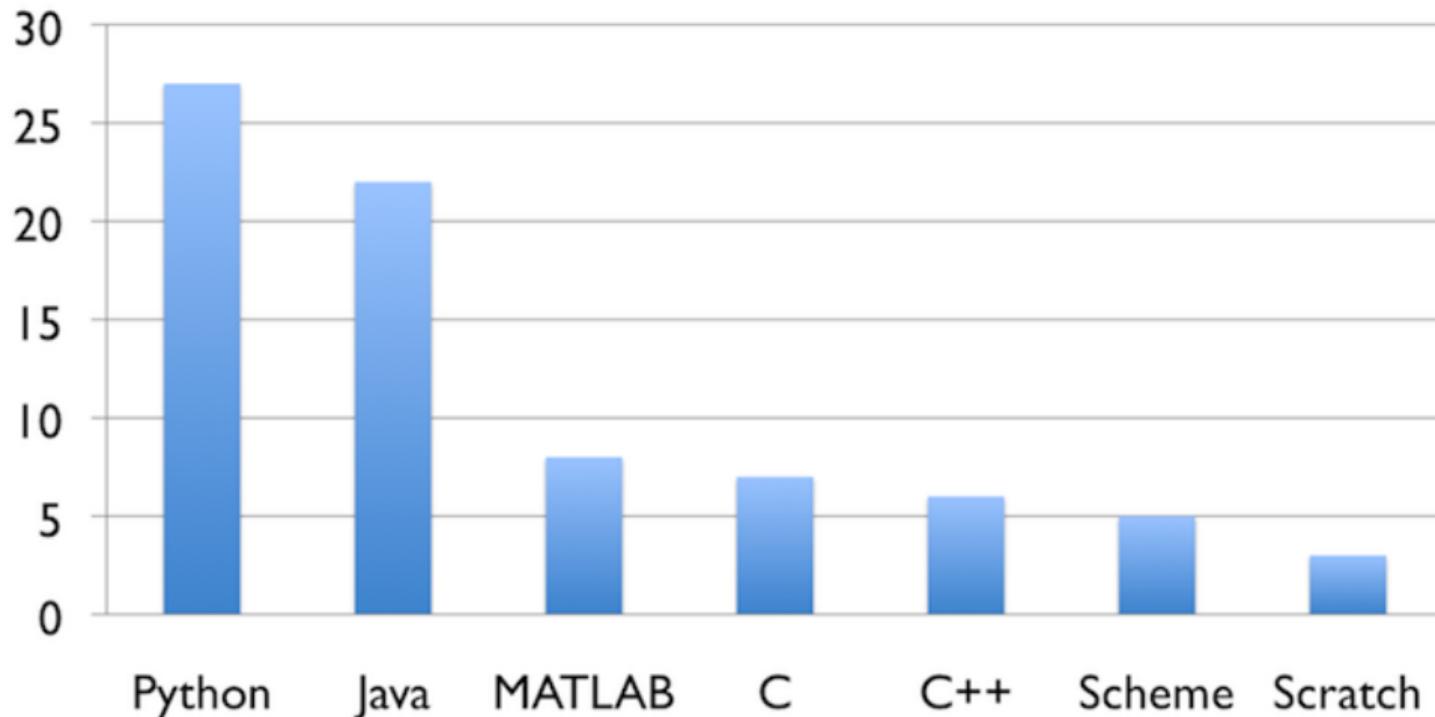
Группы критериев оценки ЯП по Mannila&de Raadt

1. Применимость в обучении
2. Язык и среды разработки
3. Поддержка и доступность
4. Использование за пределами вводного курса

	C	C++	Eiffel	Haskell	Java	JavaScript	Logo	Pascal	Python	Scheme	VB
Learning											
Is suitable for teaching (§2.1.1)			✓		✓	✓	✓	✓	✓		
Can be used to apply physical analogies (§2.1.2)			✓		✓	✓	✓	✓	✓		✓
Offers a general framework (§2.1.3)	✓	✓	✓		✓	✓	✓	✓	✓		✓
Promotes a design driven approach for teaching software (§2.1.4)			✓	✓	*1		✓				✓
Design and Environment											
Is interactive and facilitates rapid code development (§2.2.1)				✓		✓		✓	✓	✓	✓
Promotes writing correct programs (§2.2.2)			*2	✓	*2				*2		
Allows problems to be solved in "bite-sized chunks" (§2.2.3)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Provides a seamless development environment (§2.2.4)			✓		*1						
Support and Availability											
Has a supportive user community (§2.3.1)	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Is open source, so anyone can contribute to its development (§2.3.2)											✓
Is consistently supported across environments (§2.3.3)	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Is freely and easily available (§2.3.4)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Is supported with good teaching material (§2.3.5)	✓	✓		✓		✓	✓	✓	✓	✓	✓
Beyond Introductory Programming											
Is not only used in education (§2.4.1)	✓	✓	✓		✓	✓			✓		✓
Is extensible (§2.4.2)	✓	✓	✓		✓				✓		✓
Is reliable and efficient (§2.4.3)	✓	✓	✓		✓	✓		✓	✓		✓
Is not an example of the QWERTY phenomena (§2.4.4)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Authors' Score	8	11	15	6	14	9	9	7	15	8	9

А что там по факту?

Number of top 39 U.S. computer science departments
that use each language to teach introductory courses



Analysis done by Philip Guo (www.pgbvine.net) in July 2014, last updated 2014-07-29

Кое-чему места не нашлось:

- Популярные среди исследователей функциональные ЯП со статической типизацией Haskell и ML
- Широко распространённые в веб-разработке ЯП с динамической типизацией JavaScript, Ruby, PHP
- Завязанные на платформу промышленные языки Objective-C (Apple) и C#/Visual Basic (Microsoft)

Philip Guo. Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. Июль 2014, блоги САСМ, <http://bit.ly/python-at-uni>.

Глас народа

Я провёл опрос в твиттере (bit.ly/tw-first-lang):

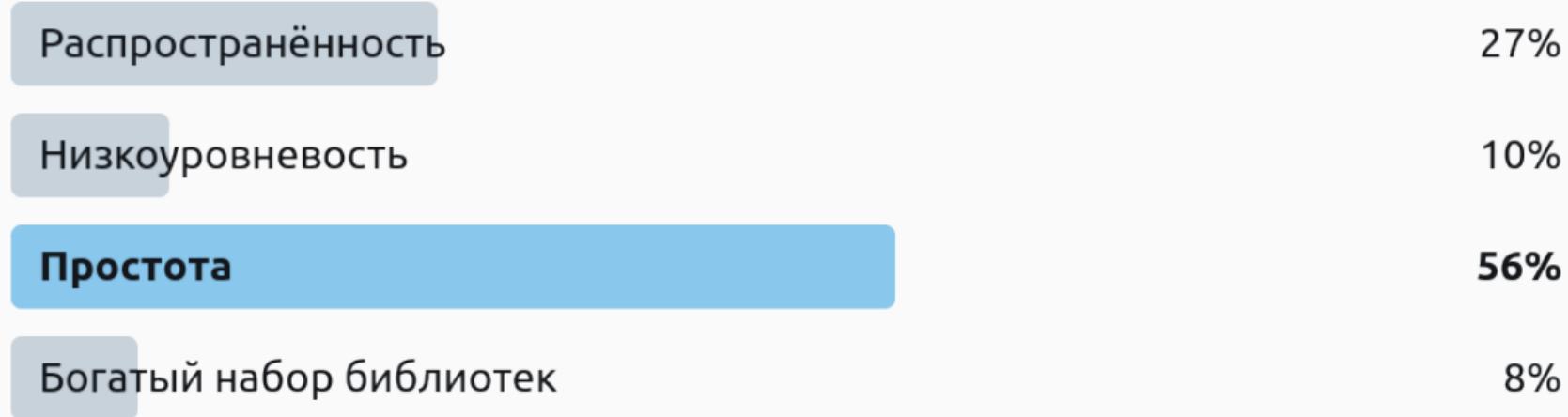
Какое свойство языка программирования
вы считаете наиболее важным при выборе
ПЕРВОГО языка для обучения программированию?

Я провёл опрос в твиттере (bit.ly/tw-first-lang):

Какое свойство языка программирования
вы считаете наиболее важным при выборе
ПЕРВОГО языка для обучения программированию?

- Распространённость
- Низкоуровневость
- Простота
- Богатый набор библиотек

Самое важное: простота!



1 294 голоса · Конечные результаты

Несколько интересных реplaев

Несколько интересных реплаев



Dmitrii Kovanikov @ChShersh · 23 ч

▼

В ответ @_bravit

Высокоуровневость и интерактивность. Чтобы получить результат работы программы и иметь возможность посмотреть его можно было с минимальным числом шагов. Программирование - это не про создание переменных, циклов или рекурсий, это про решение проблем средствами ЯП.



Несколько интересных реплаев



Artem Pelenitsyn @ulysses4ever · 18 ч

В ответ @[_bravit](#)

Наличие специализированных методических (учебники, задачники, туториалы, Q&A форумы) и технических (простые и внятные IDE с минималистичным интерфейсом, стекперы/дебаггеры с GUI) средств.

1



8



Несколько интересных реплаев



Andrey Miskov @andreymiskov · 23 ч

В ответ @_bravit



Предсказуемость: не отвлекаться на null/undefined и 0.1 + 0.2.

Постепенное увеличение возможностей: никакого ООП и мутаций, пока не освоишь данные и функции. Типа как в Racket/HtDP или в Pyret/PAPL.



Несколько интересных реплаев



Baygashevsky V. @umogSlayer · 20 ч

В ответ @_bravit

Лично я считаю, что С был и остаётся самым лучшим для более фундаментального изучения.

На мой взгляд, простой и достаточно интуитивный синтаксис, и много разных концепций (память, указатели, стэк, аллокации). Только одно НО, ни в коем случае не C89, минимум C99, а лучше C11.

1



Несколько интересных реплаев



BeiZero @BeiZero · 20 ч

В ответ @_bravit

Я бы брал Scala. Она достаточно простая на базовом уровне, распространённая, статически типизированная, мультипарадигмальная. Из серьёзных минусов для новичка это интероп с жабой т.к. чтобы уйти с базового уровня на более менее продвинутый нужно будет жабу учить и не только.



1



BeiZero @BeiZero · 20 ч

Если именно по свойствам, то мультипарадигмальность и статическая типизация.



Несколько интересных реплаев



Сатанархист @satarchist · 22 ч

В ответ @[_bravit](#)

Типизация + простота, особенно рисования и работы с интерактивной консолью

1



7



Kirill Sukhomlin @kirilloid_ru · 20 ч

Поддерживаю. Нужна возможность видеть результаты своей работы для положительного подкрепления мотивации к обучению. Впрочем, эта задача может быть решена дополнительным интерактивным учебником (у Rust очень неплохой).

1



1



Языки программирования как тоталитарные секты

- Гиперактивный прозелитизм
- Неприятие критики (и чужого мнения вообще)

Языки программирования как тоталитарные секты

- Гиперактивный прозелитизм
- Неприятие критики (и чужого мнения вообще)
- *Пример: присваивание как " = " vs " := "*

И ещё парочка реплаев



Lev Walkin @levwalkin · 23 ч

В ответ @_bravit

1/распространённость

1



13



Lev Walkin @levwalkin · 14 ч

(единица, делённая на распространённость)

1



И ещё парочка реплаев



Vadim ChelyshOv @_dos65 · 11 ч

В ответ @_bravit

Logo?

1



Vadim ChelyshOv @_dos65 · 11 ч

На самом деле вообще пофиг, просто хочу попасть на слайды доклада

1



Окончательное решение

Что отличает промышленный язык от учебного?

- слишком много способов решить задачу
- сообщения об ошибках предполагают, что язык уже известен
- другие цели (пример: компиляция кода с ошибками)

Что отличает промышленный язык от учебного?

- слишком много способов решить задачу
- сообщения об ошибках предполагают, что язык уже известен
- другие цели (пример: компиляция кода с ошибками)
- **учить языку легче, чем программированию!**

Структура и Интерпретация Компьютерных Программ

Второе издание

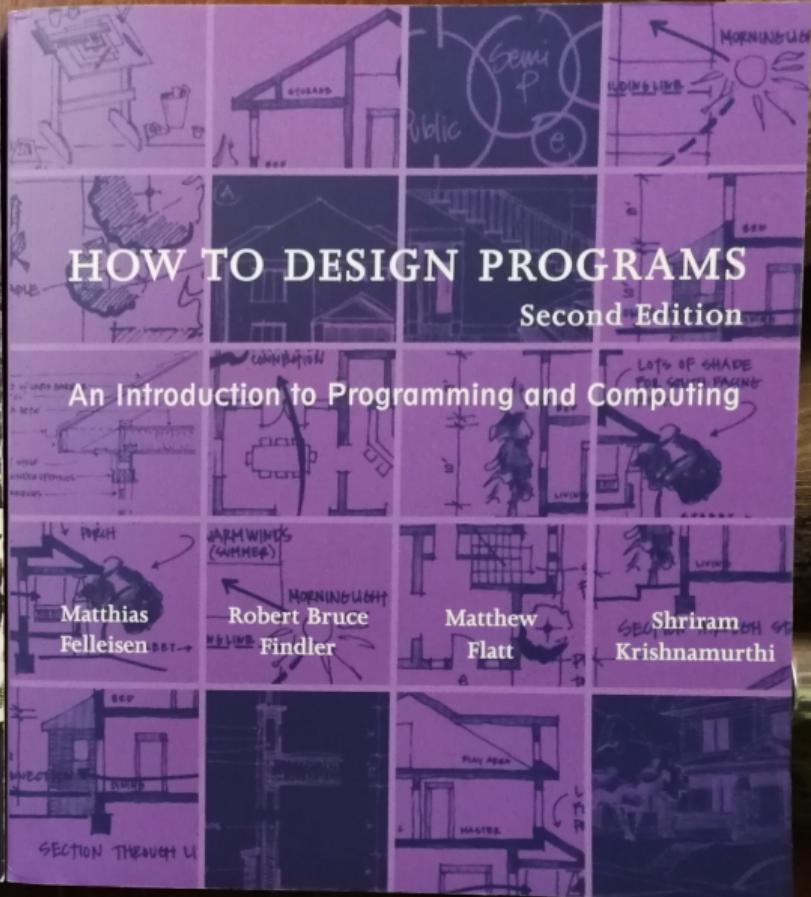


Харольд Абельсон
Джеральд Джей Сассман
при участии Джули Сассман

HOW TO DESIGN PROGRAMS

Second Edition

An Introduction to Programming and Computing



HOW TO DESIGN PROGRAMS

AN INTRODUCTION TO PROGRAMMING AND COMPUTING

SECOND EDITION

Matthias Felleisen
Robert Bruce Findler
Matthew Flatt
Shriram Krishnamurthi

N. Kelly
enjoy Oregon
—Matthew Flatt
Dad for,
29 Nov 2017

How to Design Programs: важные принципы

1. Чтение постановки задачи и выявление в ней необходимых компонентов
2. Перенос задачи на структуру программы
3. Тесты как часть программы
4. Проектирование, а не кодирование

Кредо Маттиаса Феллайзена: языки для обучения

1. BSL (определение функции, вызов функции, условная операция, арифметика)

Кредо Маттиаса Феллайзена: языки для обучения

1. BSL (определение функции, вызов функции, условная операция, арифметика)
2. *SL:
 - списки
 - рекурсия
 - функции высшего порядка
 - изменяемое состояние (присваивание)

Кредо Маттиаса Феллайзена: языки для обучения

1. BSL (определение функции, вызов функции, условная операция, арифметика)
2. *SL:
 - списки
 - рекурсия
 - функции высшего порядка
 - изменяемое состояние (присваивание)
3. Racket (язык семейства Lisp)

Безымянный - DrRacket*

Файл Правка Вид Язык Racket Вставка Tabs Справка

Безымянный (define ...) ➔ Выполнить Остановить

```
(require 2htdp/image)

(place-image (circle 5 "solid" "green")
             50 80
             (empty-scene 100 100))
```

Добро пожаловать в [DrRacket](#), версия 6.12 [3m].
Язык: [Начинающий студент](#); memory limit: 128 MB.



>

All expressions are covered Show next time?

Начинающий студент 4:2 276.19 MB 

Книга «Programming and Programming Languages»

- Авторы: Krishnamurthi, Lerner, Politz
- Язык программирования: Pyret
- Примеры глав:
 - Introduction to Tabular Data (4)
 - Interactive Games as Reactive Systems (11)
 - Examples, Testing, and Program Checking (12)
 - Graphs (19)
 - Processing Programs: Parsing (23)
 - Reasoning about Programs: A First Look at Types (27)
 - Objects: Interpretation and Types (32)

Язык Pyret (code.pyret.org)

The screenshot shows a Pyret code editor interface. At the top, there's a toolbar with a logo, 'Connect to Google Drive', 'Run' (highlighted in blue), and 'Stop'. The main area contains the following code:

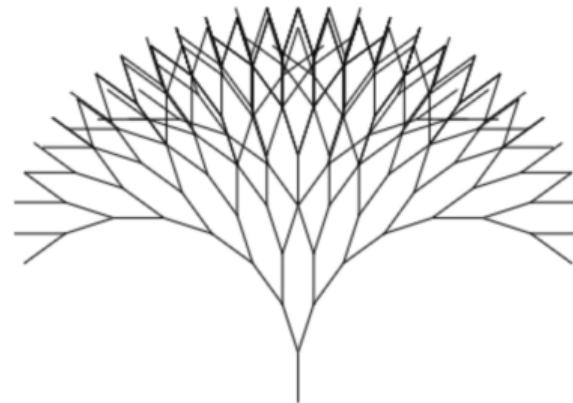
```
1 ▾ data BinTree:  
2   | leaf  
3   | node(value, left, right)  
4 end  
5  
6 ▾ fun tree-sum(t):  
7   doc: "Считает сумму значений в узлах"  
8   cases (BinTree) t:  
9     | leaf => 0  
10    | node(v, l, r) =>  
11      v + tree-sum(l) + tree-sum(r)  
12    end  
13 where:  
14   tree-sum(leaf) is 0  
15   node4 = node(4, leaf, leaf)  
16   tree-sum(node(5, node4, leaf)) is 9  
17 end  
18
```

To the right, a blue box says: "Looks shipshape, both tests passed, mate!" Below it, a green box for the `tree-sum` function shows: "tree-sum" and "Show Details". It also states: "All 2 tests in this block passed." At the bottom, two command-line examples are shown:

```
>>> tree-sum(node(0, node  
3  
>>> tree-sum(node(0,  
node(1, leaf, leaf),  
node(2, leaf, leaf)))
```

Вот ещё подход: CodeWorld

```
1 import CodeWorld
2
3 tree :: Integer -> Picture
4 tree 0 = blank
5 tree n = polyline [(0,0),(0,1)]
6     & translated 0 1 (
7         rotated (pi/10) (tree (n-1))
8         & rotated (- pi/10) (tree (n-1)))
9
10 main :: IO ()
11 main = drawingOf (tree 8)
```



123%

[Sign In](#)[Editor](#)[Guide](#)[Ask a Question](#)[Report a Bug](#)[Start Recording](#)[Share](#)[Inspect](#)[Stop](#)[Run](#)

Наш ответ заокеанским коллегам!



Современное программирование на языке Pascal

- Разработка Станислава Михалковича и его студентов (Южный федеральный университет, Ростов-на-Дону).
- Pascal нового поколения: `foreach`, внутриблочные описания переменных, `case` по строкам, длинные целые, обобщённые классы и подпрограммы, интерфейсы, перегрузка операций, лямбда-выражения, исключения, сборка мусора, методы расширения.
- Простой доступ к любым возможностям платформы .NET.

PascalABC.NET

File Edit View Program Tools Plugins Help

Eratosthenes.pas Life.pas Program558.pas* •Hanoi.pas* [Run]

{global

```
106 SetWindowSize(x3+x1, y0+50);
107 CenterWindow;
108 Window.Title := 'Tower of Hanoi';
109 Font.Size := 14;
110 Font.Name := 'Arial';
111 end;
112 begin
113   Tower of Hanoi
114   Init;
115   Init;
116   Loc;
117   Draw;
118   Move;
119 end.
```

The number of moves = 255

Output Window

Output Window Error List Compiler Messages

Compilation is completed successfully (118 lines)

Ln 119 Col 5

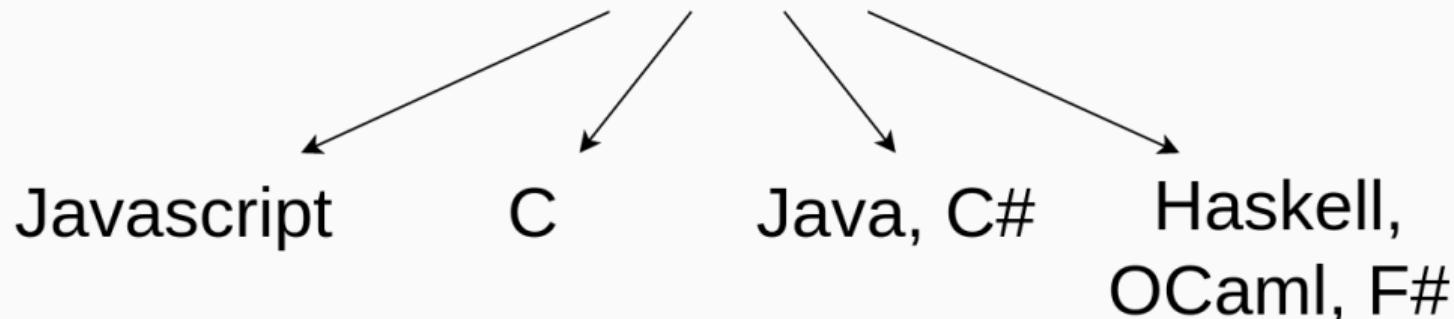
И всё-таки, окончательное решение

И всё-таки, окончательное решение

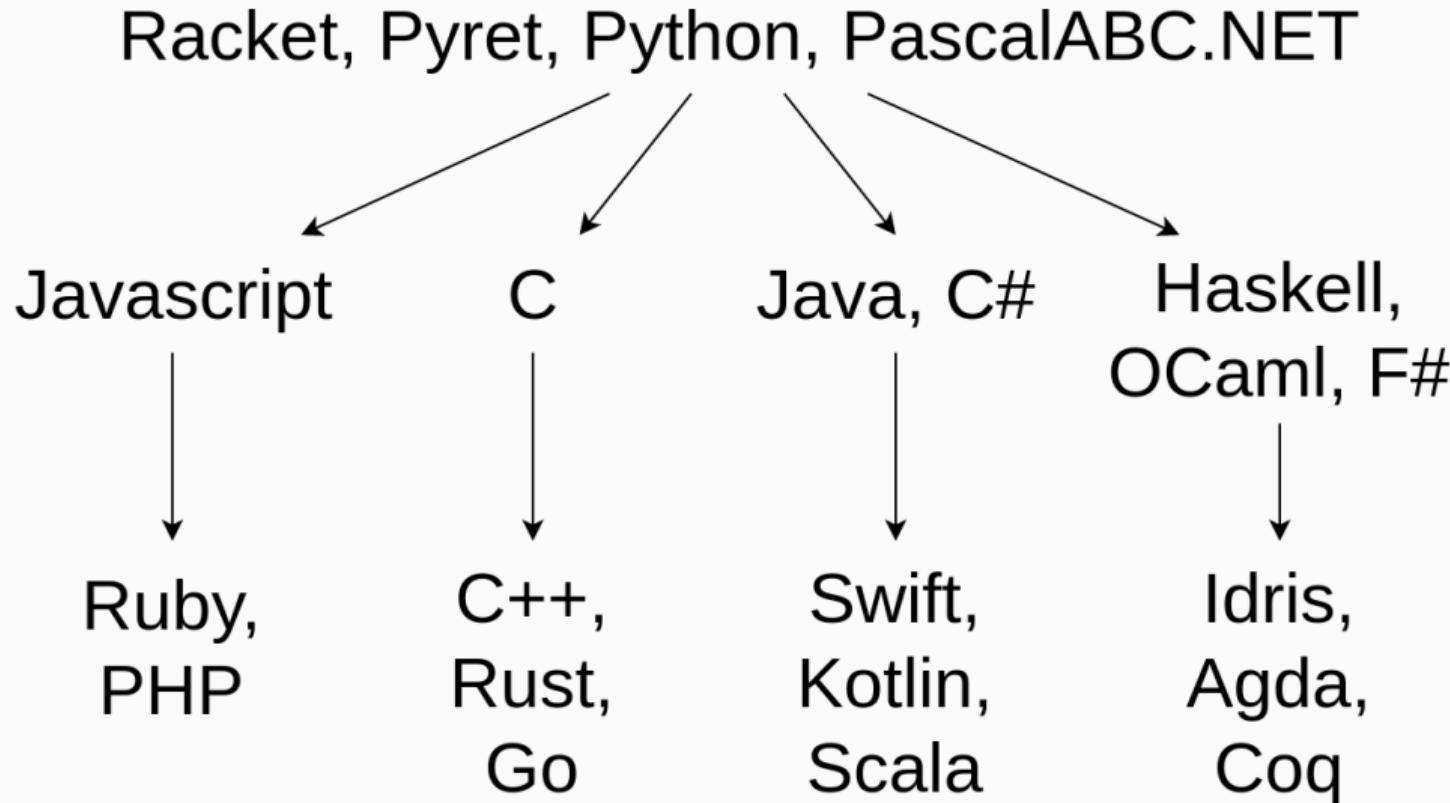
Racket, Pyret, Python, PascalABC.NET

И всё-таки, окончательное решение

Racket, Pyret, Python, PascalABC.NET



И всё-таки, окончательное решение



Q&A

Виталий Брагилевский

bravit111@gmail.com

   [_bravit \(ru\)](#)

 [VBragilevsky \(en\)](#)

 [bravit111](#)

 [bravit_about \(channel, ru\)](#)

