

# Преподавание языков программирования при обучении программистов

---

Виталий Брагилевский

23 октября, 2020

# Обо мне



JetBrains  
(Санкт-Петербург)



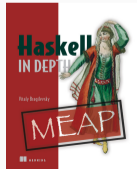
Санкт-Петербургский  
государственный  
университет



Факультет математики  
и компьютерных наук



Glasgow Haskell  
Compiler Steering  
Committee



Haskell  
in Depth

# Я в образовании

- С 2000 года работал учителем информатики в лицее, затем в детской компьютерной школе.
- С 2003 года работал в Ростовском государственном университете на мехмате (затем ЮФУ, ИММКН)
- С 2005 года участвовал в организации на факультете направления «Информационные технологии» (позднее «ФИИТ»).
- С 2019 года работаю в СПбГУ на факультете МКН, участвую в организации программы «Современное программирование» по направлению «Прикладная математика и информатика».
- Получал гранты Фулбрайта и Потанина для преподавателей.

# Я обучал многим языкам

- Turbo Pascal, Delphi
- PascalABC(.NET)
- C, C++
- Assembler
- Java
- Kotlin
- C#
- Visual Basic.NET, VBA
- Perl
- Python
- PHP
- Javascript
- Bash, Awk
- Julia
- Prolog
- Haskell
- Idris

**А это важная тема?**

---

# Важно ли, что речь об обучении программистов?

- Мало кому достаточно одного языка на протяжении всей карьеры.
- Невозможно обучить заранее всем необходимым языкам.
- Следует научить наиболее «типичным» языкам, облегчающим освоение новых языков.
- При этом язык важен сам по себе, а не только как инструмент для решения задач.

# Что такое язык программирования?

- Инструмент
- Способ выражения мыслей (и самого мышления!)
- Стиль работы

# А чему мы вообще должны учиться?

- Язык программирования
- Стиль программирования
- Решение задач



# Пример: язык Python

*Python идеален, если мы готовим:*

- математиков
- биологов
- ХИМИКОВ
- ЛИНГВИСТОВ
- историков
- ...

*Python плох для обучения:*

- программистов

*Почему?*

- устаревший синтаксис
- неудачная система типов
- слишком большая свобода

Переход Python → другой язык гораздо сложнее и опаснее, чем в обратную сторону!

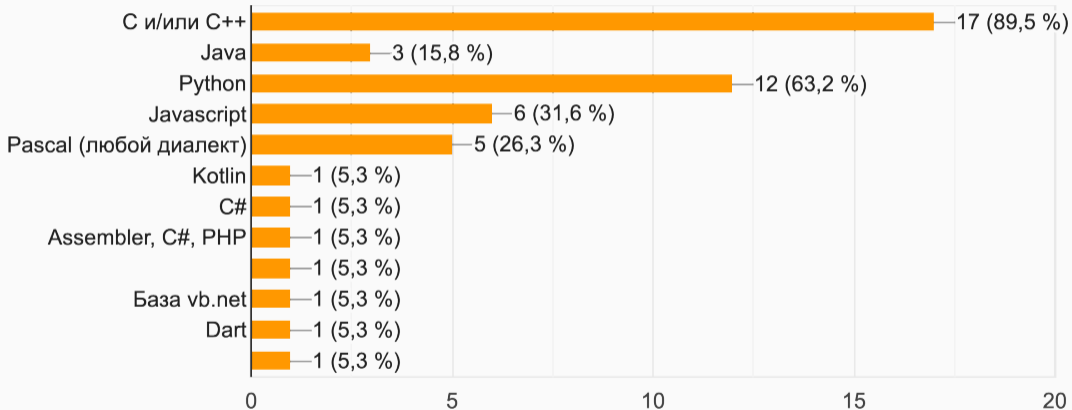
## Пример: язык C++

- Очень сложный язык, много тёмных углов
- Очень большой язык, как насчёт C++17?
- Очень легко написать «работающую», но некорректную программу
- Очень легко научить фичам, которые нельзя использовать в промышленном коде (обычные указатели?)
- Невозможно изучить «полностью»
- Практически невозможно быстро перейти к решению прикладных задач

# Язык программирования в первом семестре университета

---

# Студенты на входе (СП, МКН СПбГУ)



# Проблемы и решения

- Учить языку, который уже все или почти все «знают»?
- Переучивать на профессиональные подходы к написанию кода?
- А что делать с теми, кто не умеет программировать совсем или умеет, но на другом языке?
- *Идея:* проще взять менее знакомый студентам язык и сразу учить писать на нём хорошо.
- Примеры: PascalABC.NET, Julia, Kotlin.

# Соображение 1: а что такое «хорошо»?

## Характеристики качества кода

- Удобство сопровождения
- Гибкость
- Портируемость
- Возможность повторного использования
- Удобочитаемость
- Тестируемость
- Понятность

## Соображение 2: Паскаль устарел!

### Метод Монте-Карло на PascalABC.NET

```
begin
```

```
  var n := 10000000;
```

```
  var pp := Range(1,n)
```

```
    .Select(x->Rec(Random(), Random()))
```

```
    .Where(p->sqr(p.Item1)+sqr(p.Item2)<1)
```

```
    .Count/n*4;
```

```
  Print(pp);
```

```
end.
```

## Соображение 3: хотим решать полезные задачки!

- Извините, в первом семестре есть дела поважнее.
- Можно быстро перейти к изучению внешних библиотек (только сложновато найти что-то достаточно простое, не требующее более глубокого знания языка).



А что важнее?

**Надо учить писать  
хороший код!**

## Другие (сомнительные) соображения

- Java – говорят, ООП владеет миром, а Java – главный объектно-ориентированный язык!
- Javascript, PHP – и пусть идут работать с весны первого курса!
- Язык ассемблера, С – студенты должны знать, как оно всё устроено внутри!
- Haskell – у нас есть крутой специалист по Haskell!

# Языки программирования в учебном плане

---

## Цитаты великих людей

- «Я могу изучить любой язык программирования за две недели»
- «Все языки программирования одинаковые»
- «Языкам программирования учить не нужно, их легко можно изучить самостоятельно»
- «Нужно учить решать задачи»
- «Вот документация, изучите и приходите»
- «Вот задачи с автоматической проверкой, как наберёте N баллов, приходите»

**Я не согласен!**

# Подход СП МКН СПбГУ (первый курс)

- Сентябрь-октябрь первого курса:
  - «Основы программирования» (Kotlin, качество кода)
  - «Основы Linux» (немного Bash)
- Ноябрь-декабрь:
  - Введение в ООП (Kotlin)
  - Низкоуровневое программирование (C++, но почти как C)
- Также Python используется для примеров в курсе дискретной математики — читать легко, даже не изучая
- Второй семестр:
  - Kotlin
  - C++
  - Алгоритмы и структуры данных (язык любой, но C/C++ быстрее)

## Подход СП МКН СПбГУ (2–4 курсы)

- Третий семестр: Haskell (расширяем кругозор)
- Четвёртый семестр: Prolog (в рамках курса математической логики – продолжаем расширять кругозор)
- Третий курс: Python в курсе машинного обучения
- Дисциплины по выбору в 5–8 семестрах: Python, Scala, Java, C#, Rust, языки для веба (Javascript, PHP), ...

# Базовые принципы

- Несколько существенно разных обязательных языков
- Обучение последующим языкам с учётом наличия серьёзной базы
- Изучение вглубь в дисциплинах по выбору
- Чем позднее язык в учебном плане, тем больше его изучение ориентировано на промышленное использование



# Примеры наборов обязательных языков

- Kotlin, C/C++, Haskell, Prolog
- PascalABC.NET, C#, C/C++, Haskell
- Racket, Java, Rust, Javascript
- ...



# Приходите на круглый стол!

- Суббота, 24 октября, 15:40
- Преподавание языков программирования при обучении программистов
- Участники круглого стола:
  - Павел Егоров (ФИИТ УрФУ, СКБ Контур)
  - Станислав Михалкович (ИММКН ЮФУ)
  - Денис Москвин (НИУ ВШЭ в Санкт-Петербурге)
  - Иван Пономарёв (МФТИ, КУРС)

# Вопросы и ответы

---

Виталий Брагилевский  
bravit.guru



Слайды: [bit.ly/bravit-teach-pls](https://bit.ly/bravit-teach-pls)