Practical Scala

Dianne Marsh
Emerging Technology for the Enterprise Conference
dmarsh@srtsolutions.com
03/26/2009

Outline for Discussion

- Motivation and migration
- Demonstrate key implementation details
- Compare/contrast with Java
- Production Ready?

Quick overview

- Object-oriented/functional hybrid
- Download at www.scala-lang.org
- Runs on JVM
- Lightweight feel
- Static typing

Java 7 and Scala

- Java 7 (Likely)
  - Type Inference
  - ~~Closures~~
  - ~~Properties~~
  - ~~Operator Overloading~~

2010

- Scala
  - Type Inference
  - Closures
  - Properties
  - Operators really functions

Now

SRT SOLUTIONS
We Develop Software Developers

Getting started with Scala

- No statics
- No operators (just methods)
- No primitives (just objects)
- if/else, while
- for (variants of)
- Immutables vs. mutables

Functional

(From David Pollak's Roman Numeral Blog post: http://scala-blogs.org/2008/01/roman-numerals-in-scala.html)

Suspend for a while …

- Variants of for expressions
- Match expressions
- Traits vs. classes

Simple application (Swing)

- Used to scan rare books (online access, on-demand printing)
- Need to tag images with page numbers, page descriptions (TOC, etc.)

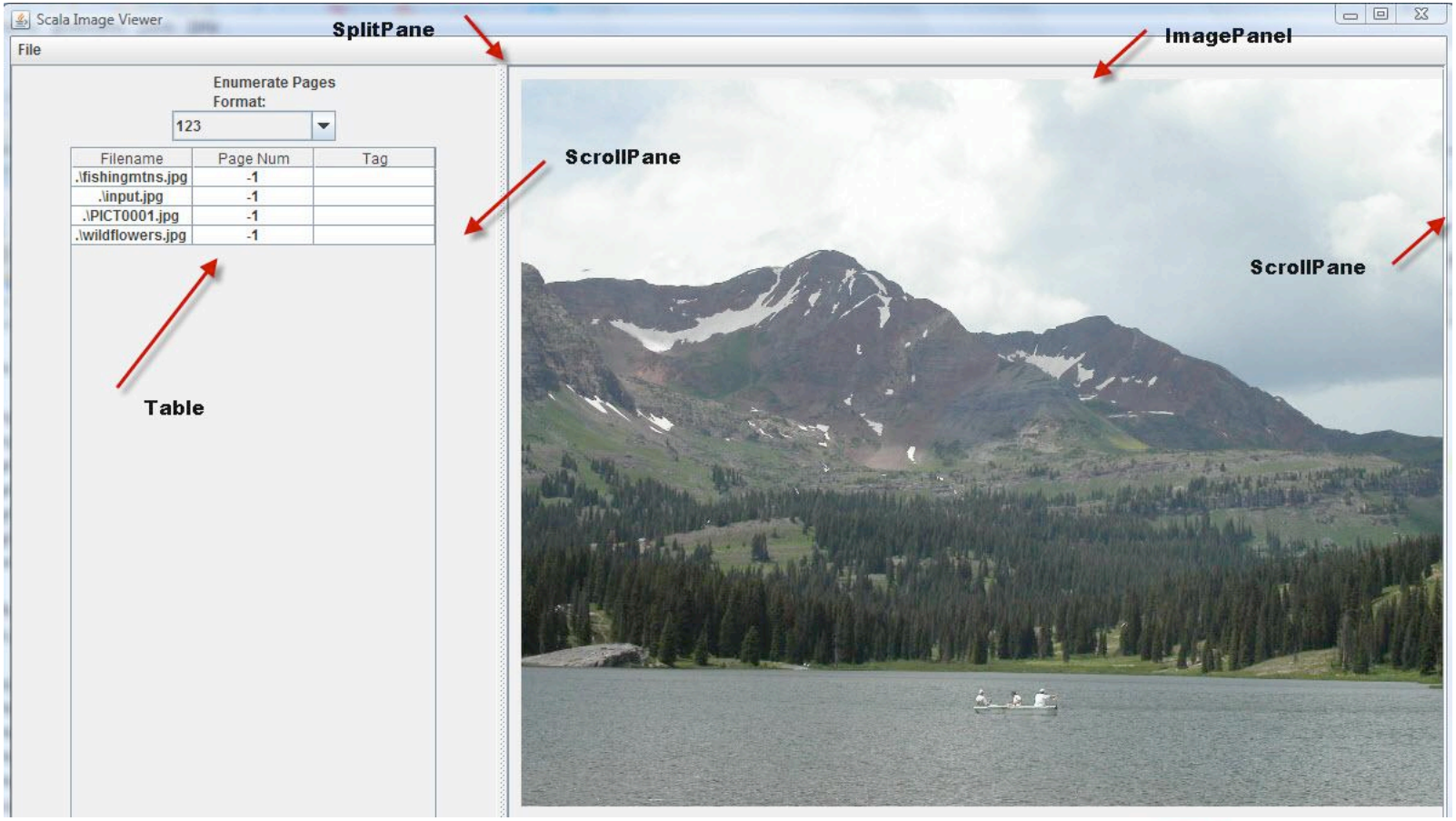
Why?

- Typical desktop app
- Simple functional component (numbering)
- Scala features in context
- Scala wraps Swing library

- If it can simplify Swing …

SRT SOLUTIONS
We Develop Software Developers

Overview



SRT SOLUTIONS
We Develop Software Developers

Features implemented

- Menu bar
- File chooser
- Dialog box
- Display image for selected filename
- Select range in table, enumerate page numbers by Roman Numeral or 123

Really Simple GUI Application

```
/********************** JAVA ************************/
public class SimpleJavaApp {
      public static void main(String []args) {
       SimpleJavaApp theApp = new
      SimpleJavaApp();
    }
    SimpleJavaApp()
    {
        JFrame frame = new JFrame("Title goes
      here");
        frame.setSize(new Dimension(1200, 800));
        frame.setVisible(true);
    }
 }
```

SRT SOLUTIONS
We Develop Software Developers

Really Simple GUI Application

```
/********************** JAVA ************************/
public class SimpleJavaApp {
    public static void main(String []args) {
     SimpleJavaApp theApp = new
    SimpleJavaApp();
  }
  SimpleJavaApp() {
     JFrame frame = new JFrame("Title goes
    here");
     frame.setSize(new Dimension(1200, 800));
     frame.setVisible(true);
  }
}
```

```
/************************ SCALA ************************/
object SampleGUIApp extends
    SimpleGUIApplication {
  def top = new MainFrame {
     title = "First Scala Swing App"
     size = new java.awt.Dimension(1200, 800)
  }
}
```

SRT SOLUTIONS
We Develop Software Developers

Did you notice?

- No semicolons (optional)
- “object” specifies singleton
- “def” starts function definition
- title and size are properties

SRT SOLUTIONS
We Develop Software Developers

Next: add menu bar

```
/********************JAVA**************************/
FileMenuBar menubar = new FileMenuBar(frame);
frame.setJMenuBar(menubar);
```

Next: add menu bar

```
/********************JAVA***************************/
FileMenuBar menubar = new FileMenuBar(frame);
frame.setJMenuBar(menubar);
```

```
/******************** SCALA ****************************/
def top = new MainFrame {
    title = "Title goes here"
    menuBar = new MenuBar
}
```

SRT SOLUTIONS
We Develop Software Developers

Building the Menu

```
/***************** JAVA *********************************/
public class FileMenuBar extends JMenuBar {
        protected JMenu fileMenu;
        public FileMenuBar(ActionListener parent)   {
                fileMenu = new JMenu("File");
                JMenuItem item = new JMenuItem("Open...");
                item.addActionListener(parent);
                fileMenu.add(item);

                item = new JMenuItem("Exit");
                item.addActionListener(parent);
                fileMenu.add(item);

                // add all of the menus to the menubar
                add(fileMenu);
  }
```

Building the Menu

```
/***************** JAVA *********************************/
public class FileMenuBar extends JMenuBar {
        protected JMenu fileMenu;
        public FileMenuBar(ActionListener parent)   {
                fileMenu = new JMenu("File");
                JMenuItem item = new JMenuItem("Open...");
                item.addActionListener(parent);
                fileMenu.add(item);

                item = new JMenuItem("Exit");
                item.addActionListener(parent);
                fileMenu.add(item);

                // add all of the menus to the menubar
                add(fileMenu);
  }
```

```
/************************** SCALA *********************************/
val filemenu = new Menu("File")
  filemenu.contents += new MenuItem(Action("Open"){
          // open file dialog
   })
  filemenu.contents += new MenuItem(Action("Exit"){
       dispose()
   })
  menuBar.contents += filemenu
```

SRT SOLUTIONS
We Develop Software Developers

Handling events in Java

```
public class SecondAppFrame extends JFrame implements ActionListener {

    public void actionPerformed(ActionEvent event)
    {
            if (event.getActionCommand().equals("Open...")) {
            }
            else if (event.getActionCommand().equals("Exit")) {
            }
     }
}
```

(We did this in Scala already)

SRT SOLUTIONS
We Develop Software Developers

Progress …

Menu bar

- Table
  - Class Parameters
  - val vs. var
  - Traits vs. Classes
  - Pattern Matching
  - Properties

Table

```
/************************ JAVA ************************/
public class TableWithMenu extends JTable
      implements ActionListener {
 TableWithMenu(ImagePageTableModel tm,
      ActionListener parent) {
            super(tm);
    }
}
```

Table

```
/************************ JAVA**************************/
public class TableWithMenu extends JTable
      implements ActionListener {
 TableWithMenu(ImagePageTableModel tm,
      ActionListener parent) {
            super(tm);
    }
}
```

```
/********************** SCALA ************************/
var tableData: Array[Array[Any]]

val table = new Table(rows, cols) {
     model = new ChangeableDataTableModel {
          rowData = tableData
          columnNames = List("Filename", "Page
      Num", "Tag")
   }
```

SRT SOLUTIONS
We Develop Software Developers

Scala Table Model

```
class ChangeableDataTableModel(var rowData: Array[Array[Any]], val columnNames: Seq[String]) extends
    AbstractTableModel  {
  override def getColumnName(column: Int) = columnNames(column).toString
  def getRowCount() = rowData.length
  def getColumnCount() = columnNames.length
  def getValueAt(row: Int, col: Int): AnyRef = rowData(row)(col).asInstanceOf[AnyRef]
  override def isCellEditable(row: Int, column: Int) = true
  override def setValueAt(value: Any, row: Int, col: Int) {
    rowData(row)(col) = value
    fireTableCellUpdated(row, col)
  }
  ...
```

SRT SOLUTIONS
We Develop Software Developers

Table Model Notables

- Comparable to Java (not shown)
- AbstractTableModel data unchangeable
- Scala: convenient getter/setter for data

Traits

- Sophisticated mixins (1 or more)
- For reuse in multiple unrelated classes
- Can define behavior/maintain state
- Understand linearization if mixing in several (“super” dynamically bound)

Traits example

What Can't Traits Do?

- No "class" params
- Use pre-initialized fields

Defining Traits (vs. Classes)

Trait:

```
trait ChangeableDataTableModel extends AbstractTableModel
{
    var rowData: Array[Array[Any]]
    var columnNames : Seq[String]

...
```

Class:

```
class ChangeableDataTableModel1(var rowData: Array[Array[Any]], var columnNames: Seq[String]) extends
      AbstractTableModel
```

SRT SOLUTIONS
We Develop Software Developers

ParameterizingTraits (vs. Classes)

Trait:

```
val table = new Table(rows, cols) {
      model = new ChangeableDataTableModel {
      rowData = tableData
      columnNames = List("Filename", "Page Num", "Tag")
 }
```

Class:

```
model = new ChangeableDataTableModel1 (tableData, List("Filename", "Page Num", "Tag"))
```

Properties in Scala

- Every var that is a non-private member of some object automatically gets a getter and a setter
- Can define getter/setter explicitly
- No special syntax

Properties Example

Properties (getters, setters) for vars in class

```
var rowData: Array[Array[Any]]
```

Generates (implementations for):

```
def rowData : Array[Array[Any]]
def rowData_=(d: Array[Array[Any]])
```

Accessed by caller as:

```
tableModel.rowData = tableData
```

SRT SOLUTIONS
We Develop Software Developers

Java Listeners, Handlers

```
public class ImageSplitPane extends JFrame implements ActionListener, ListSelectionListener {
...
    table.getSelectionModel().addListSelectionListener(this);

    public synchronized void valueChanged(ListSelectionEvent e)
    {
...
     }
}
```

Java Listeners, Handlers, cont.

```
// ImageSplitPane, cont ...
. . .
    public void actionPerformed(ActionEvent event)
    {
        if (event.getActionCommand().equals("Exit"))          handleExit();
        else if (event.getActionCommand().equals("Open ..."))  handleOpenFile();
        else if (event.getActionCommand().equals("Roman"))     enumerateRoman();
. . .
    }
}
```

SRT SOLUTIONS
We Develop Software Developers

Scala Listeners, Handlers

```
val editUI = new BoxPanel(Orientation.Vertical) {
    val numberBox = new ComboBox(List("123", "Roman", "Clear"))
    contents += numberBox
    listenTo(numberBox.selection)
    listenTo(table.selection)
    reactions += {
        case TableRowsSelected(_, range, false) =>
            replaceImage(imagename)
        case SelectionChanged(`numberBox`) =>
            processSelection(table, numberBox.selection.item)
        case _ =>
            true
    }
...
```

Simple Pattern Matching

```
def processSelection(table: Table, selectedItem: Any) {
    selectedItem match {
    case "123" =>
        val entered = startVal.toInt
        val startRow = table.selection.rows.toArray(0)
        table.selection.rows.foreach( r => table.update(r, 1, r - startRow + entered))
        true
    case "Roman" =>
        functionalRomanEnumeration(table, startVal)
        true
    case _ =>
        false
}
```

SRT SOLUTIONS
We Develop Software Developers

Match Expressions

- No fall-throughs between cases
- Must match something (use _)
- No “return” needed at end of block
- Always result in a value

More than just switch

```
override def rendererComponent(isSelected: Boolean, hasFocus: Boolean, row: Int, column: Int):
    Component =
        ImageViewer.rowData(row)(column) match {
            case s: String => l.componentFor(this, isSelected, hasFocus, s, row, column)
            case _ => super.rendererComponent(isSelected, hasFocus, row, column)
        }
```

Other Notables …

- Generics
- Interoperability
- Testing
- Production environment?

SRT SOLUTIONS
We Develop Software Developers

Generics

- Like Java, still uses erasure (type info not preserved at runtime)
- Supports variance annotations

Simple Generics Example

```
/************* JAVA ******************
class Pair<K,V> {
K key;
V value;
 Pair(K key, V value) {
     this.key = key
     this.value = value
}
```

```
/*********************** SCALA ************************/
class Pair[K, V] (val key: K, val value: V) {
}
```

Java/Scala interoperability

- Scala Swing library code
- Daniel Spiewak's blog: http://www.codecommit.com/blog/java

What about testing?

- JUnit 3 and 4 supported
- Scalatest

SRT SOLUTIONS
We Develop Software Developers

Using JUnit Tests in Scala

```
class TestRoman extends TestCase {
  def testRomanize = {
    assertTrue(RomanNumeralConverter.romanize(5) == "V")
    assertTrue(RomanNumeralConverter.romanize(100) == "C")
    assertTrue(RomanNumeralConverter.romanize(7) == "VII")
    assertTrue(RomanNumeralConverter.romanize(1964) == "MCMLXIV")
  }

  def testValidFormat = {
    assertTrue(RomanNumeralConverter.isValidRoman("0") == false)
  }
```

ScalaTest

- http://www.artima.com/scalatest/
- Supports behavior driven testing
- Traditional testing
- JUnit/ScalaCheck Integration
- TestNG style

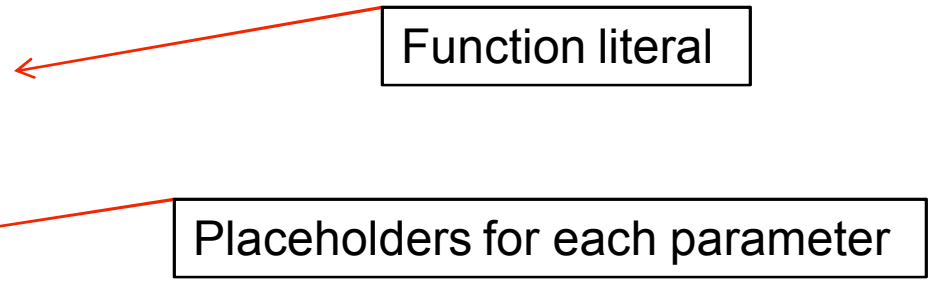
Deployment?

- Can make a jar
  - Include scala-library.jar
  - Include scala-swing.jar
  - Size issues
- Scala bazaar

What Happened to FUNCTIONAL?

- Getting your code (brain?) ready
  - vals vs. vars
  - Immutable collections
  - Use pattern matching
  - Start exploring map, filter, etc.

OK, just a taste …



SRT SOLUTIONS
We Develop Software Developers

Principles to like …

- Conciseness
- Uniform Access Principle
- Building Blocks + Libraries

Specifics to like …

- Type Inference
- Class Parameters
- Properties
- Pattern Matching
- For expressions
- Multiple return values using tuples

Ready … Set … Go?

- Still early adopter stage
- IDE: plugins for NetBeans, IntelliJ, Eclipse
- Documentation: sign of maturity

SRT SOLUTIONS
We Develop Software Developers

The JVM needs “Java 7”

- JVM improvements critical
- Tail call optimization

For more information …

- "*Programming in Scala*", book by Odersky, Spoon, Venners on Artima
- *"Beginning Scala"* – David Pollak, alpha from APress
- "*Programming Scala*" (concurrency) – Venkat Subramaniam, May 2009, Prags
- Java Posse podcast (Roundup '08 recording: "*What does Scala Need*")
- *Java Posse Roundup '09 recordings …*

SRT SOLUTIONS
We Develop Software Developers