Независимая научно-практическая конференция «Разработка ПО 2011»



31 октября - 3 ноября, Москва

# Функциональное программирование для разработки распределенных, облачных и веб-приложений на языке F#

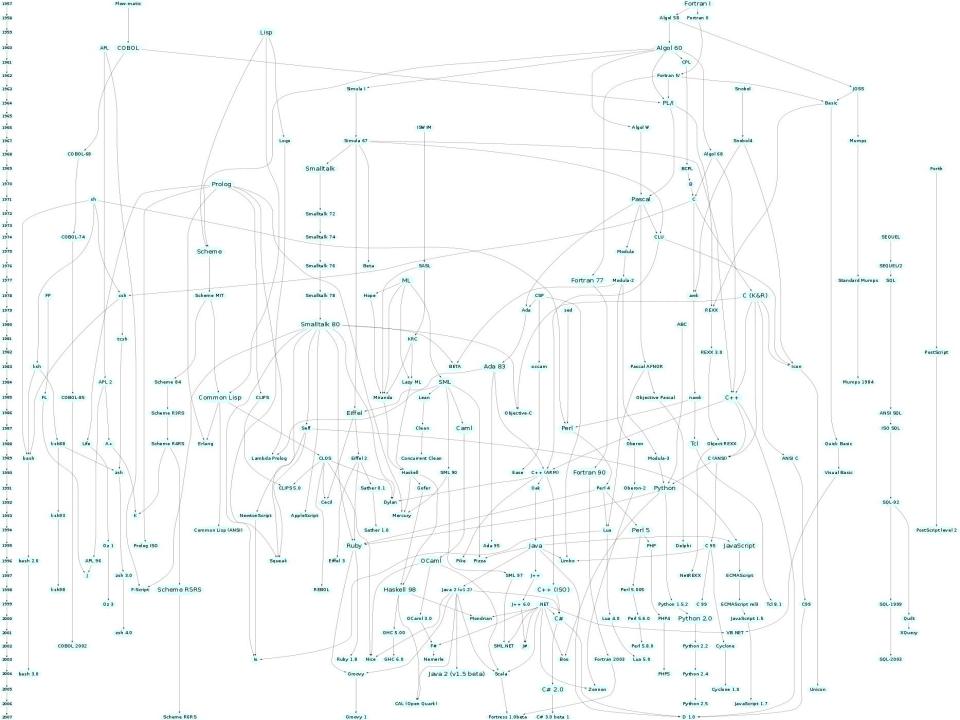
Дмитрий Сошников

Майкрософт Россия / НИУ ВШЭ

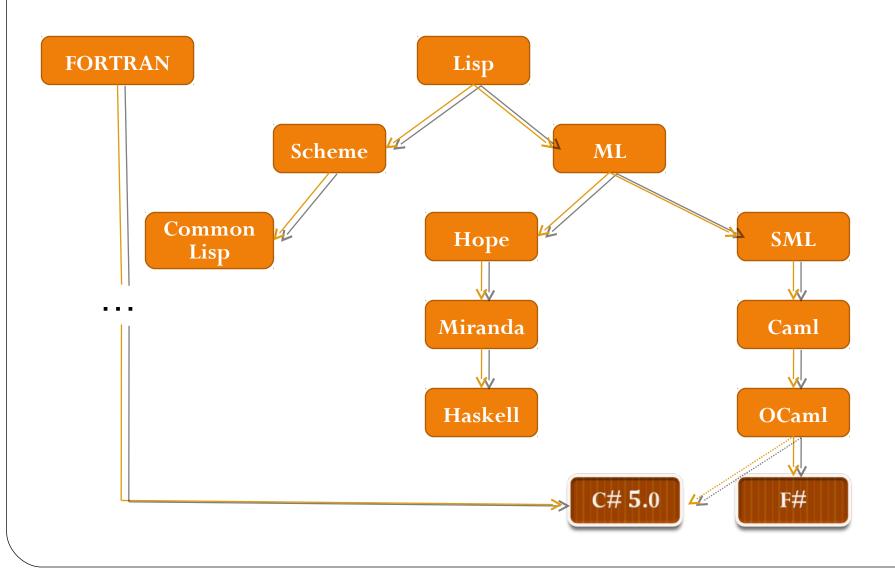
http://twitter.com/shwars dmitryso@microsoft.com

## О чем пойдёт речь?

- Функциональное программирование на практике
  - Недавно на habr: создание веб-сайта на Common Lisp
- Про что поговорим
  - Немного истории и философии
  - Немного про F#: Построение фрактала на F# в 2+10 строчек кода
  - Концепция 3 экранов
  - F# в веб-программировании: ASP.NET, Windows Azure
  - Распределенное программирование на F#: Quotations
  - WebSharper
  - О книге «Функциональное программирование на F#»
- Q&A



## Откуда взялся F#?



#### Зачем?

 John Hughes, Why Functional Programming Matters, 1984

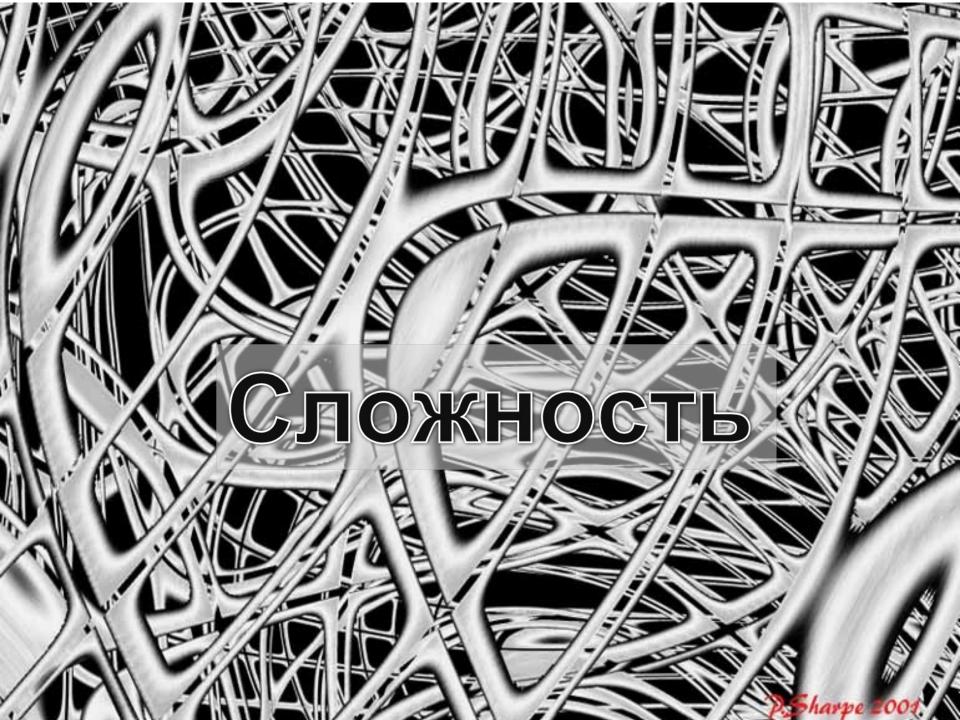


Более выразительный язык (меньше писать, больше думать)

Нет побочных эффектов => нет ошибок

Нет переменных => проще параллелизм





### Deloitte.

Manufacturing

## The challenge of complexity.

Critical trends in supply chain management across global manufacturing

Introduction to

## Parallel Computing

Second Edition

GEORGE KARYPUS WITH KUMAR

## Параллельные вычисления

Audit. Tax. Consulting. Corporate Finance.

## Функциональное программирование

Парадигма программирования

- Свои языки программирования
- F#, Haskell, LISP, Scheme, Erlang, ML, OCaml, ....

Программа – это функция

- Применяется к входным данным и даёт выходные
- Строится из других (более простых) функций

В функциональном программировании нет

- переменных
- побочных эффектов

## Вкус функционального программирования

```
let double x = x*2
Console.WriteLine(double 2)
```

```
let rec pow_of_2 x n =
  if n<=32 then
    printfn "2**%d = %d" n x
    pow_of_2 (double x) (n+1)</pre>
```

```
let rec rpt n f x =
  if n=0 then x
  else f (rpt (n-1) f x)
```

```
let rec rpt n f =
   if n=0 then fun x->x
   else f >> (rpt (n-1) f)
```

```
let rec для a b f =
    if a<=b then
    f a
    для (a+1) b f
```

```
для 1 32 (fun i-> printfn "2**%d = %d" i (rpt i double 1))
```

```
for i in 1..32 do
  printfn "2**%d = %d" i (rpt i double 1)
```



#### Немного математики

- Множество М точек z ∈ C на комплексной плоскости
- Рассмотрим последовательность
  - $z_{n+1}(c) = z_n^2(c) + c$
  - $z_0(c)=0$

• Множество Мандельброта:

$$M = \{ c \in \mathbf{C} \mid \lim_{n} z_n(c) < \infty \}$$

• Множество, похожее на множество Мандельброта:

$$M' = \{ c \in C \mid |z_{20}(0)| < 1 \}$$

## Реализация F#

```
let mandelf (c:Complex) (z:Complex) = z*z+c
let ismandel c = Complex.Abs(rpt (mandelf c) 20 Complex.zero)<1.0</pre>
```

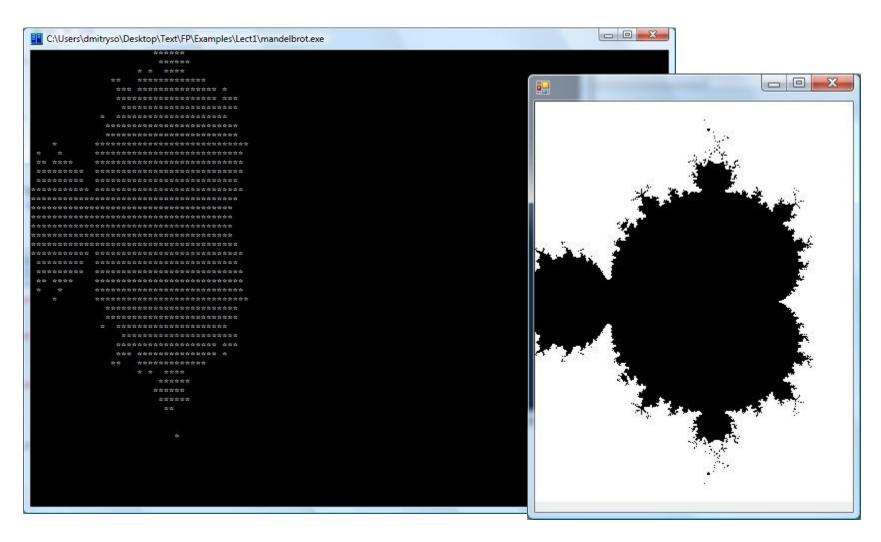
```
let scale (x:float,y:float) (u,v) n = float(n-u)/float(v-u)*(y-x)+x

for i = 1 to 60 do
  for j = 1 to 60 do
   let lscale = scale (-1.2,1.2) (1,60) in
   let t = complex (lscale j) (lscale i) in
     Write(if ismandel t then "*" else " ")
WriteLine("")
```

## WinForms

```
#light
open System.Drawing
open System.Windows.Forms
let form =
   let image = new Bitmap (400, 400)
   let lscale = scale (-1.0, 1.0) (0, 400)
   for i = 0 to (image.Height-1) do
     for j = 0 to (image.Width-1) do
       let t = complex (lscale i) (lscale j) in
       image.SetPixel(i,j,if ismandel t then Color.Black else Color.White)
   let temp = new Form()
   temp.Paint.Add(fun e -> e.Graphics.DrawImage(image, 0, 0))
   temp
[<STAThread>]
do Application.Run(form);;
```

## Результат



### F# - это:

## Мультипарадигмальный язык

- Функционально-императивный
- Сакцентом на функциональном программировании

#### Интероперабельность с .NET

- Все возможности .NET Framework
- Двухсторонняя интероперабельность с пользовательским кодом

#### Компактный код

- Автоматический вывод типов при статической типизации!
- Встроенные структуры данных, своя библиотека обработки

#### Эффективный язык

- Статическая типизация
- Оптимизаторы порождают качественный .NET-код (оптимизация хвостовой рекурсии)
- Ленивые вычисления
- на платформе Microsoft.NET

### Роль F# в экосистеме языков

- F# новаторский язык, в котором воплощаются смелые идеи
  - F# 2.0 асинхронное/параллельное программирование (перекочевало в C# 5.0)
  - F# 3.0 type providers
- F# используется во внутренних проектах
  - Driver Development Kit 0- верификация кода
  - F# написан на F# (догадайтесь, как такое возможно?)
  - Игры XBox Live Arcade
  - AdCenter Challenge
- F# начинает проникать в различные сферы
  - Финансовая аналитика (на смену Excel)
  - Научные вычисления (на смену Matlab/MathCad -> eg. VSLab)
  - Genome Viewer
  - DSL

## Концепция трех экранов













## Функциональное веб-программирование

на платформе Microsoft.NET

## Почему ФП хорошо для веба?

- Веб это про обработку данных
  - Модель запрос-ответ
  - Web is stateless, as are pure functions
- В распределённых веб-системах много асинхронности
  - F# удобен для таких задач
  - Но это выходит за рамки нашего разговора
- F# это вообще круто
  - Меньше кода, больше думать

## Варианты?

#### ASP.NET Web Forms

- Интерфейс на С#, библиотека на F#
- Интерфейс на F#

#### ASP.NET MVC

- Models, Controllers
- Unit Tests

#### Облачные вычисления Windows Azure

- Worker Role
- Web Role

#### WebSharper

- Серверный код
- Клиентский код

## ASP.NET Web Forms на F#

Чтобы это работало, надо соответствующим образом подправить файл конфигурации

## Облачные вычисления

#### Web Role

- Web Interface / Web Service
- ASP.NET WF / MVC
- C#

#### Worker Role

- C# / F#
- Background Computation / Web Service

#### **Azure Storage**

- Прозрачный доступ к очередям / blob
- Модели данных на С# / F# для таблиц

Естественно, можно использовать F# из внешнего проекта в отдельной сборке

## Azure Worker Role

```
type Worker() =
    inherit RoleEntryPoint()
    let log message kind = Trace.WriteLine(message, kind)
    let mutable queue in : CloudQueue = null
    let mutable queue out : CloudQueue = null
    override wr.Run() =
        log "Starting computation agent..." "Information"
        while(true) do
            let msg = queue in.GetMessage()
            if msg=null
                then Thread. Sleep (1000)
                else
                    ProcessMessage(msg.AsString)
                    queue in.DeleteMessage (msg)
            log "Processing queue" "Information"
```

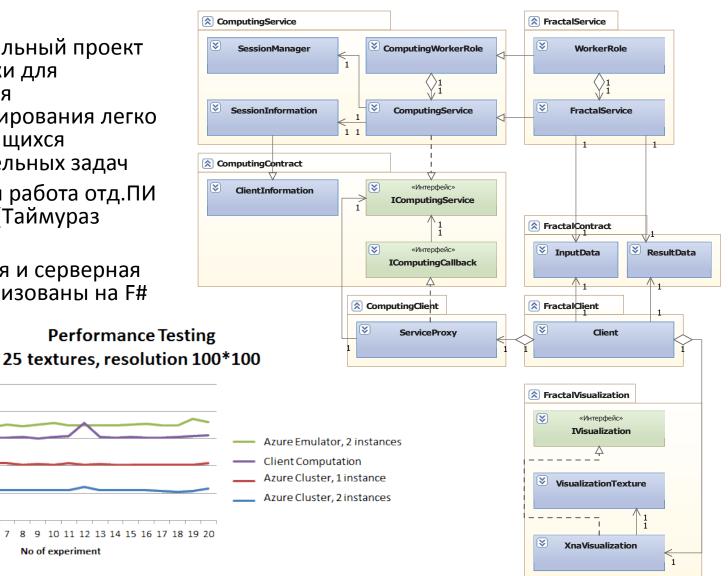
## Проект слабосвязанных облачных вычислений (Parametric Sweep)

- Концептуальный проект библиотеки для облегчения программирования легко параллелящихся вычислительных задач
- Выпускная работа отд.ПИ НИУ ВШЭ (Таймураз Абаев)
- Клиентская и серверная часть реализованы на F#

No of experiment

10

Computation time, sec.



## F# для распределенного программирования

Quotations, Remote Execution, WebSharper

## Quotations

```
let rec power n x =
  if n=0 then 1
  else x*power (n-1) x

let pow5 = power 5
let pow5 x = x*x*x*x*x
```

```
let rec metapower n x =
  if n=0 then <@ 1 @>
  else (fun z -> <@ %x * %z @>)
(metapower(n-1) x)
```

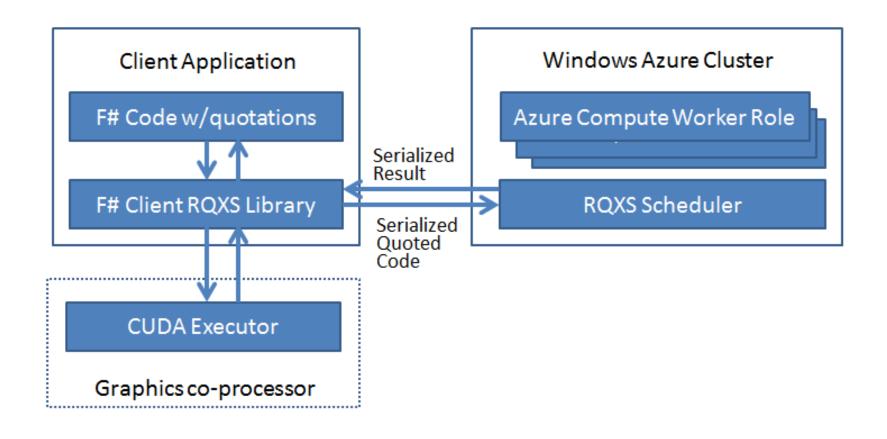
```
let qeval = Microsoft.FSharp.Linq.QuotationEvaluator.Evaluate
let pow5 x = metapower 5 <@ x @> |> qeval
```

## Удалённое выполнение

```
let ex = new RExecutor<int>()
ex.Eval(<@ 1+2+3 @>) |> printfn "%A"
```

```
ex.Eval <@ [1..100] |> List.sum @> |> printfn "%A"
[1;2;3;4]
|> List.map (fun x -> ex.Eval <@ x+1 @>)
|> printfn "%A"
```

## Архитектура библиотеки удалённого выполнения



## WebSharper

- Идея: программировать веб-систему на одном языке
- Разделение кода между клиентским и серверным
- Реализация:
  - Клиентский F#-код транслируется в JavaScript с использованием Quotations
  - Серверный код выполняется на сервере обычным образом
  - Удалённый вызов производится прозначным образом!
- DSL для генерации кода страницы
- http://www.websharper.com

## W# Client Execution

```
[<JavaScriptType>]
type MainPage() =
   inherit Web.Control()
[<JavaScript>]
   override this.Body =
     let result = P [Text ""]
     Div [
        P [Text "Press to retrieve data"]
        Input [Type "Button"; Value "Get Data"]
        |>! OnClick (fun e a -> result.Text <- Application.GetPrimes())
        result ]</pre>
```

## Трансляция в JavaScript



```
(WebSharperProject$.Application$).primes =
function ( arg1)
 var 1;
 if (arg1.$ == 1)
  {
   var t = arg1.$1; var h = arg1.$0;
   var 1 = t;
   var p = function (x) { return x % h > 0; };
    1 =
   new W$.$List({$: 1,$0: h, $1:
((WebSharperProject$.Application$).primes)
                 ((W$.Seq$).toList)(((W$.Seq$).filter)(p, 1)))
                 });}
 else { 1 = new W$.$List({ $: 0 }); }
 return 1; };
```

## W# Server Execution

```
[<JavaScriptType>]
type MainPage() =
   inherit Web.Control()
[<JavaScript>]
   override this.Body =
     let result = P [Text ""]
     Div [
        P [Text "Press to retrieve data"]
        Input [Type "Button"; Value "Get Data"]
        |>! OnClick (fun e a -> result.Text <- Application.GetPrimes())
        result ]</pre>
```

## Абстракции AJAX-интерфейса

## Пейджлеты

• Элемент управления, выполняющийся на клиенте

## Формлеты

• Элемент управления, привязанный к определённой модели данных

## Флоулеты

• Определение последовательности смены формлетов

## W# Formlets



## Книга

«Функциональное программирование на F#»

Издательство ДМК Пресс

## На правах рекламы

http://www.soshnikov.com/fsharp



### Источники

- Д. Сошников. **Функциональное программирование на F#.** ДМК-Пресс, 2011.
- К. Смит. Программирование на F#. М.: Символ-Плюс, 2011. (перевод С. Smith, Programming F#: A comprehensive guide for writing simple code to solve complex problems. O'Reilly, 2010)
- Курс Д.Сошникова «Функциональное программирование» на intuit.ru
- R.Pickering, Beginning F#, Apress, 2009.
- D.Syme, A.Granicz, A.Cisternio. Expert F# 2.0. A-Press, 2010
- T.Neward, A.Erickson, T.Crowell, R.Minerich. Professional F# 2.0. Wiley Publishing, 2011.
- J.Harrop, F# for Scientists, Wiley, 2008.

## Полезнейшие веб-ресурсы:

- http://tryfsharp.org вы можете попробовать F# прямо в браузере!
- http://fssnip.net множество коротких примеров кода на F# на все случаи жизни
- http://www.soshnikov.com/fsharp множество примеров кода, последовательно раскрывающих материал
- http://www.ms-library.ru библиотека учебных курсов Майкрософт



Поиск MSDN с помощью Bing



Россия (Русский) **Dmitry Soshnikov** 

Обучение Мероприятия Сообщество Поддержка Библиотека Скачать Форумы О проекте Главная

#### Библиотека учебных курсов Microsoft Учебные курсы, лекции и семинары – для преподавателей, студентов и самообразования: всё в одном месте на mslibrary.ru!

Данный сайт адресован преподавателям, аспирантам, студентам и энтузиастам, преподающим и изучающим современные информационные технологии. Представлен каталог русскоязычных ресурсов, в котором собраны и классифицированы учебные курсы и методические материалы по преподаванию информационных технологий, разработанные в университетах и учебных центрах, а также обучающие материалы и книги для широкого круга разработчиков и ИТ-специалистов. Данные материалы, размещенные в международном ресурсном центре Faculty Resource Center, могут бесплатно использоваться как для самостоятельного изучения, так и для постановки и внедрения собственных курсов при указании ссылки на автора исходных курсов (см. Microsoft Curriculum License Agreement).

Став постоянным посетителем Faculty Resource Center и нашего каталога, вы получите доступ к актуальным материалам на русском, английском и других языках, а также сможете представить на этом сайте свои учебные и методические материалы.

- Основы программирования
- Объектно-ориентированное программирование
- Языки программирования
- Теория программирования
- Функциональное программирование
- Технология разработки программных систем
- Компьютерная графика
- Разработка игр
- Веб-разработка
- Управление информацией и базы данных
- Офисцов программирования

- Высокопроизводительные вычисления
- Информационная безопасность
- Компьютерные сети
- Автоматное программирование
- В помощь администратору MSDN AA
- Лабораторные работы
- Методические материалы курсов повышения квалификации преподавательского состава
- Операционные системы
- Открытые лекции Microsoft
- Статьи
- Упобино курси сории "Партнорство в



#### Новые поступления

- Решения Microsoft для виртуализации ИТ инфраструктуры предприятий
- Разработка бизнес-приложений для облака с использованием Visual Studio LightSwitch 2011
- Microsoft Visual Studio LightSwitch 2011. Первое знакомство
- Интеллектуальный анализ данных средствами MS SQL Server 2008
- Архитектура, возможности и методы использования Microsoft Windows Azure

Технологии и продукты

Visual Studio NET Azure

## Вопросы



- Дмитрий Сошников
  - Технологический евангелист, координатор академических программ
  - dmitryso@microsoft.com
  - http://blogs.msdn.com/sos
  - http://twitter.com/shwars