

Introduction to Scala Programming Language

Barcelona Developers Conference 2013 (#bcd13)
Scala Developers Barcelona (#scalabcn)

Jordi Pradel (@agile_jordi)
Ignasi Marimon-Clos (@ignasi35)



thanks!

about me

n. /iŋ'nazi/ - @ignasi35

1) ~~problem solver, Garbage Collector, mostly~~
java, learning scala, some agile

2) kayaker

3) under construction

4) wears glasses

about me

jordi pradel - @agile_jordi

- 1) agile software developer
- 2) agile punk
- 3) scala practitioner
- 4) ~~glasses~~, braid

about you



Some Context

What they say

“If I were to pick a language today other than Java it would be Scala”

James Gosling (created Java)

“I can honestly say if someone had shown me the Programming Scala book by Martin Odersky, Lex Spoon & Bill Venners back in 2003 I’d probably have never created Groovy”

James Strachan (created Groovy)

What they say

“What’s next?
LISP”



Paolo Perrotta at Baruco 2013

<http://www.youtube.com/watch?v=ahh-QkttjuM&t=115>



Job Trends

Job Trends

[Job Postings Per Capita](#)

[Job Market Competition](#)

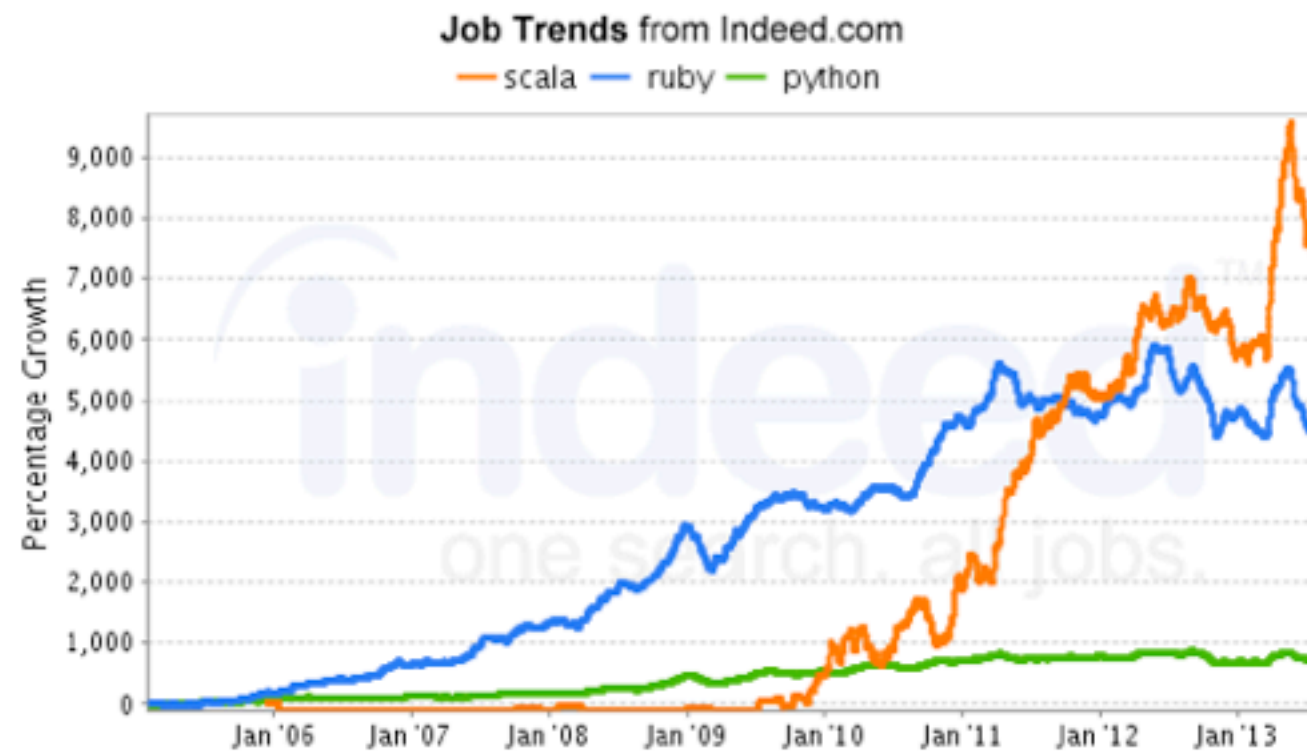
[Industry Employment Trends](#)

scala, ruby, python Job Trends

Scale: [Absolute](#) - [Relative](#)

► [Email to a friend](#)

► [Post on your blog/website](#)



Top Job Trends

1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows relative growth for jobs we find matching your search terms.

Find [Scala jobs](#), [Ruby jobs](#), [Python jobs](#)

Feel free to ► [share this graph](#)

For jobs in Spain, visit [Indeed Spain](#)

[Jobs](#) - [Salaries](#) - [Trends](#) - [Forums](#) - [Browse Jobs](#) - [Tools](#) - [API](#) - [About](#) - [Contact](#)

©2013 Indeed - [Cookies Use](#), [Privacy](#) and [Terms](#)

<http://www.indeed.com/jobtrends?q=scala%2C+ruby%2C+python&l=&relative=l>



Scala Jobs

[Home](#)[Find IT Jobs](#)[Directory](#)[About](#)[Permanent IT Jobs](#)[Contract IT Jobs](#)

+1

0



Tweet

33



Like

11

The IT Job Board

Your search made simple

Scala

Job Type - Any

Date Posted - All

[Search Jobs](#)[Email me jobs like this](#)

Job Title or Keywords



The first table below looks at the demand for **Scala** skills in IT jobs advertised across the UK. Included is a guide to the average salaries offered in IT jobs that have cited Scala over the 3 months to 1 November 2013 with a comparison to the same period in the previous 2 years. The second table is for comparison and provides aggregates for all of the **Programming Languages** category.

Location	UK	3 months to 1 Nov 2013	Same period 2012	Same period 2011
Scala ⓘ				
Rank		381	526	547
Rank change year-on-year		▲ +145	▲ +21	
Permanent jobs citing Scala		707	410	400
As % of all permanent IT jobs located in the UK		0.65%	0.36%	0.31%
As % of the Programming Languages category		1.21%	0.68%	0.59%
Number of salaries quoted		603	349	379
Average salary		£62,500	£60,000	£55,000
Average salary % change year-on-year		+4.16%	+9.09%	
90% offered a salary of more than		£42,750	£44,750	£40,000
10% offered a salary of more than		£100,000	£85,250	£80,000

[Software Eng](#)

Slough, Thame

Telefonica

Salary: Compe

Posted: 3 days

[Senior Java](#)

London/Berli

London

bridgenoble

Salary: 55000

+ benefits

Posted: 3 days

[Java Develop](#)

Docklands, Lo

harringtonstarr

Salary: £60000

Benefits + Bonu

Posted: 5 days

[Java Develop](#)[Javascript](#)

Warwickshire,

Hudson

Salary: £30000

Bonus + Benefi

Posted: 16 day

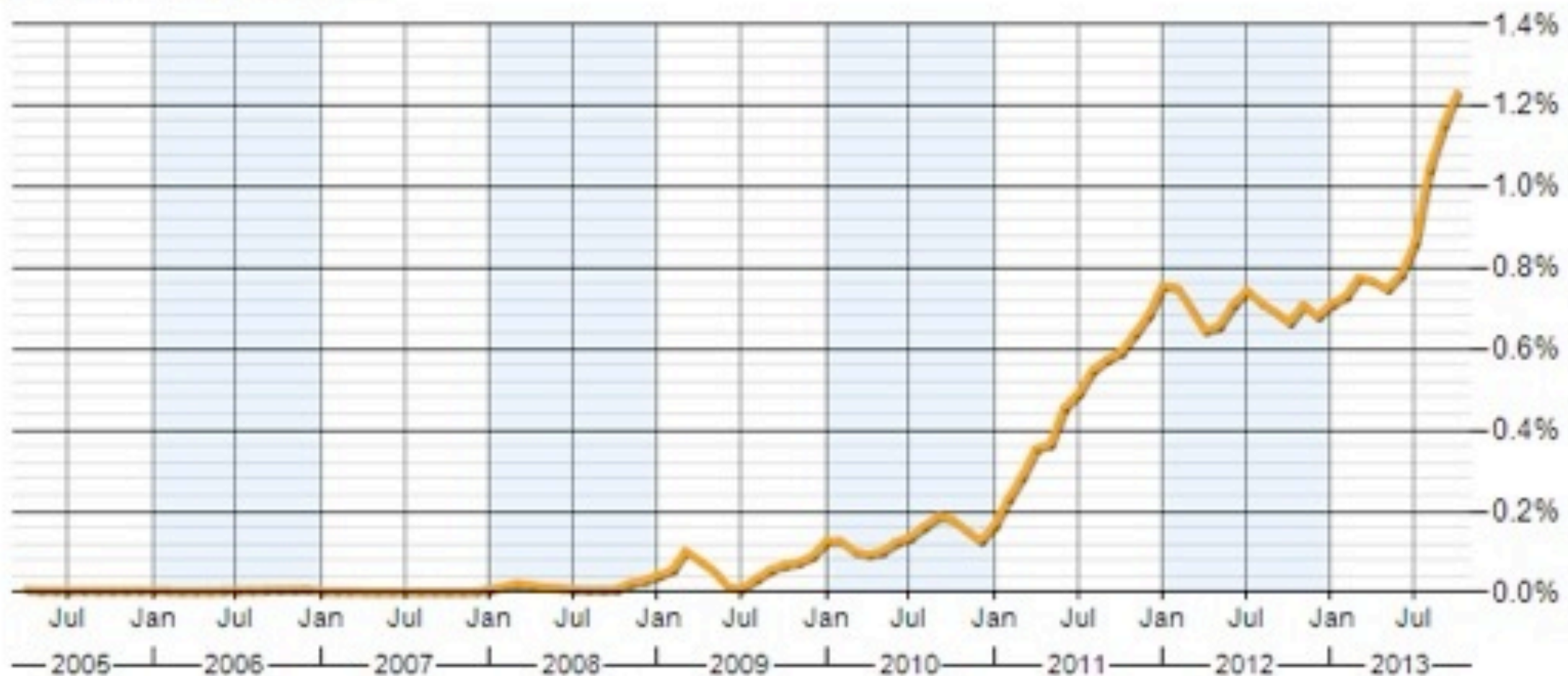
[Java Team Le](#)

<http://www.itjobswatch.co.uk/jobs/uk/scala.do>

As % of all permanent IT jobs located in the UK	53.59%	53.79%	53.33%
Number of salaries quoted	50360	52927	59941
Average salary	£42,500	£42,000	£42,000
Average salary % change year-on-year	+1.19%	-	
90% offered a salary of more than	£28,000	£27,500	£27,500
10% offered a salary of more than	£65,000	£67,500	£65,000
UK excluding London average salary	£37,500	£37,500	£37,500

Scala Jobs Demand Trend

This chart provides the 3-month moving total of permanent IT jobs citing Scala within the UK as a proportion of the total demand within the **Programming Languages** category.



<http://www.itjobswatch.co.uk/jobs/uk/scala.do>

open s
£50k +
Cambri
ECM S
Salary:
Posted:
Senior
London
LShift
Salary:
Posted:
JAVA D
City of
Hays S
Salary:
annum
Posted:
Java D
Agile
City of
Hays S
Salary:
Posted:
Java S
City of
Huxley
Salary:
Posted:

Start with why

A Problem:

- A service to process a Basket Purchase
 - buy items online
- Pay via VISA
- Pay via PayPal

This is your (jee) life

```
public class PurchaseProcessorFactory {  
  
    public PurchaseProcessor forPayPal() {  
        return new PurchaseProcessor() {  
            @Override  
            public void process(Purchase t) {  
                // process t using PayPal system  
            }  
        };  
    }  
  
    public PurchaseProcessor forVisa() {  
        return new PurchaseProcessor() {  
            @Override  
            public void process(Purchase t) {  
                // process t using Visa  
            }  
        };  
    }  
}
```



This is your (jee) life

```
package com.meetup.java;
```

```
public class Item {
```

```
    private final String _productId;  
    private final int _rupees;
```

```
    public Item(String _productId, int _rupees) {  
        super();  
        this._productId = _productId;  
        this._rupees = _rupees;  
    }
```

```
    @Override
```

```
    public String toString() {  
        return "Item [_productId=" + _productId + ", _rupees=" + _rupees + "];"  
    }
```

```
    @Override
```

```
    public int hashCode() {  
        final int prime = 31;  
        int result = 1;  
        result = prime * result  
            + ((_productId == null) ? 0 : _productId.hashCode());  
        result = prime * result + _rupees;  
        return result;  
    }
```

```
    @Override
```

```
    public boolean equals(Object obj) {  
        if (this == obj)  
            return true;  
        if (obj == null)  
            return false;  
        if (getClass() != obj.getClass())  
            return false;  
        Item other = (Item) obj;  
        if (_productId == null) {  
            if (other._productId != null)  
                return false;  
        } else if (!_productId.equals(other._productId))  
            return false;  
        if (_rupees != other._rupees)  
            return false;  
        return true;  
    }
```

```
package com.meetup.java;
```

```
import java.util.Collections;  
import java.util.List;
```

```
public class Purchase {
```

```
    private final List<Item> _items;
```

```
    public Purchase(List<Item> items) {  
        super();  
        _items = items;  
    }
```

```
    public List<Item> getItems() {
```

```
        return Collections.unmodifiableList(_items);  
    }
```

```
    @Override
```

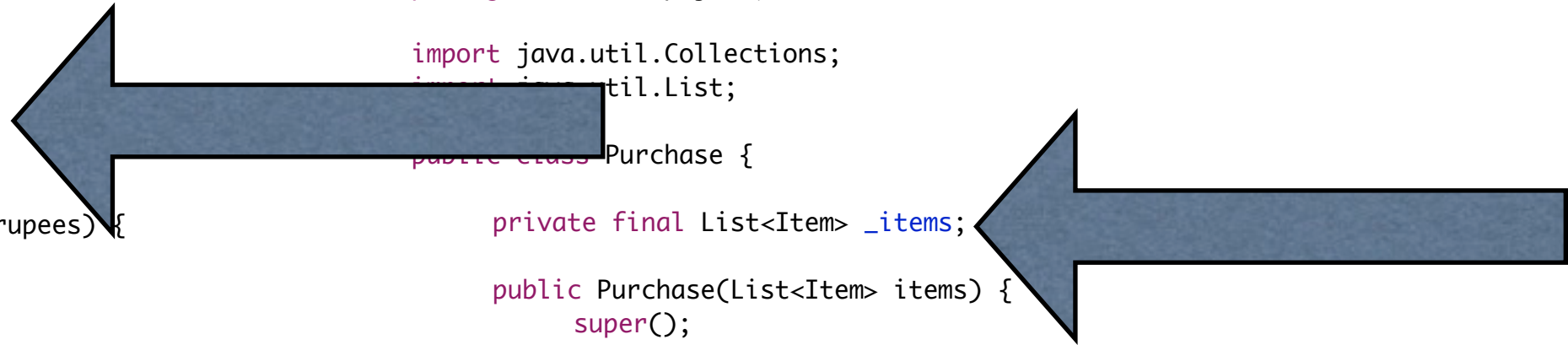
```
    public String toString() {  
        return "Purchase [_items=" + _items + "];"  
    }
```

```
    @Override
```

```
    public int hashCode() {  
        final int prime = 31;  
        int result = 1;  
        result = prime * result + ((_items == null) ? 0 : _items.hashCode());  
        return result;  
    }
```

```
    @Override
```

```
    public boolean equals(Object obj) {  
        if (this == obj)  
            return true;  
        if (obj == null)  
            return false;  
        if (getClass() != obj.getClass())  
            return false;  
        Purchase other = (Purchase) obj;  
        if (_items == null) {  
            if (other._items != null)  
                return false;  
        } else if (!_items.equals(other._items))  
            return false;  
        return true;  
    }
```



What's wrong there?

- class per file (sort of)
- unnecessary redefinition of stuff
- duplicate intent
 - (DRY is not only about copy/paste)
- unmaintainability
- JAVA

OH: “If a Notary could write code, she’d use JAVA”

Let's redo that

```
package com.meetup.scala

case class Item(productId : String, rupees : Int)
case class Purchase(items : List[Item])

trait Processor[T] {
  def process(t : T) : Unit
}
trait PurchaseProcessor extends Processor[Purchase]

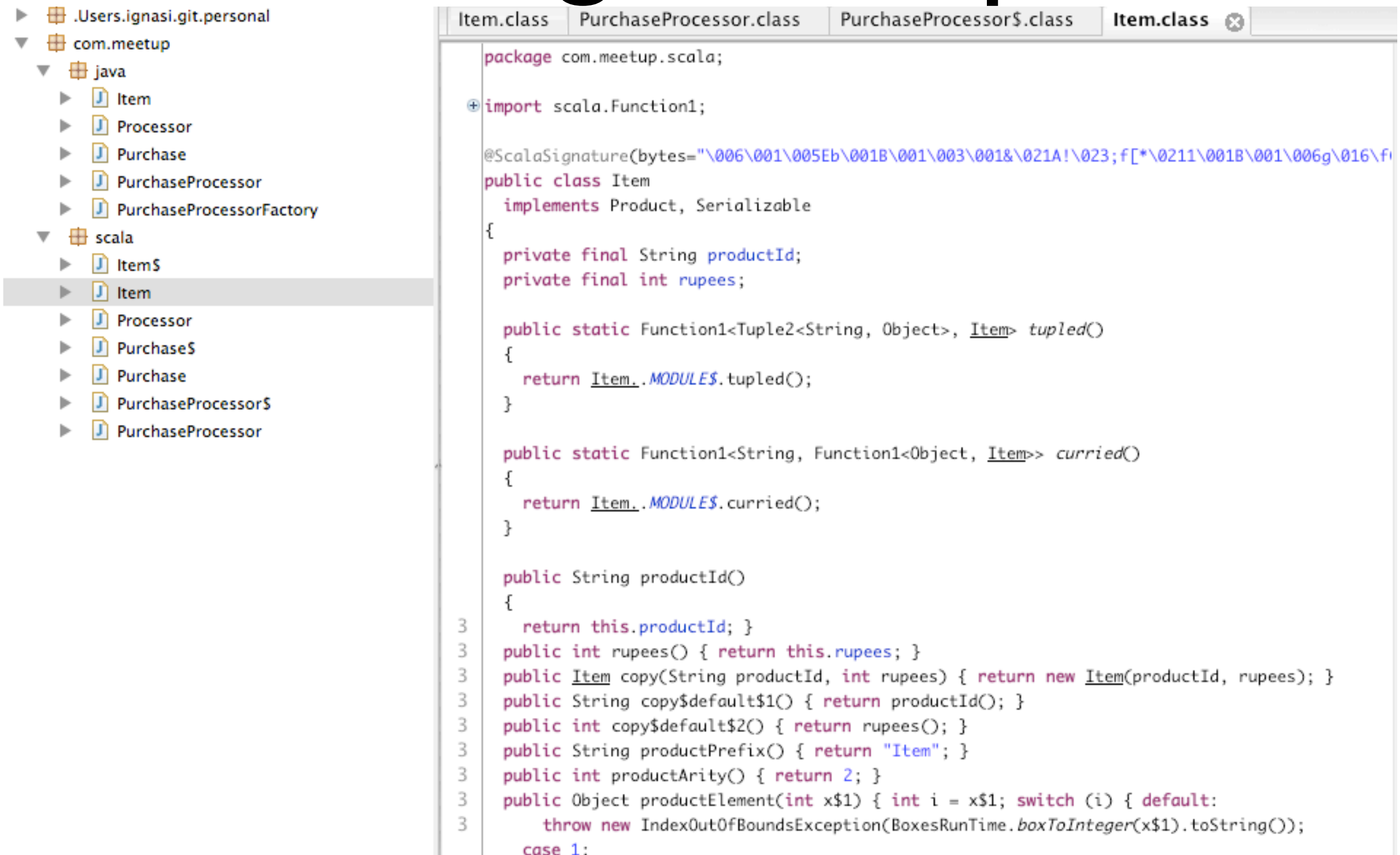
object PurchaseProcessor {

  val forPayPal = new PurchaseProcessor {
    def process(p : Purchase) = println(p) // do real stuff here
  }
  val forVisa = new PurchaseProcessor {
    def process(p : Purchase) = println(p) // do real stuff here
  }

}
```

fits in one slide (and fixes one bug!)

Entering decompiler



case classes

```
case class Item(productId : String, rupees : Int)
case class Purchase(items : List[Item])
```

- all boilerplate gone
- no equals/hashCode/toString maintenance
- immutable
- named params
- optional params (not shown)
- pattern matching

pater-WAT ??

Pattern Matching

```
def run(input : Any) = {  
  input match {  
    case s : String    => println(s)  
    case i : Int if(i<0) => println("A negative int: " + i)  
    case i : Int        => println("A number: " + i)  
    case _              => println("something else") // aka default:  
  }  
}
```

> run: (input: Any)Unit

```
run("A long time ago in a galaxy far far away...")
```

> A long time ago
> in a gala...

```
run(234)  
run() => 42)
```

> A number: 234
> something else

Worksheet rocks! REPL rocks!

Functions!

```
run(()) => 42  
val f = () => 42  
run(f)
```

- Functions as first class citizens
 - Functional Programming! Bitches!
- parens + arrow + statements
- 'return' keyword is optional
 - SOLUTION: A function with no params that simply returns '42'

Functional Programming

- referential transparency
- immutability (see case classes)
- no side effects
- separate data from behavior
- monads *

* We don't know what monads are but a talk about
FP requires using the word monad

Back to coding!

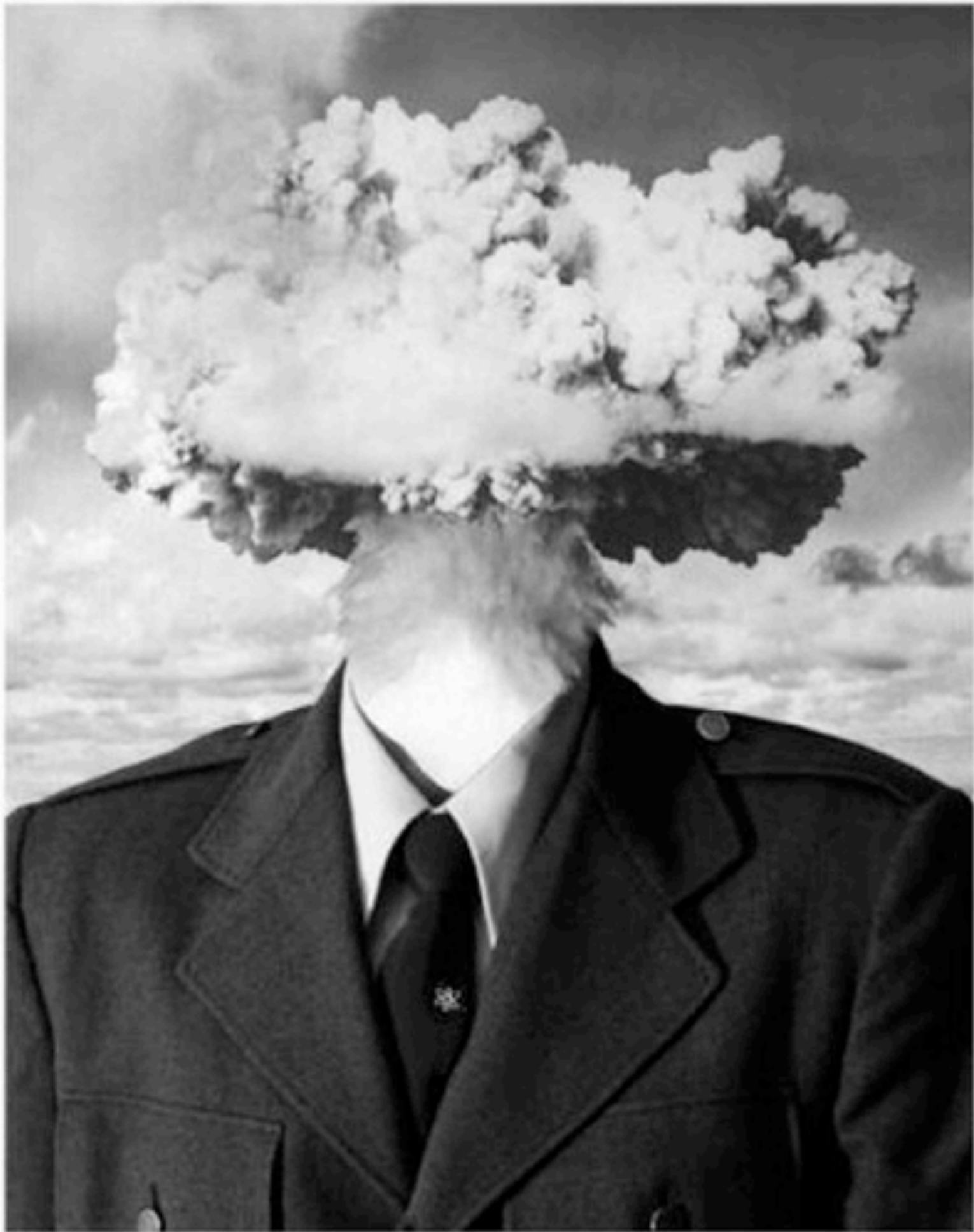
Syntactic Sugar

- Dot is replaceable with whitespace
- Parens are optional and usually replaceable with curly braces
- Functions can be declared inline
- `return` and semicolon are optional
- `_`

```
l map { twice andThen thrice }  
l.map((i:Int) => twice.andThen(thrice)(i))
```

so far

- conciseness
- case classes
- traits
- *multi-hierarchy*
- pattern matching
- Functions
- immutability
- monads *
- spaces/parens/dots
- return keyword



conclusions

- A lot of syntactic sugar
- Many small features but all related
- functional!
- object oriented (unlike clojure/erlang/...)
- typesafe without the pain
- Classes, Objects, Companions, Traits...

Yak shaving

- Any seemingly pointless activity which is actually necessary to solve a problem which solves a problem which, several levels of recursion later, solves the real problem you're working on.

<http://www.urbandictionary.com/define.php?term=yak%20shaving>

other stuff

- Macros
- Currying, partially applied functions, ...
- Structural Typing (see? we `_do_` have duck typing!)
- implicits
- Value classes
- Lazy evaluation
- Tail recursion
- ...

want more?

- Intro to scala: <http://scalacamp.pl/intro/#/start>
- Twitter's Scala School: http://twitter.github.io/scala_school/
- 99 problems in scala: <http://aperiodic.net/phil/scala/s-99/>
- Scala Puzzles: <http://scalapuzzlers.com/>
- “Programming in Scala” (1st ed free online: <http://www.artima.com/pins1ed/>)
- **Scala Developers Barcelona Meetup**
<http://www.meetup.com/Scala-Developers-Barcelona/>



thanks!

Oh! And we know what monads are. ;-)