Зачем нужен еще один язык программирования?

Роман Елизаров для TechTrain 2018



Обо мне



- В индустрии программирования с 2000 года
- Разрабатывал всякое высоконагруженное ПО @ Devexperts
- Преподаю параллельное программирование @ Санкт-Петербургский ИТМО
- Председатель жюри

 @ Northern Eurasia Contest / ICPC
- Занимаюсь библиотеками для Kotlin @ JetBrains

Самые популярные языки?

TIOBE

- Количество веб-страниц про "xxx programming"
- Top 3: Java, C, C++

PYPL

- Частота поиска по "xxx tutorial"
- Top 3: Python, Java, JavaScript

RedMonk

- GitHub проекты + StackOverflow тэги
- Top 3: JavaScript, Java, Python

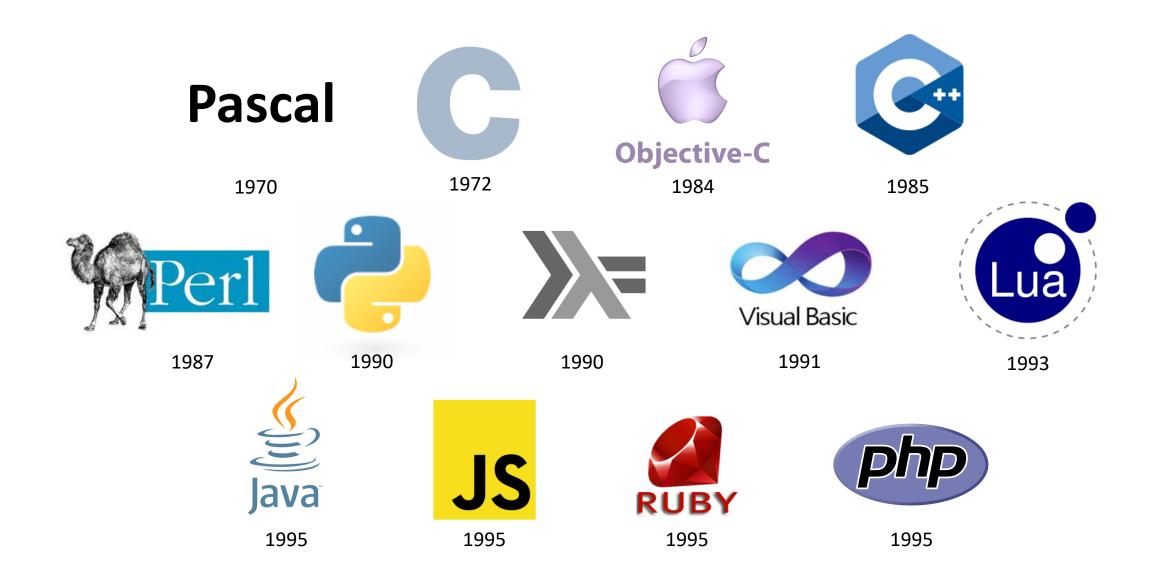
Языки программирования



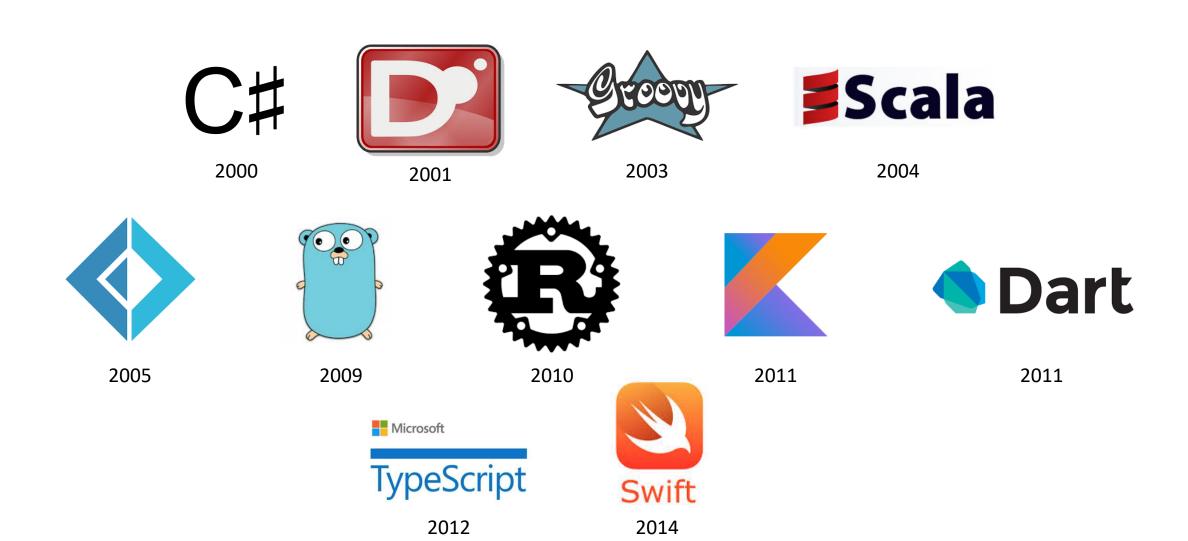


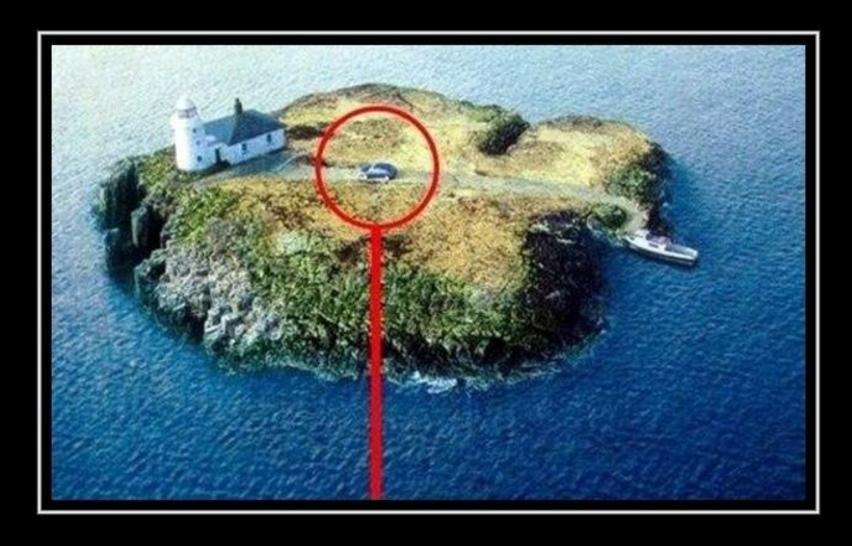


Универсальные языки программирования

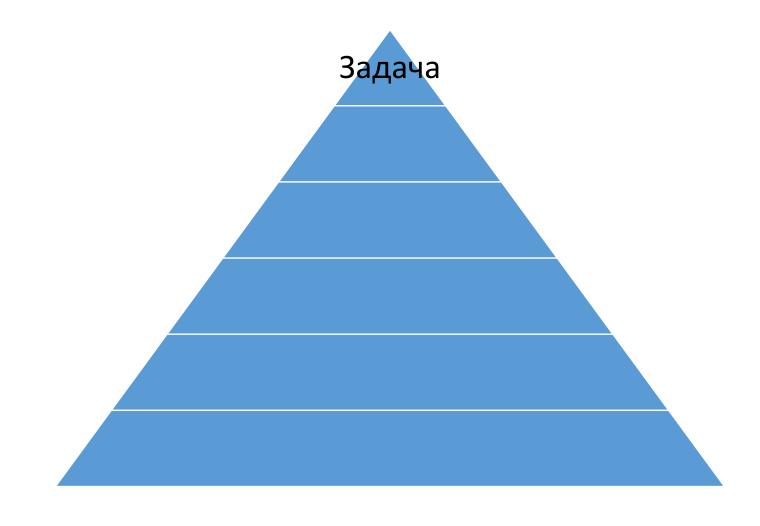


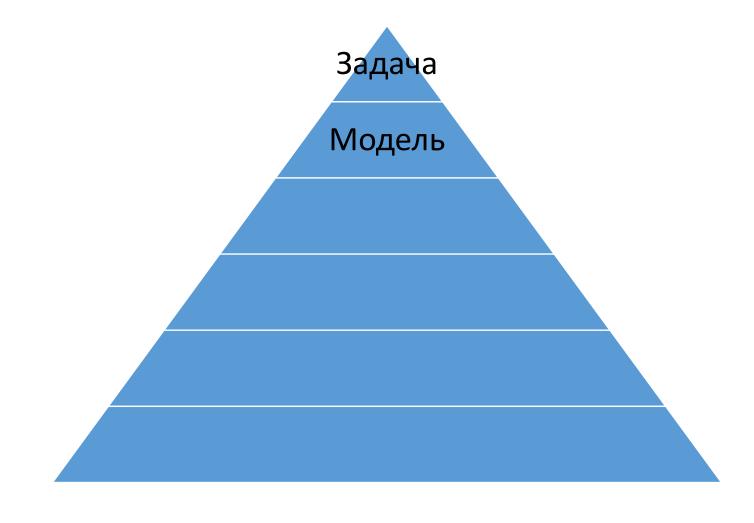
XXI век





ОДИН ВОПРОС зачем?















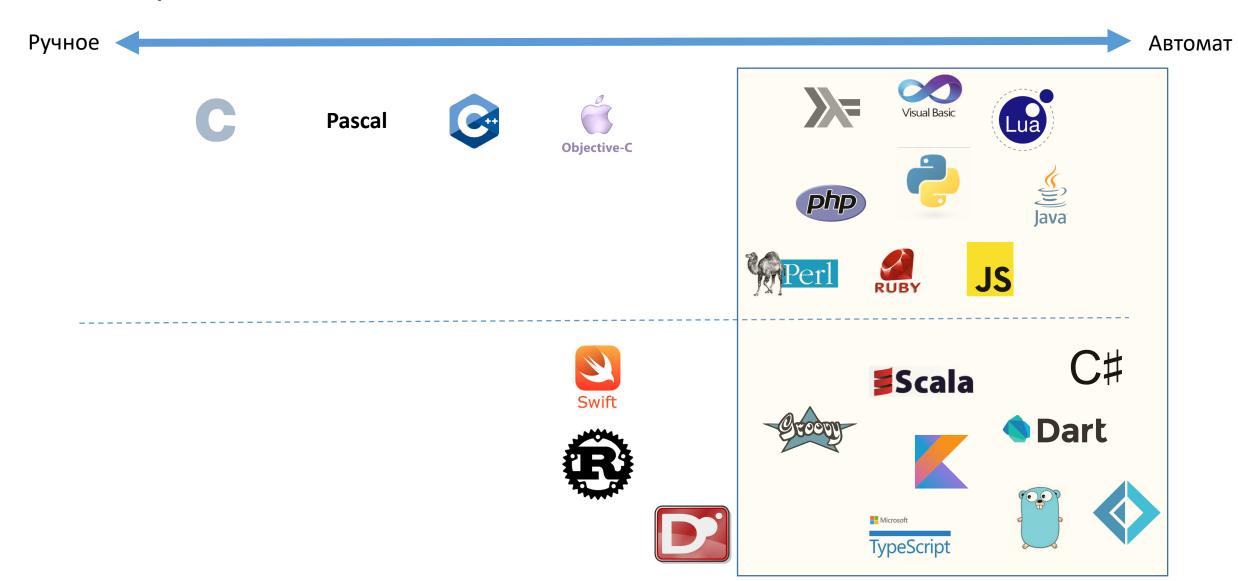




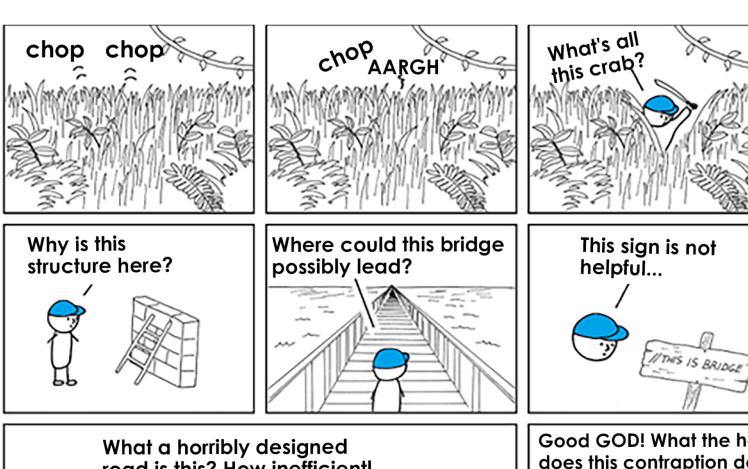
Управление памятью

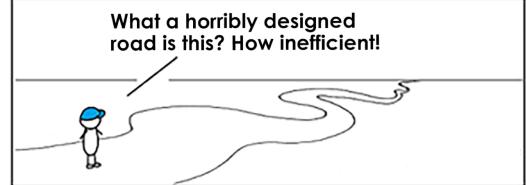


Управление памятью



Что программист делает бо́льшую часть времени?







```
function updateOrder()
function updateOrder($1, $2)
function updateOrder(order, address)
function updateOrder(order: PlainOrder, address: MailAddress): PlainOrder
function updateOrder(order: PlainOrder, address: MailAddress?): PlainOrder
```

Типизация

Динамическая

Статическая



JS





























Типизация

Динамическая





JS



php











































Типизация

Динамическая

Статическая









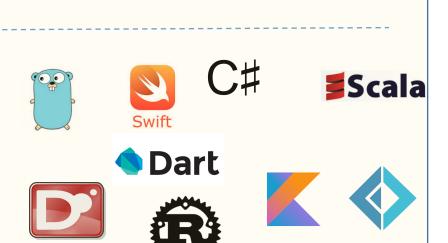












Масштабирование разработки



Слишком мало типизации

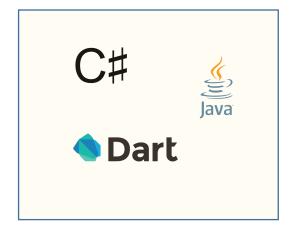
```
Map items = new HashMap();
```

Слишком много типизации?

```
Map<Warehouse, List<OrderItem>> items =
    new HashMap<Warehouse, List<OrderItem>>();
```

Вывод типов!

var items = new HashMap<Warehouse, List<OrderItem>>();



Вывод типов!

auto items = ...

C#



Dart



Вывод типов!

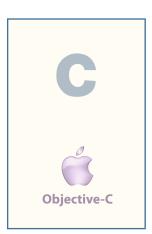












Было раньше

```
int count = ...
double average = ...
List<String> strings = ...
Map<Warehouse, List<OrderItem>> items = ...
```

Стало с выводом

```
var count = ...
var average = ...
var strings = ...
var items = ...
```

Стало с выводом

```
var count = ...
var average = ...
var strings = ...
Map<Warehouse, List<OrderItem>> items = ...
```

Типы справа!

```
var count = ...
var average = ...
var strings = ...
var items: Map<Warehouse, List<OrderItem>> = ...
```

Где тип?

Слева





































Где тип?

Слева

Справа

































Читать код...

```
Map<Warehouse, Double> totalWeight(List<Order> orders) {
    var result = new HashMap<Warehouse, Double>();
    for (var order : orders) {
        for (var item : order.getItems()) {
            var sum = result.get(item.getWarehouse());
            if (sum == null) {
                sum = 0.0;
            result.put(item.getWarehouse(), sum + item.getWeight());
    return result;
```

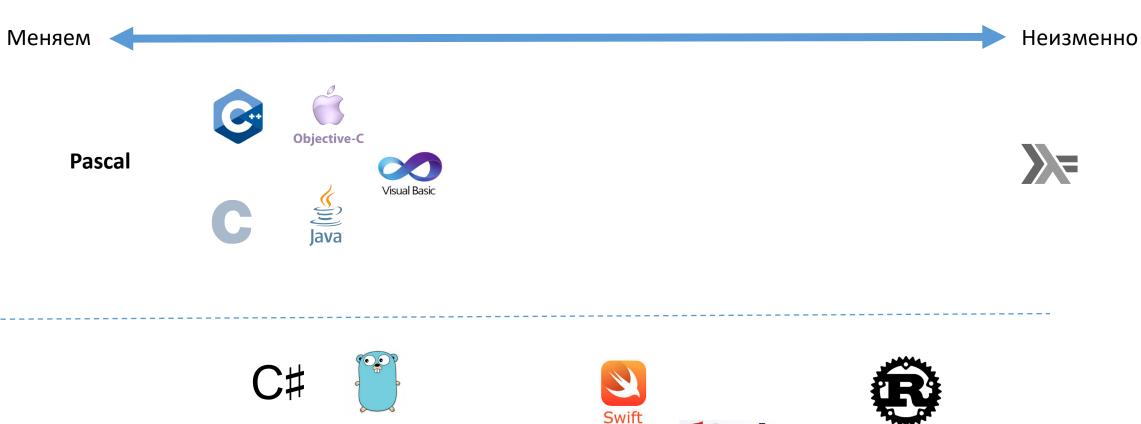
Лучше

Лучше

```
fun totalWeight(orders: List<Order>): Map<Warehouse, Double> =
    orders
        .flatMap { it.items }
        .groupingBy { it.warehouse }
        .fold(0.0) { acc, item -> acc + item.weight }
```

Make val not var

Переменные или значения по умолчанию









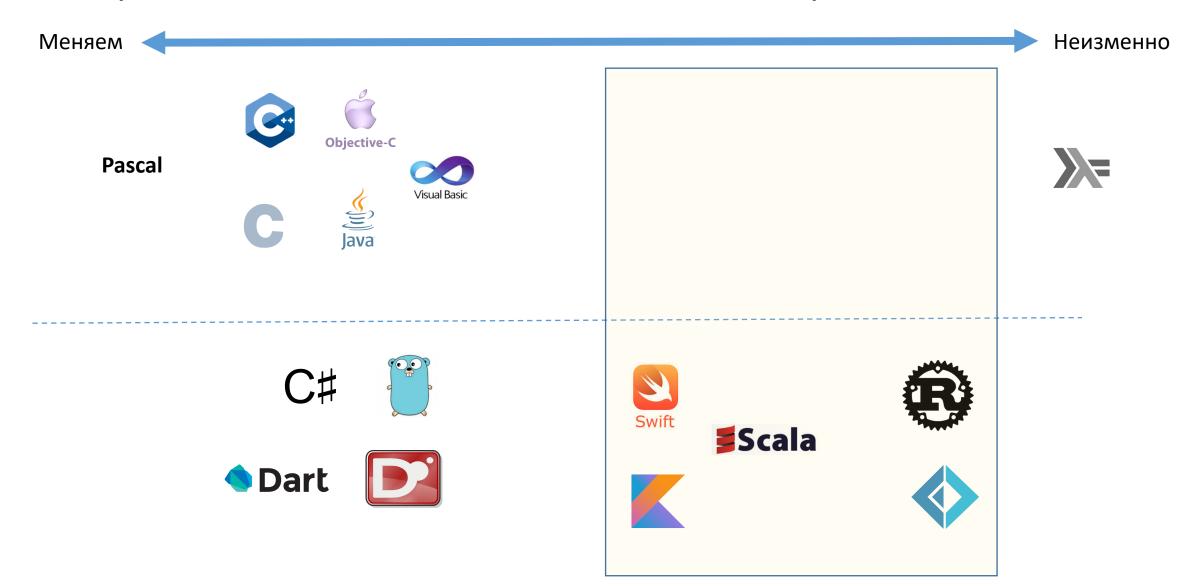








Переменные или значения по умолчанию



Где предел сложности типов?

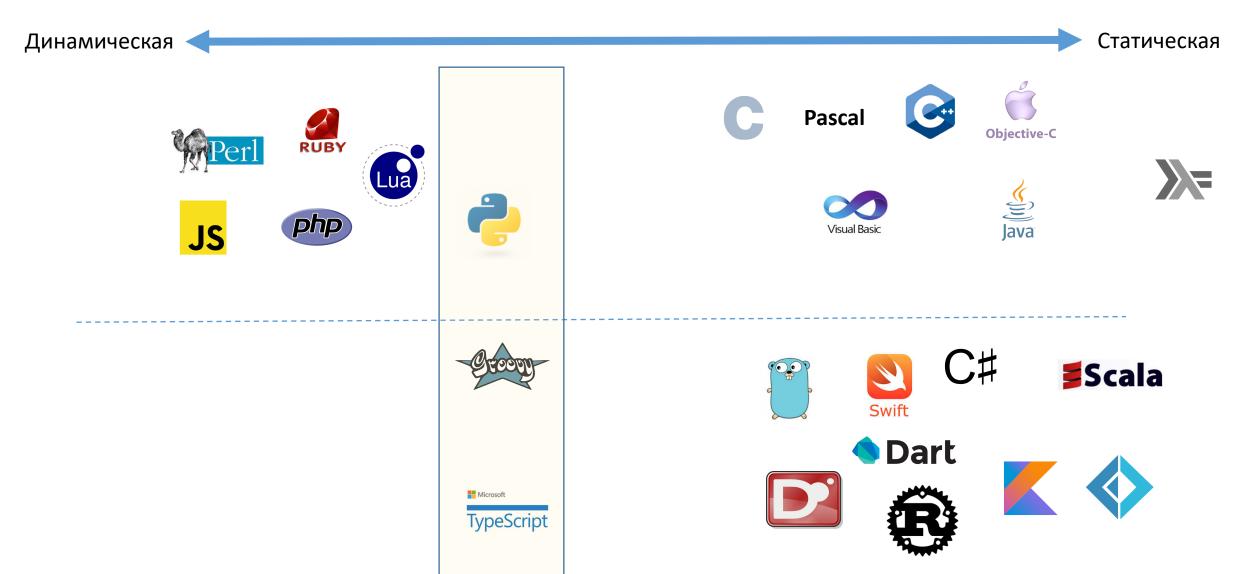


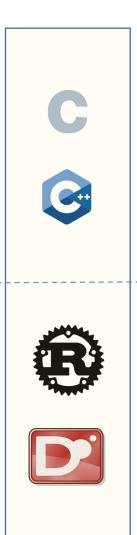


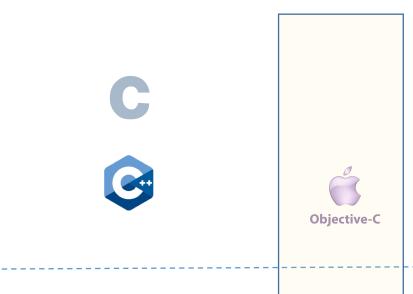
Фичи, фичи... А что вообще надо?



Динамическая экосистема



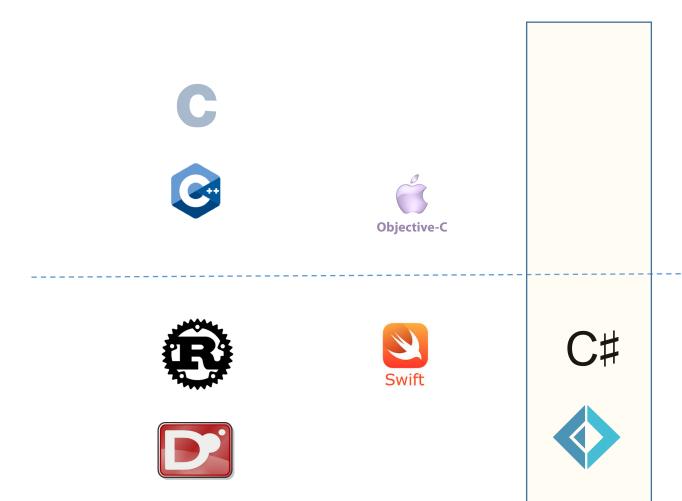










































И что в сухом остатке?









2011

2003 2004



Как завести более строгие типы?



Доменно-специфичные языки











```
<body>
<div>Hello, world!</div>
</body>
```

```
fun HTML.render() {
    body {
        div { +"Hello, world!" }
    }
}
```

Мораль?

Прогресс в языках не остановить



Спасибо. Вопросы?

