A Tour of the Scala Programming Language

Julien Wetterwald
CS94SI

The Scala Programming Language

- stands for “scalable language”
- multi-paradigm language
- runs on the standard Java and .NET platforms
- interoperates seamlessly with all Java libraries

Scalable

Scalability of the systems and the methods for constructing them:

- build interesting structures from simple parts
- reuse and adapt existing components

Multi-paradigm

- integrates features of functional and object-oriented languages
- the two programming styles are complementary

Scala is functional

- every function is a value
- lightweight syntax for defining anonymous functions
- higher-order functions
- sequence comprehensions
- pattern matching

Scala is object-oriented

- every value is an object
- types and behavior of objects are described by classes and traits
- class abstractions are extended by subclassing and mixin composition

Scala is also...

- statically typed
- extensible
- concise (factor of 10 compared to Java)
  - higher-level than Java
  - local type inference (no redundant type annotations)

Java

```
boolean nameHasUpperCase = false;
for (int i = 0; i < name.length(); ++i) {
    if (Character.isUpperCase(name.charAt(i))) {
        nameHasUpperCase = true;
        break;
    }
}
```

Scala

```
val nameHasUpperCase = name.exists(_.isUpperCase)
```

Who is responsible for Scala?

- the LAMP group at EPFL, Switzerland
- led by Prof. Martin Odersky
  - wrote javac 1.1-1.4 and designed the good parts of Java Generics
  - ACM Fellow
- a quickly growing community of contributors, users and bloggers

Resources

- the Scala web site
  http://www.scala-lang.org
- the Scala mailing list
  http://listes.epfl.ch/doc_en.cgi?liste=scala-user
- ... more resources on the course web site
  http://cs94si.stanford.edu