

Обзор курса

Олег Смирнов
oleg.smirnov@gmail.com

УНК "ИПСА" НТУУ "КПИ"

16 сентября 2011 г.

Программа на сегодня

- Вводная лекция и обзор курса
- Первая лекция – язык F#; порядок применения функции

Немного об авторе

- 1999-2005: Факультет прикладной математики КПИ
- 2003-2005: x86 Asm, C; reverse engineering, системное программирование
- 2005-2009: C++, Perl, Python; распределённые системы, сетевые протоколы
- 2009-2010: Системы поддержки операций; курс алгоритмов в Global Logic
- 2011-сейчас: Erlang, Ocaml; высоконагруженные системы, социальные сервисы

Языки программирования

- Java
 - C/C++
 - Python
 - Ruby
 - JavaScript
 - ...?
- Lisp / Scheme
 - SML / Ocaml / F#
 - Scala
 - Haskell
 - Eiffel

“Функциональщина” глазами обычного программиста

```
q a b c = putStrLn $ b ++ [toEnum 10, 'q', '('] ++ show b  
        ++ [','] ++ show c ++ [','] ++ show a ++ [')']
```

```
main = q "q a b c = putStrLn $ b ++ [toEnum 10, 'q', '(']  
        ++ show b ++ [','] ++ show c ++ [','] ++ show a ++ [')]"  
"def q(a,b,c): print b+chr(10)+'q'+repr(b)+','+repr(c)+  
"','+repr(a)+')' def e(x) return 34.chr+x+34.chr end;"  
"def q(a,b,c) print b+10.chr+'main=q'+e(b)+''+e(c)+''  
"+e(a)+''+10.chr end"
```

“Функциональщина” глазами обычного программиста



Цели курса

- 1 Практика функционального программирования
- 2 Изучение новой парадигмы
- 3 Приёмы функционального программирования

Практика функционального программирования



Цели курса

- 1 Практика функционального программирования
- 2 Изучение новой парадигмы
- 3 Приёмы функционального программирования

Изучение новой парадигмы

```
public int fact(int n)
{
    int f = 1;
    for (int i = 2; i <= n; i++)
    {
        f = f * i;
    }
    return f;
}
```

```
let rec fact n =
    match n with
    | 0 -> 1
    | _ -> n * fact (n - 1);;
```

Изучение новой парадигмы

- **Неизменяемое состояние** – меньше ошибок, чище код
- Верификация – доказываем программы как теоремы
- Параллелизм – технология Google MapReduce и Apache Hadoop

Изучение новой парадигмы

- Неизменяемое состояние – меньше ошибок, чище код
- Верификация – доказываем программы как теоремы
- Параллелизм – технология Google MapReduce и Apache Hadoop

Изучение новой парадигмы

- Неизменяемое состояние – меньше ошибок, чище код
- Верификация – доказываем программы как теоремы
- Параллелизм – технология Google MapReduce и Apache Hadoop



Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6 (OSDI'04)*, Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10.



Mike Cafarella and Doug Cutting. 2004. Building Nutch: Open Source Search. *Queue* 2, 2 (April 2004), 54-61.

Цели курса

- 1 Практика функционального программирования
- 2 Изучение новой парадигмы
- 3 Приёмы функционального программирования

Приёмы функционального программирования

- Сборка мусора: появилась в LISP (1958), стала мейнстримом в 1995 (Java)
- Замыкания: появились в Scheme (1975), стали мейнстримом в 2000-х (C#, JavaScript)
- Ленивая обработка: появилась в Miranda (1985), стала мейнстримом в 2000-е (LINQ)

Приёмы функционального программирования

- Сборка мусора: появилась в LISP (1958), стала мейнстримом в 1995 (Java)
- Замыкания: появились в Scheme (1975), стали мейнстримом в 2000-х (C#, JavaScript)
- Ленивая обработка: появилась в Miranda (1985), стала мейнстримом в 2000-е (LINQ)

Приёмы функционального программирования

- Сборка мусора: появилась в LISP (1958), стала мейнстримом в 1995 (Java)
- Замыкания: появились в Scheme (1975), стали мейнстримом в 2000-х (C#, JavaScript)
- Ленивая обработка: появилась в Miranda (1985), стала мейнстримом в 2000-е (LINQ)

Приёмы функционального программирования

- Параметрический полиморфизм: появился в 1983, сейчас мейнстрим в Java, C#, C++
- Функции высшего порядка: появились в LISP, сейчас мейнстрим везде, кроме Java
- Вывод типов: появился в ML (1979), стал мейнстримом в 2007 (C#, Scala, C++0x)

Приёмы функционального программирования

- Параметрический полиморфизм: появился в 1983, сейчас мейнстрим в Java, C#, C++
- Функции высшего порядка: появились в LISP, сейчас мейнстрим везде, кроме Java
- Вывод типов: появился в ML (1979), стал мейнстримом в 2007 (C#, Scala, C++0x)

Приёмы функционального программирования

- Параметрический полиморфизм: появился в 1983, сейчас мейнстрим в Java, C#, C++
- Функции высшего порядка: появились в LISP, сейчас мейнстрим везде, кроме Java
- Вывод типов: появился в ML (1979), стал мейнстримом в 2007 (C#, Scala, C++0x)

Программа курса: первый семестр

- Язык F#
- Азы: рекурсия, функции высших порядков, замыкания, алгебраические типы данных
- Практика: простые алгоритмы и структуры данных, парсинг
- Параллельное программирование: свёртки, пробеги и технология MapReduce

Программа курса: первый семестр

- Язык F#
- Азы: рекурсия, функции высших порядков, замыкания, алгебраические типы данных
- Практика: простые алгоритмы и структуры данных, парсинг
- Параллельное программирование: свёртки, пробеги и технология MapReduce

Программа курса: первый семестр

- Язык F#
- Азы: рекурсия, функции высших порядков, замыкания, алгебраические типы данных
- Практика: простые алгоритмы и структуры данных, парсинг
- Параллельное программирование: свёртки, пробеги и технология MapReduce

Программа курса: первый семестр

- Язык F#
- Азы: рекурсия, функции высших порядков, замыкания, алгебраические типы данных
- Практика: простые алгоритмы и структуры данных, парсинг
- Параллельное программирование: свёртки, пробеги и технология MapReduce

Программа курса: второй семестр

- Язык Haskell
- Чисто функциональные структуры данных и персистентность
- Программирование с монадами
- Классы типов
- Математические основы ФП и начало теории типов

Программа курса: второй семестр

- Язык Haskell
- Чисто функциональные структуры данных и персистентность
- Программирование с монадами
- Классы типов
- Математические основы ФП и начало теории типов

Программа курса: второй семестр

- Язык Haskell
- Чисто функциональные структуры данных и персистентность
- Программирование с монадами
- Классы типов
- Математические основы ФП и начало теории типов

Программа курса: второй семестр

- Язык Haskell
- Чисто функциональные структуры данных и персистентность
- Программирование с монадами
- Классы типов
- Математические основы ФП и начало теории типов

Программа курса: второй семестр

- Язык Haskell
- Чисто функциональные структуры данных и персистентность
- Программирование с монадами
- Классы типов
- Математические основы ФП и начало теории типов

Пререквизиты курса

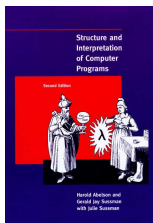
- Программирование и алгоритмические языки
- Дискретная математика
- Математическая логика и теория алгоритмов

- Два семестра по шестнадцать лекций
- Восемь лабораторных, две контрольные и два экзамена
- Лекции по пятницам на пятой паре (16:10) в аудитории 02-13
- Два лектора плюс гостевые лекции
- Программа курса: <http://goo.gl/RnbdH>

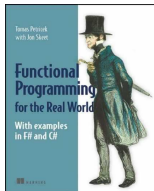
- Два семестра по шестнадцать лекций
- Восемь лабораторных, две контрольные и два экзамена
- Лекции по пятницам на пятой паре (16:10) в аудитории 02-13
- Два лектора плюс гостевые лекции
- Программа курса: <http://goo.gl/RnbdH>

- Два семестра по шестнадцать лекций
- Восемь лабораторных, две контрольные и два экзамена
- Лекции по пятницам на пятой паре (16:10) в аудитории 02-13
- Два лектора плюс гостевые лекции
- Программа курса: <http://goo.gl/RnbdH>

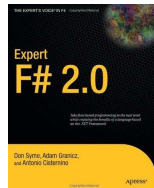
- Два семестра по шестнадцать лекций
- Восемь лабораторных, две контрольные и два экзамена
- Лекции по пятницам на пятой паре (16:10) в аудитории 02-13
- Два лектора плюс гостевые лекции
- Программа курса: <http://goo.gl/RnbdH>



Харольд Абельсон, Джеральд Джей Сассман “Структура и интерпретация компьютерных программ”



Томаш Петричек, Джон Скит
“Functional Programming for the Real World”



Дон Сайм “Expert F# 2.0”

Рекомендуемая литература



Журнал “Практика функционального программирования”, <http://fprog.ru>

Вопросы?