User Interfaces Made Simple with Groovy and Swing

James Williams

Agenda

- Brief overview of the Google Summer of Code
- Groovy Primer
- SwingX/SwingXBuilder
- Painters

[any material that should appear in print but not on the slide]

What is the Google Summer of Code?

A program sponsored by Google to:

- get more open source software created
- inspire young developers to participate in open source software
- provide students with summer jobs that apply to their majors
- give students more exposure to real world software development scenarios

Program Statistics

- Ran from May 28th to August 20th
- Students are given a $4500 stipend and mentoring organizations receive $500
- Over 900 students were accepted out of more than 6,000 applications
- 81% of the participants successfully completed the program

What is Groovy?

- an agile, dynamic language for the Java Virtual Machine
- an easy entry point for Java developers who want a scripting language but want to leverage Java knowledge
- can make use of Java libraries or classes
- can subclass Java classes and vice versa
- can be executed directly from Java with the Scripting API(JSR 223) or on its own
- has a Java-like syntax

Groovy can run most Java

This is valid Groovy code:

```
public class HelloWorld {
     public static void main(String [] args) {
```

```
            System.out.println("Hello, World!");
     }
}
```

But this is much more concise:

```
println "Hello, World!"
```

Declarations

Variables and functions must have an access modifier and or a type, or def.
These are valid declarations:

```
def a
int i
private float b
def fact(n) { .. }
```

Classes

- Accessor and mutator methods are generated automatically
- The no-argument constructor is implied so declaring it is optional
- Member variables can be instantiated from the constructor
- Given this typical class:

```
class Album {
     String title
     String artist
     List songs = []
     float price
}
```

Classes Part 2

Let's instantiate an Album object:

```
def album = new Album(title:'My Album', songs:['Song 1', 'Song 2', 'Song 3'])
```

Let's set the price:

```
album.setPrice(10.00)
```

Let's set the artist:

```
album.artist = 'Some Girl'
```

Note: album.title is NOT field-access, it's a shortcut for the mutator function.
-->

Closures

- are snippets of code that can be used as an object
- can exist outside a class
- can take input parameters and return values
- have the built-in 'it' variable

Some examples:

```
def square = { it * it }
3.times { println "Hi" }
1.upto(5) {println it }
```

```
20.downto(15) { x *= it }
```

Collections - Lists

- All of the Java functions work on Groovy Lists if you remember that they are automatically ArrayLists.
- An empty list is declared with empty brackets. `Ex: def values = []`
- There are many helper functions in Groovy to help with the manipulation of lists, here are a few:

```
values.each {println it }
 values.find {it > 10 }
 values.findAll { it > 10}
 values += [10]
 values -= [10]
```

Collections - Maps

- Maps are automatically LinkedHashMaps
- An empty list is declared with empty brackets with a colon. `Ex: def namesAges = [:]`
- Given the map declared with `def nameAges = ['John':18, 'Mary':23, 'Joe':25]` Here are some operations on that data:

```
nameAges.each {println it }
 nameAges.find {it.value > 10 }
 nameAges.findAll { it.key != 'Mary'}
 println nameAges.Joe
```

Builders

- are nested tree-structures used to build complex objects
- increase readability and writeability of code
- are fun to use.
- prebuilt into Groovy include AntBuilder, SAXBuilder, MarkupBuilder, and SwingBuilder

SwingBuilder Example

```
import javax.swing.WindowConstants
import groovy.swing.SwingXBuilder

def swing = new SwingXBuilder()

def frame = swing.frame(size:[200,200],
defaultCloseOperation:WindowConstants.EXIT_ON_CLOSE) {
    label("Hello, World!")
}

frame.show()
```

What is SwingX?

- They are extensions to classic Swing components and libraries.
- They provide polish and shortcuts that the users expect anyway
- Several components have graduated to mainline Swing including GroupLayout and the SystemTray

Painters

- are delegates that components can use to draw their backgrounds and foregrounds
- can be combined to form more complex CompoundPainters
- can be modified by using Effects to adjust how they are drawn

Painter Demo

Demo

Painter Demo Source

```
def compoundPaint = swing.compoundPainter() {
    mattePainter(fillPaint:new Color(51,51,51))
    pinstripePainter(paint:new Color(1.0f,1.0f,1.0f,0.17f), spacing:5.0f)
    glossPainter(paint:new Color(1.0f,1.0f,1.0f,0.2f),
position:GlossPainter.GlossPosition.TOP)
}
```

Links

- Groovy
  http://groovy.codehaus.org
- SwingLabs
  http://www.swinglabs.org
- SwingBuilder
  http://groovy.codehaus.org/Swing+Builder
- SwingXBuilder
  http://groovy.codehaus.org/SwingXBuilder
- Shameless plug:
  My blog: http://jameswilliams.be/blog