

Язык Scala

Олег Галако

Денис Анойкин

История создания

- Создатель: Мартин Одерски
- Швейцария, EPFL, 2001
- Эксперименты: Pizza, Funnel
- Работа над Java
- 1.0 в 2003 г., текущая – 2.10.2



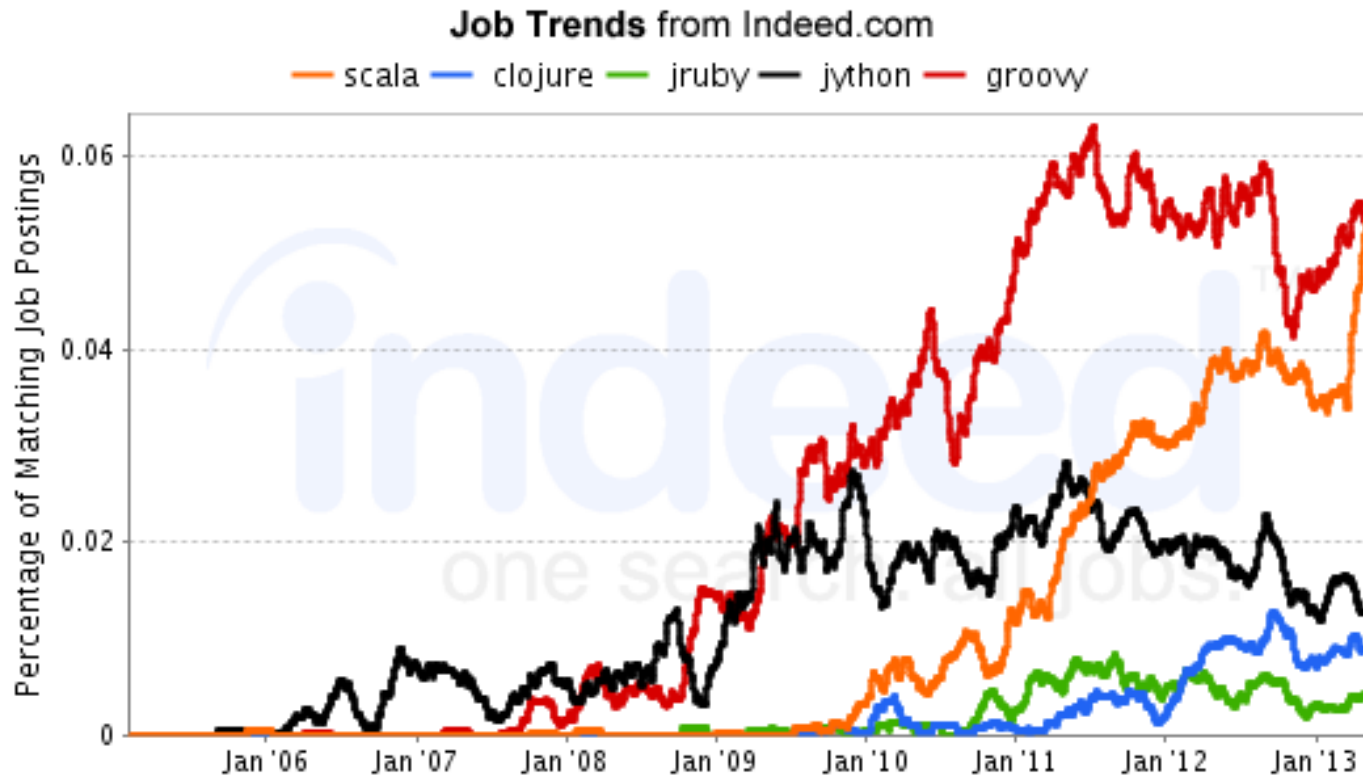
Особенности

- Мощная статическая типизация
- Объектно-ориентированный
- Функциональный
- Основная платформа: JVM
- Параллельные и распределённые вычисления



Позиции языка

- Рост среди альтернативных языков на JVM



Позиции языка

- Рост среди альтернативных языков на JVM



Позиции языка

- Активно используется в крупных проектах
- Библиотеки и фреймворки «взрослеют»
- Крупный исследовательский грант
- Инвестиции: Typesafe

twitter

Linked in

foursquare

Novell

intel

 Typesafe

Развитие языка Java

Было запланировано в JDK 7	Статус
Small language enhancements	Сделано
Project Lambda	Отложено
Support for modular programming	Отложено
Language support for collections (immutable, etc.)	Отложено
Annotations on Java types	Отложено

JDK 8: ~~октябрь 2012~~ ~~сентябрь 2013~~ март 2014

Пример: NullPointerException

// NullPointerException в последней строке

```
Map<String, Integer> myStuff = new HashMap<>();  
myStuff.put("hello", 42);
```

```
int result = myStuff.get("goodbye") * 2;
```

// Корректный вариант:

```
Integer tmp = myStuff.get("goodbye");  
int result = tmp == null ? 0 : tmp * 2;
```


Пример: NullPointerException

```
// Ошибка на стадии компиляции в  
// последней строке, потому что в Map:  
// def get (key: A) : Option[B]
```

```
val myStuff = Map("hello" -> 42)  
val result = myStuff("goodbye") * 2
```

```
// Корректный вариант:
```

```
val result = myStuff("goodbye") .getOrElse(0) * 2
```

Пример: NullPointerException

// Если частые проверки на null, то обертка

```
Map<String, Integer> myStuff = new  
MyPropertiesMap<>();  
myStuff.put("hello", 42);
```

```
int result = myStuff.get("goodbye", 0) * 2;
```

Пример: Lazy

```
// Java
```

```
class LazyCache {  
    private Cache<String, String> bigMap;  
    private void check() {  
        if (bigMap != null) {  
            bigMap = readFromDB(...);  
        }  
    }  
    public String getByName(String str) {  
        check();  
        return bigMap.get(str);  
    }  
  
    public String getLatest (String str) {  
        check();  
        return bigMap.latest();  
    }  
}
```

Пример: Lazy

```
// Scala
```

```
class LazyCache {
```

```
    lazy val bigMap:Cache[String, String] = read(...);
```

```
    def getByName(key:String) = bigMap(key);
```

```
    def getLatest() = bigMap.latest;
```

```
}
```

Пример: Lazy

```
// Java
class LazyCache {
    private Cache<String, String> bigMap;
    protected Cache<String, String> getBigMap() {
        if (bigMap != null) {
            bigMap = readFromDB(...);
        }
        return bigMap();
    }
    public String getByName(String str) {
        return getBigMap().get(str);
    }

    public String getLatest (String str) {
        return getBigMap() ...;
    }
}
```

Пример: Immutable

```
public class LinksCache {  
    private Map<User, List<Group>> groupsNoEnemies = ...  
    private Map<Group, Map<User, List<User>>> enemies = ...  
    private Map<User, Map<Group, List<User>>> friends = ...  
  
    public synchronized void registerFriend (User u, User f) {  
        ...  
    }  
    public synchronized void removeUser (User u) {  
        ...  
    }  
    public synchronized void addUserInGroup(User u, Group g) {  
        ...  
    }  
    public List<Group> getGroupsNoEnemies(User u) {  
        ...  
    }  
}
```

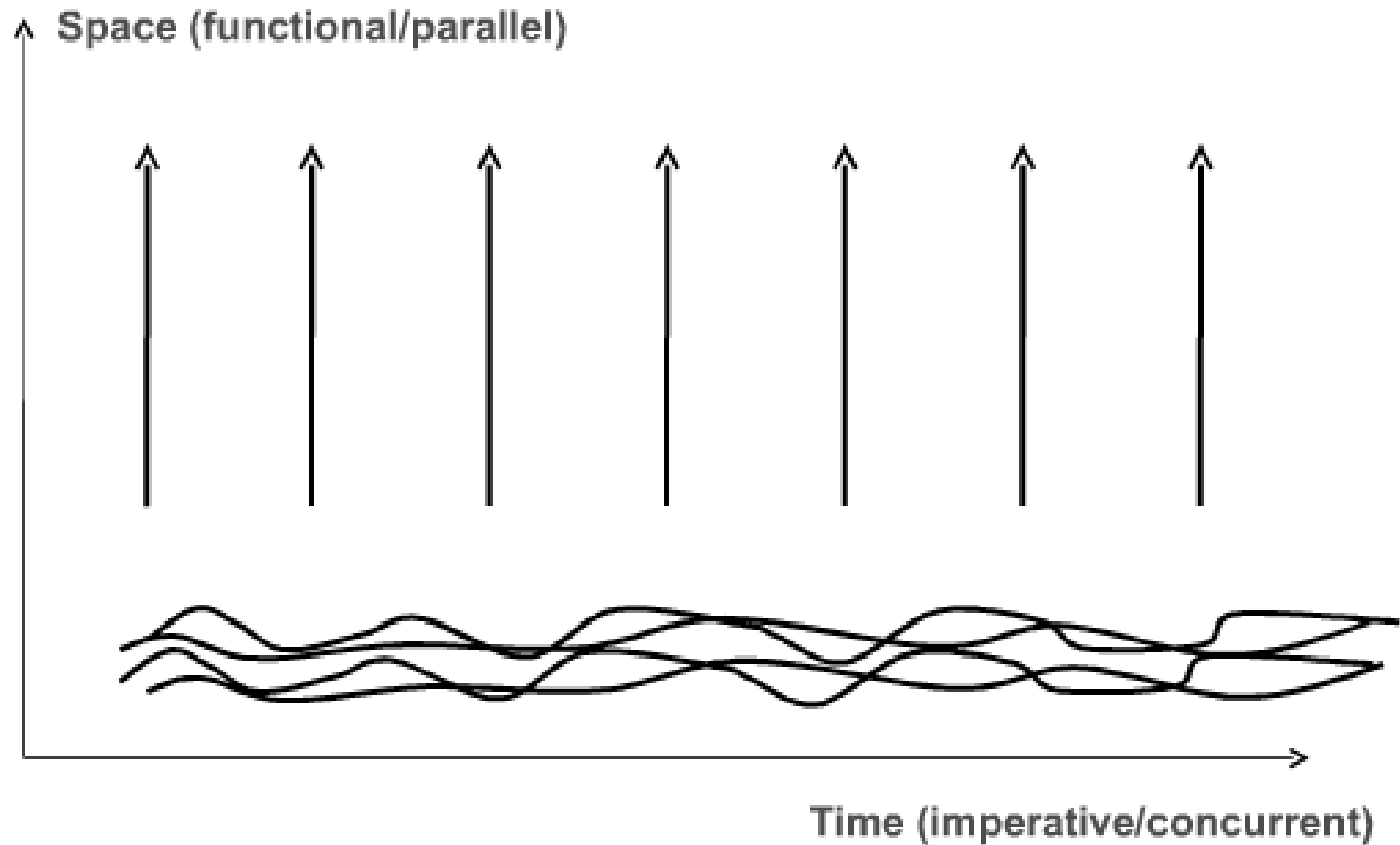
Пример: Immutable

// Scala

НЕ ИСПОЛЬЗУЙТЕ mutable state

```
class Manager {  
    private LinksCache cache = ...  
  
    void onUpdate() {  
        cache = rebuildLinksCache();  
    }  
}  
  
public class LinksCache {  
    private Map<User, List<Group>> groupsNoEnemies = ...  
    private Map<Group, Map<User, List<User>>> enemies = ...  
    private Map<User, Map<Group, List<User>>> friends = ...  
  
    public List<Group> getGroupsNoEnemies(User u) {  
        ...  
    }  
}
```

Пример: Immutable



Переходим на Scala?

Scala – альтернатива языку Java при разработки сложных высоконагруженных систем на Java платформе

FourSquare: ... heavy duty web server application had successfully ported to the Scala/Lift web framework. He explained that the conversion took a surprisingly short time, just 3 months.

Électricité de France (EDF) is the largest French energy company ... replaced a substantial part of their 300,000 lines of Java code ... has seen a significant increase in development productivity

Scala: Грант € 2,300,000 от Европейского исследовательского совета в рамках конкурса «Popular Parallel Programming»

Переходим на Scala?

Scala – **функциональный язык**, но поддерживающий **строгую типизацию**, что дает ему преимущество перед динамическими языками

Bruce Eckel: The highest complement you can deliver in the Python world is to say that something is "Pythonic" - that it feels and fits into the Python way of thinking. I never imagined that a static language could feel this way, but Scala does -- and possibly even better.

Robey Pointer at Twitter has moved their main message queue from Ruby to Scala for improved performance.

Переходим на Scala?

Scala – **новый язык**, вобравший в себя все последние наработки ООП и функционального программирования. На нем просто **интересно и приятно** программировать.

Bruce Eckel: Scala does suffer from the *mistaken idea that it's complicated*, and for good reason. Many early adopters have been language enthusiasts who love to show how clever they are, and this only confuses beginners. But you can see ... that learning Scala should be a lot easier than learning Java!