

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ НА SCALA

Шабалин Денис, LAMP/EPFL

ОБО МНЕ

- Закончил ЛИТ в 2008
- Бакалавр НаУКМА в 2012
- Магистр EPFL в 2014
- На данный момент аспирант (PhD student) в EPFL

ОБО МНЕ

Пять лет занимаюсь исследованиями вокруг языка программирования Scala.



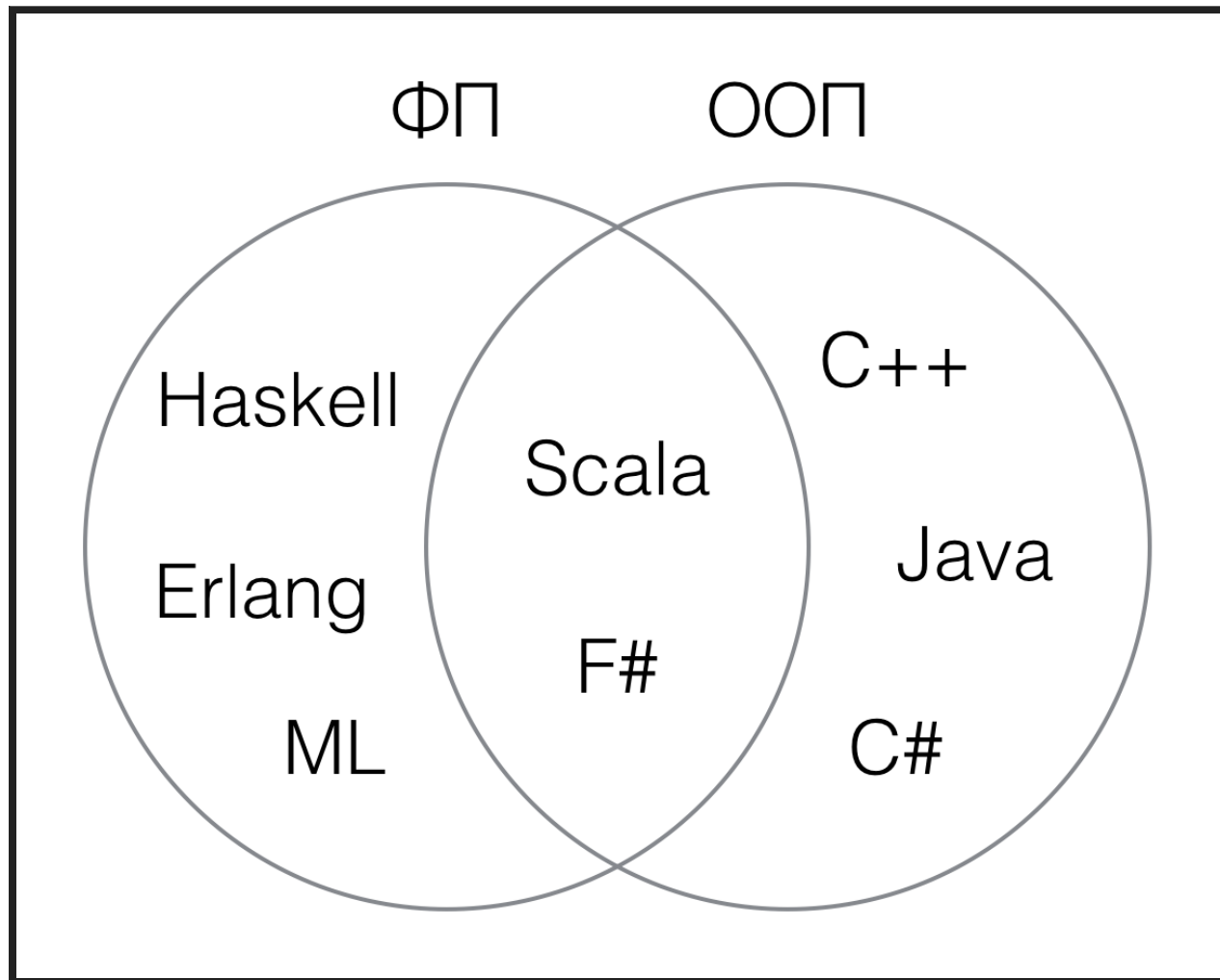
СЕГОДНЯ

- Что такое функциональное программирование?
- Что такое Scala и для чего её используют?

ПАРАДИГМЫ

1. Структурное программирование
2. Объектно-ориентированное программирование (ООП)
3. Функциональное программирование (ФП)

ЯЗЫКИ



ООП

- Инкапсуляция
- Наследование
- Полиморфизм

ФП

- Неизменяемые структуры данных
- Строгая типизация с выводом типов
- Сопоставление с образцом
- Функции высших порядков и рекурсия

НЕИЗМЕНЯЕМЫЕ СТРУКТУРЫ ДАНЫХ

```
scala> abstract class List[+T]  
      | case class Cons[T](head: T, tail: List[T]) extends List[T]  
      | case object Nil extends List[Nothing]  
  
scala> val lst = Cons(1, Cons(2, Cons(3, Cons(4, Nil))))  
lst: Cons[Int] = Cons(1,Cons(2,Cons(3,Cons(4,Nil))))
```

СОПОСТАВЛЕНИЕ С ОБРАЗЦОМ

```
scala> def count(lst: List[Int]): Int =  
  |   lst match {  
  |     case Nil           => 0  
  |     case Cons(head, tail) => 1 + count(tail)  
  |   }  
  
scala> count(lst)  
res1: Int = 4
```

ФУНКЦИИ ВЫСШИХ ПОРЯДКОВ

```
scala> def filter[A](lst: List[A])(f: A => Boolean): List[A] =  
  |   lst match {  
  |     case Nil =>  
  |       Nil  
  |     case Cons(head, tail) if f(head) =>  
  |       Cons(head, filter(tail)(f))  
  |     case Cons(_, tail) =>  
  |       filter(tail)(f)  
  |   }
```

```
scala> filter(lst)(x => x % 2 == 0)  
res5: List[String] = Cons(2, Cons(4, Nil))
```

ВОПРОС: ЗАЧЕМ ЭТО ВСЕ?

Ответ: Распределенные системы.

РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ



SPARK

Простой код и автоматическое распределение:

```
scala> val textFile = sc.textFile("huge-file.txt")
textFile: spark.RDD[String] = spark.MappedRDD@2ee9b6e3

scala> textFile.filter(line => line.contains("Spark")).count()
res3: Long = 15
```

СВОДКА

- ФП — это модная парадигма программирования
- Scala — это гибридный ФП-ООП язык
- ФП хорошо ложится на распределенные системы

ВОПРОСЫ?