Scala programming language

Dimitris Halatsis

Software Engineering 2018

PART I
Introduction to Scala

Scala is

- Functional
- Pure object-oriented
- Statically typed

Functional

- Every Function is a Value
- Currying
- Higher Order Functions
- Singleton Object
- Case Classes
- Pattern Matching
- Lazy Evaluation

Functions vs Methods

Functions

```scala
val addOne = (x: Int) => x + 1
println(addOne(1)) // 2
```

```scala
val add = (x: Int, y: Int) => x + y
println(add(1, 2)) // 3
```

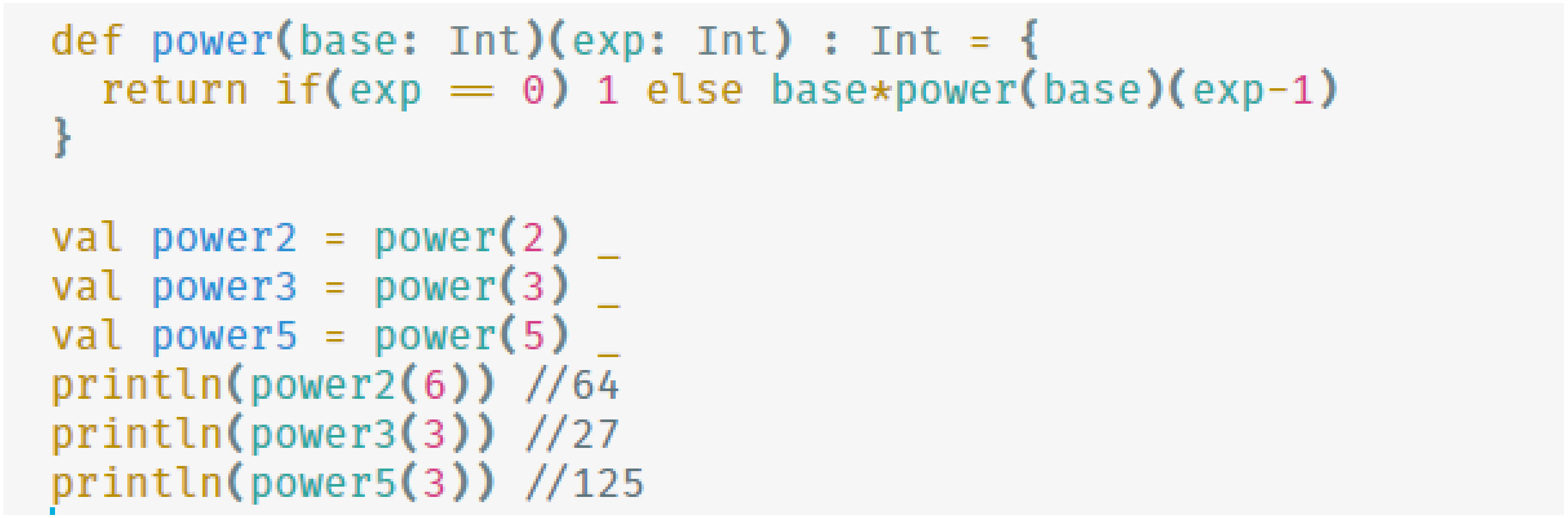
```scala
val getTheAnswer = () => 42
println(getTheAnswer()) // 42
```

Methods

```scala
def add(x: Int, y: Int): Int = x + y
println(add(1, 2)) // 3
```

```scala
def getSquareString(input: Double): String = {
  val square = input * input
  square.toString
}
```

Currying



```scala
def power(base: Int)(exp: Int) : Int = {
  return if(exp == 0) 1 else base*power(base)(exp-1)
}

val power2 = power(2) _
val power3 = power(3) _
val power5 = power(5) _
println(power2(6)) //64
println(power3(3)) //27
println(power5(3)) //125
```

Higher Order Functions

```scala
def sum(f: Int => Double, a: Int, b: Int): Double =
  if (a > b) 0
  else f(a) + sum(f, a + 1, b)


val square = (x: Int) => 1.0*x*x
val cube = (x: Int) => 1.0*x*x*x
val negative = (x: Int) => 1.0/x

println(sum(square, 2, 5))      // 54.0
println(sum(cube, 2, 9))        // 2024.0
println(sum(negative, 2, 1024)) // 6.509175672278128
```

Case Classes and Pattern Matching

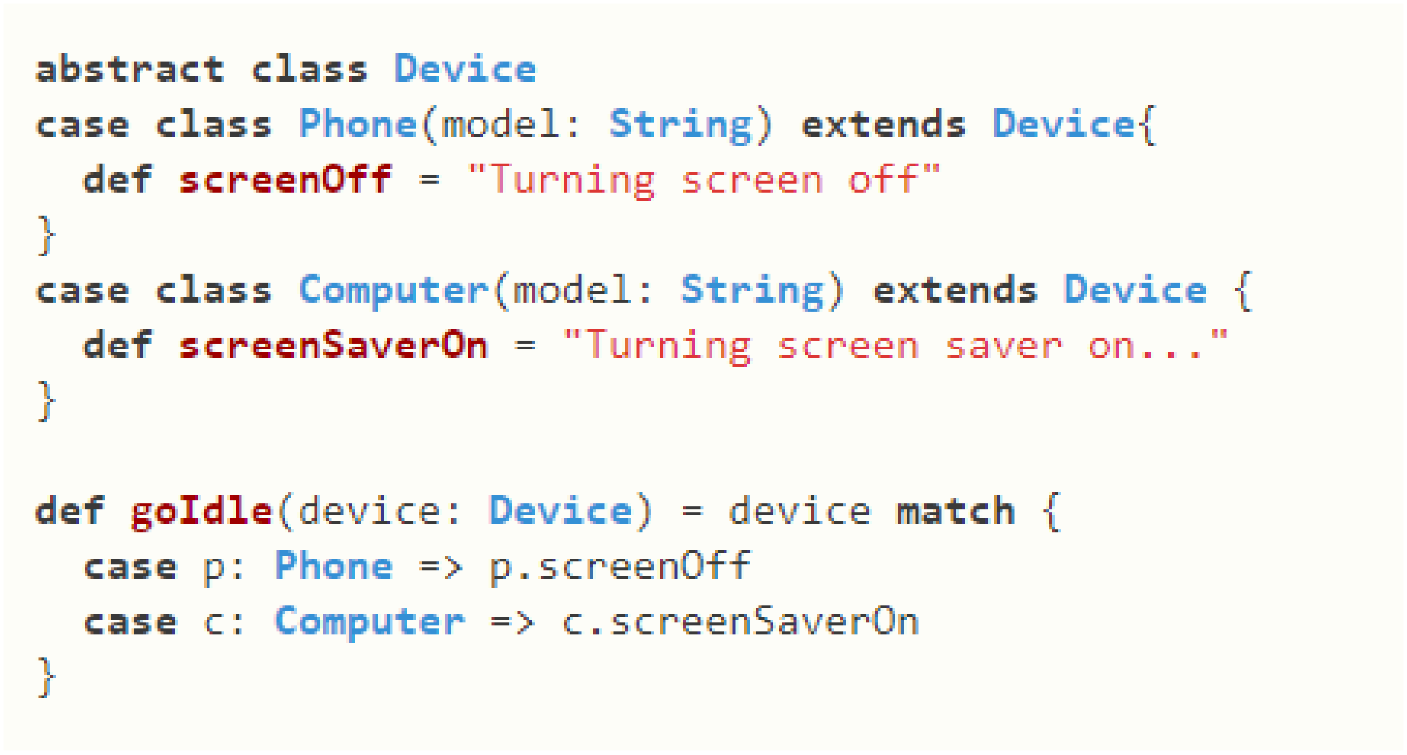
```scala
abstract class Notification

case class Email(sender: String, title: String, body: String) extends Notification

case class SMS(caller: String, message: String) extends Notification

case class VoiceRecording(contactName: String, link: String) extends Notification
```

```scala
val someSms = SMS("867-5309", "Are you there?")
val someVoiceRecording = VoiceRecording("Tom", "voicerecording.org/id/123")
val importantEmail = Email("jenny@gmail.com", "Drinks tonight?", "I'm free after 5!")
val importantSms = SMS("867-5309", "I'm here! Where are you?")
```

Case Classes and Pattern Matching

```scala
def showNotification(notification: Notification): String = {
  notification match {
    case Email(email, title, _) =>
      s"You got an email from $email with title: $title"
    case SMS(number, message) =>
      s"You got an SMS from $number! Message: $message"
    case VoiceRecording(name, link) =>
      s"you received a Voice Recording from $name! Click the link to hear it: $link"
  }
}
val someSms = SMS("12345", "Are you there?")
val someVoiceRecording = VoiceRecording("Tom", "voicerecording.org/id/123")

println(showNotification(someSms))  // prints You got an SMS from 12345! Message: Are you there?

println(showNotification(someVoiceRecording))  // you received a Voice Recording from Tom!
                                              //Click the link to hear it: voicerecording.org/id/123
```

Case Classes and Pattern Matching


```
abstract class Device
case class Phone(model: String) extends Device{
  def screenOff = "Turning screen off"
}
case class Computer(model: String) extends Device {
  def screenSaverOn = "Turning screen saver on..."
}

def goIdle(device: Device) = device match {
  case p: Phone => p.screenOff
  case c: Computer => c.screenSaverOn
}
```

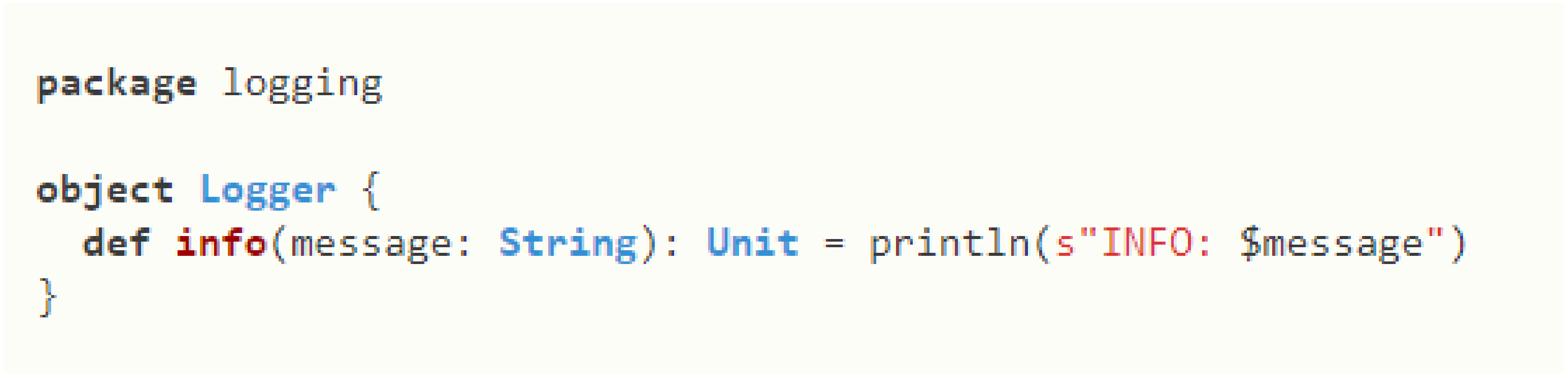
Lazy evaluation

```scala
def fibo(n: Int): Int = {

  if (n== 1 || n == 0)
    return 1
  else
    return fibo(n-1)+fibo(n-2)
}

lazy val lval = fibo(49353)

println(0) //prints 0!
```

```scala
def power(base: => Int, exp: => Int): Int = {
  var rv = 1
  if (base == 1)
    return rv
  else
    for (i <- 1 to exp)
      rv *= base
  return rv
}

println(power(1,fibo(49353))) //prints 1!
```

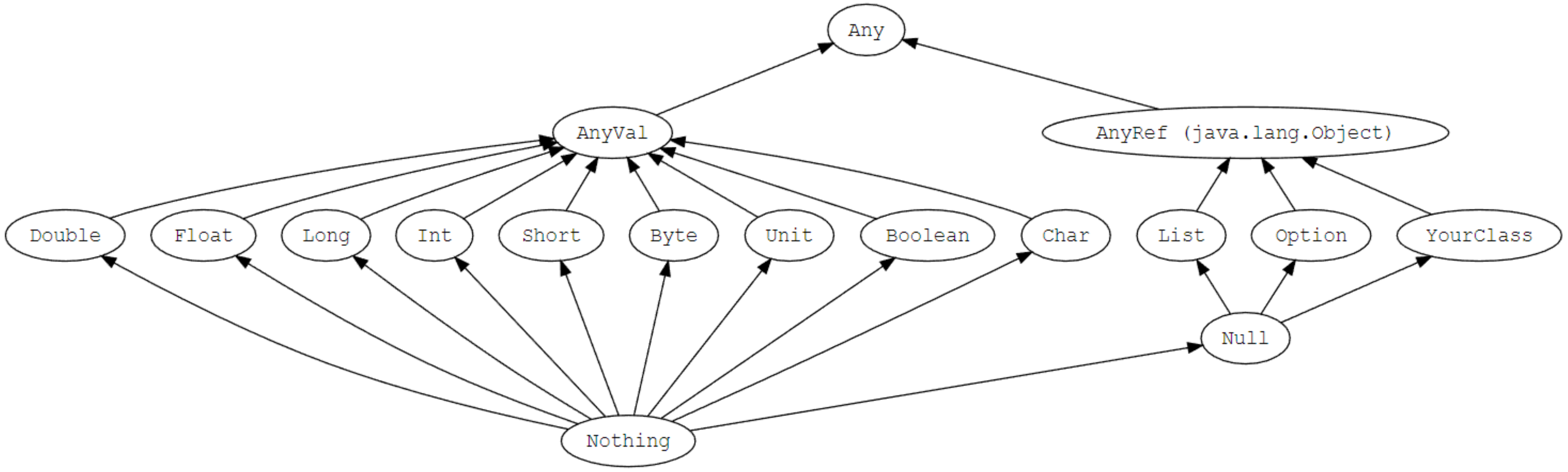
```scala
import scala.math.sqrt

def streamRange(lo: Int, hi: Int): Stream[Int] =
  if (lo >= hi) Stream.empty
  else Stream.cons(lo, streamRange(lo + 1, hi))

def isPrime(n: Int) =  {
  !Range(2, sqrt(n).toInt+1).exists(n % _ == 0) }

println(
  ((1000 to 10000) filter isPrime)(1))

println(
  (streamRange(1000,10000) filter isPrime)(1))
```

Singleton Object



```scala
package logging

object Logger {
  def info(message: String): Unit = println(s"INFO: $message")
}
```

Pure object-oriented



Traits

```scala
trait Iterator[A] {
  def hasNext: Boolean
  def next(): A
}

class IntIterator(to: Int) extends Iterator[Int] {
  private var current = 0
  override def hasNext: Boolean = current < to
  override def next(): Int =  {
    if (hasNext) {
      val t = current
      current += 1
      t
    } else 0
  }
}


val iterator = new IntIterator(10)
iterator.next()  // returns 0
iterator.next()  // returns 1
```

```scala
import scala.collection.mutable.ArrayBuffer

trait Pet {
  val name: String
}

class Cat(val name: String) extends Pet
class Dog(val name: String) extends Pet

val dog = new Dog("Harry")
val cat = new Cat("Sally")

val animals = ArrayBuffer.empty[Pet]
animals.append(dog)
animals.append(cat)
animals.foreach(pet => println(pet.name))  // Prints
```

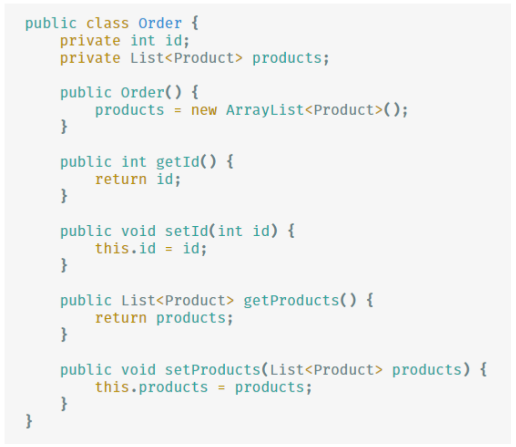
Class composition and mixins

```scala
abstract class A {
  val message: String
}
class B extends A {
  val message = "I'm an instance of class B"
}
trait C extends A {
  def loudMessage = message.toUpperCase()
}
class D extends B with C

val d = new D
println(d.message)  // I'm an instance of class B
println(d.loudMessage)  // I'M AN INSTANCE OF CLASS B
```

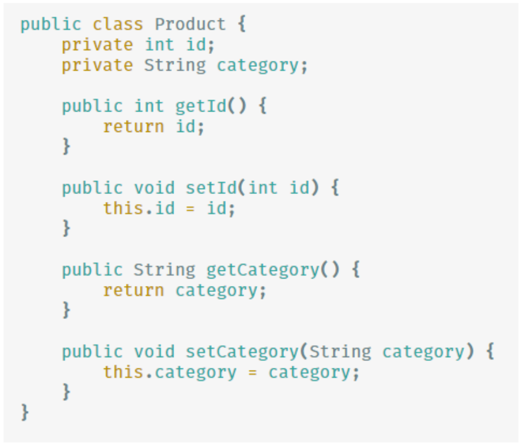
$$\begin{tikzcd}
 & A \arrow[dl] \arrow[dr] & \\
B \arrow[dr] & & C \\
 & D &
\end{tikzcd}$$

PART II
Scala vs Java

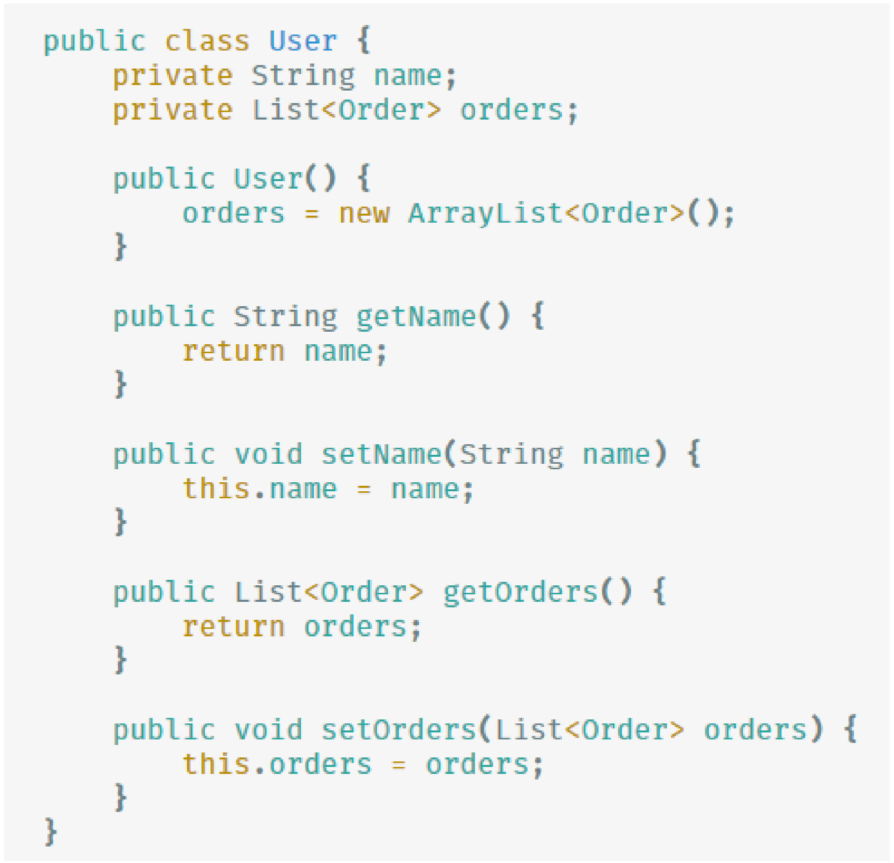
Traditional class with setters/getters in Java



```java
public class Order {
    private int id;
    private List<Product> products;

    public Order() {
        products = new ArrayList<Product>();
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public List<Product> getProducts() {
        return products;
    }

    public void setProducts(List<Product> products) {
        this.products = products;
    }
}
```

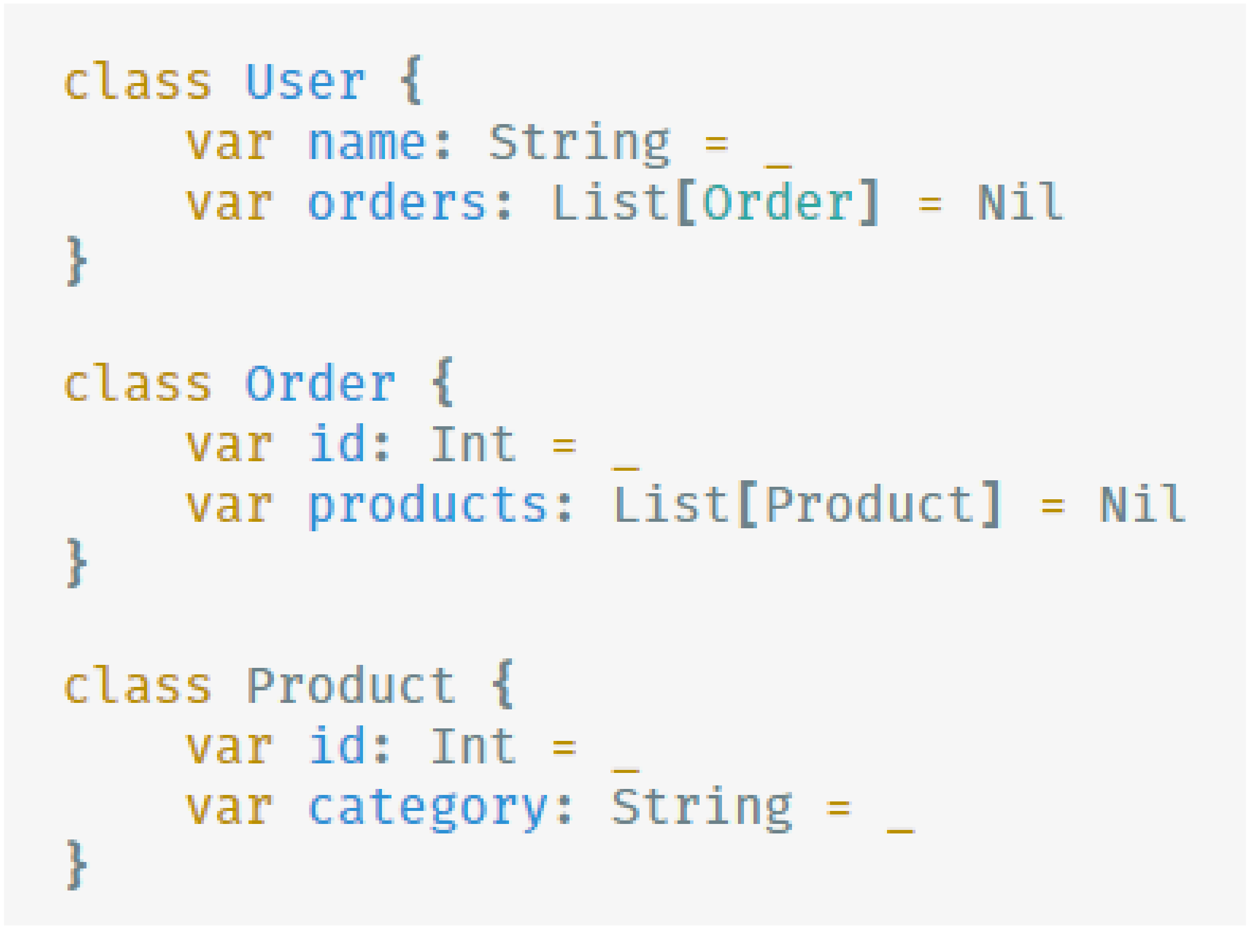


```java
public class Product {
    private int id;
    private String category;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }
}
```



```java
public class User {
    private String name;
    private List<Order> orders;

    public User() {
        orders = new ArrayList<Order>();
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Order> getOrders() {
        return orders;
    }

    public void setOrders(List<Order> orders) {
        this.orders = orders;
    }
}
```

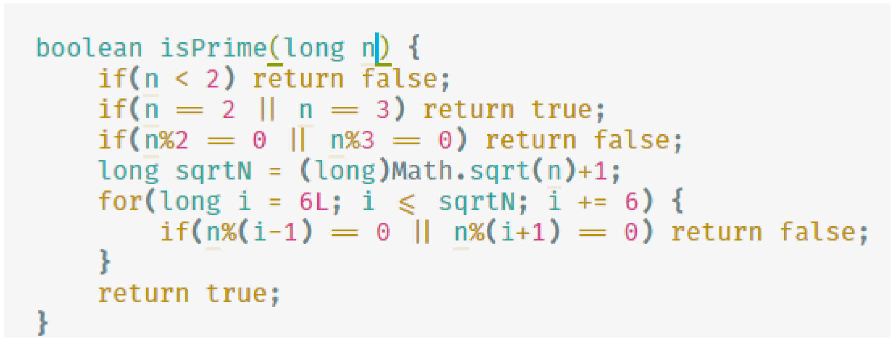
...and in Scala



```scala
class User {
    var name: String = _
    var orders: List[Order] = Nil
}

class Order {
    var id: Int = _
    var products: List[Product] = Nil
}

class Product {
    var id: Int = _
    var category: String = _
}
```

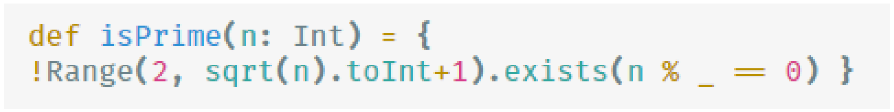
High level

java



```java
boolean isPrime(long n) {
    if(n < 2) return false;
    if(n == 2 || n == 3) return true;
    if(n%2 == 0 || n%3 == 0) return false;
    long sqrtN = (long)Math.sqrt(n)+1;
    for(long i = 6L; i <= sqrtN; i += 6) {
        if(n%(i-1) == 0 || n%(i+1) == 0) return false;
    }
    return true;
}
```

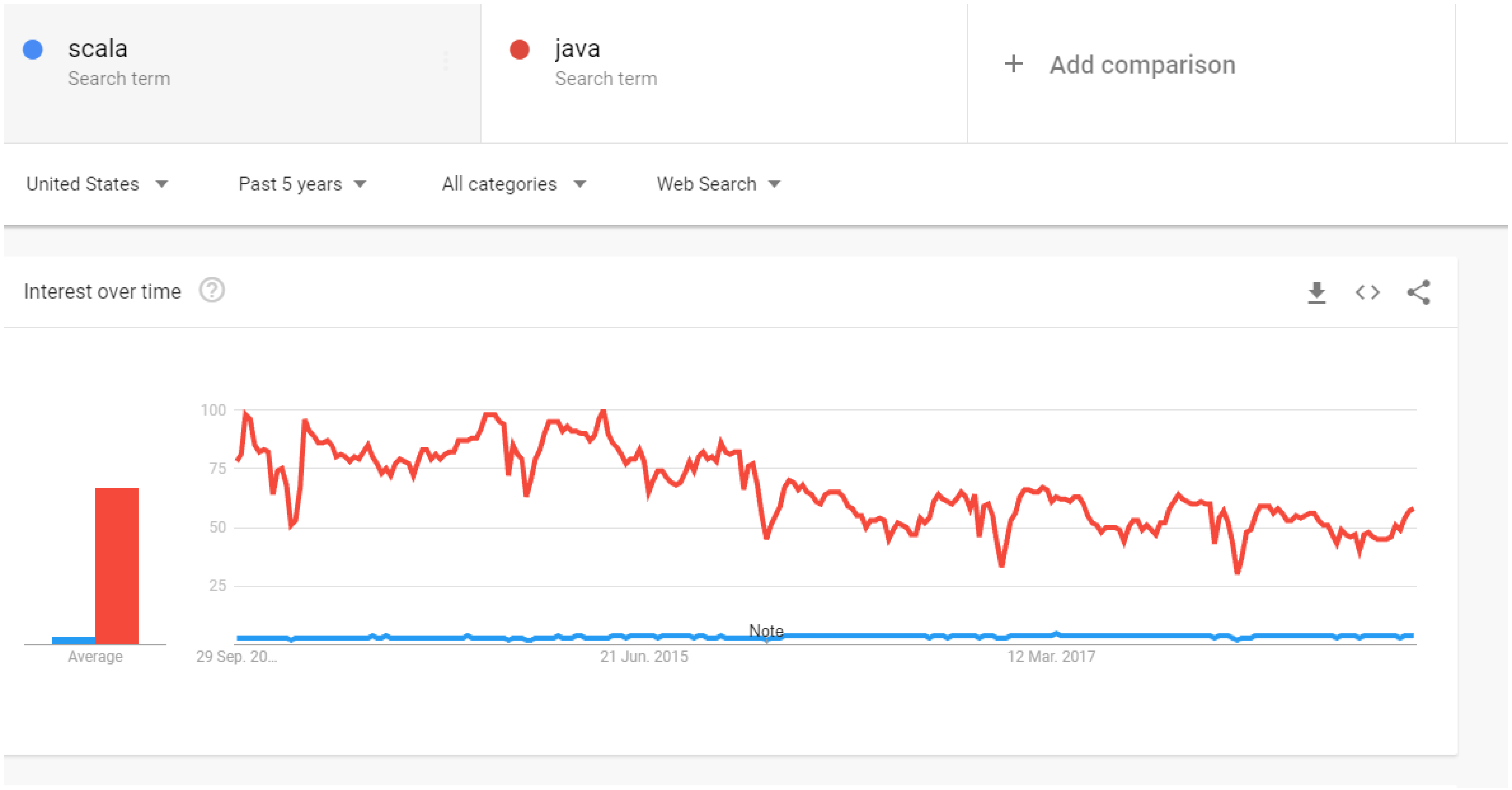
scala



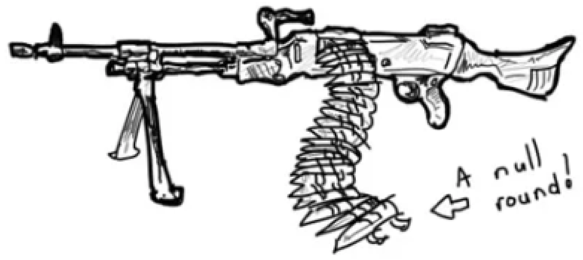
```scala
def isPrime(n: Int) = {
!Range(2, sqrt(n).toInt+1).exists(n % _ == 0) }
```

Scala in big applications

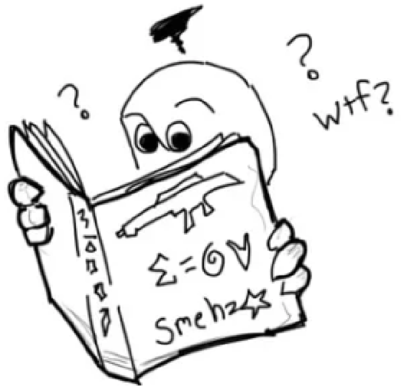
Trends


● scala
Search term
● java
Search term
+ Add comparison
United States
Past 5 years
All categories
Web Search

If Programming Languages Were Weapons



Java is a belt fed 240G automatic weapon where sometimes the belt has rounds, sometimes it doesn't, and when it doesn't during firing you get an NullPointerException, the gun explodes and you die.



Scala is a variant of the 240G Java, except the training manual is written in an incomprehensible dialect which many suspect is just gibberish.

Sources

- Scala's official documentation
- https://www.scala-exercises.org
- Why should I learn Scala?
- 9gag

Thank you