The Scala Programming Language

Troy Hut and Benjamin Killeen

Introduction

Scala is:

Introduction

Scala is:

- Object oriented

Introduction

Scala is:

- Object oriented
- Functional

Introduction

Scala is:

- Object oriented
- Functional
- Statically typed

Scala is **Scalable**

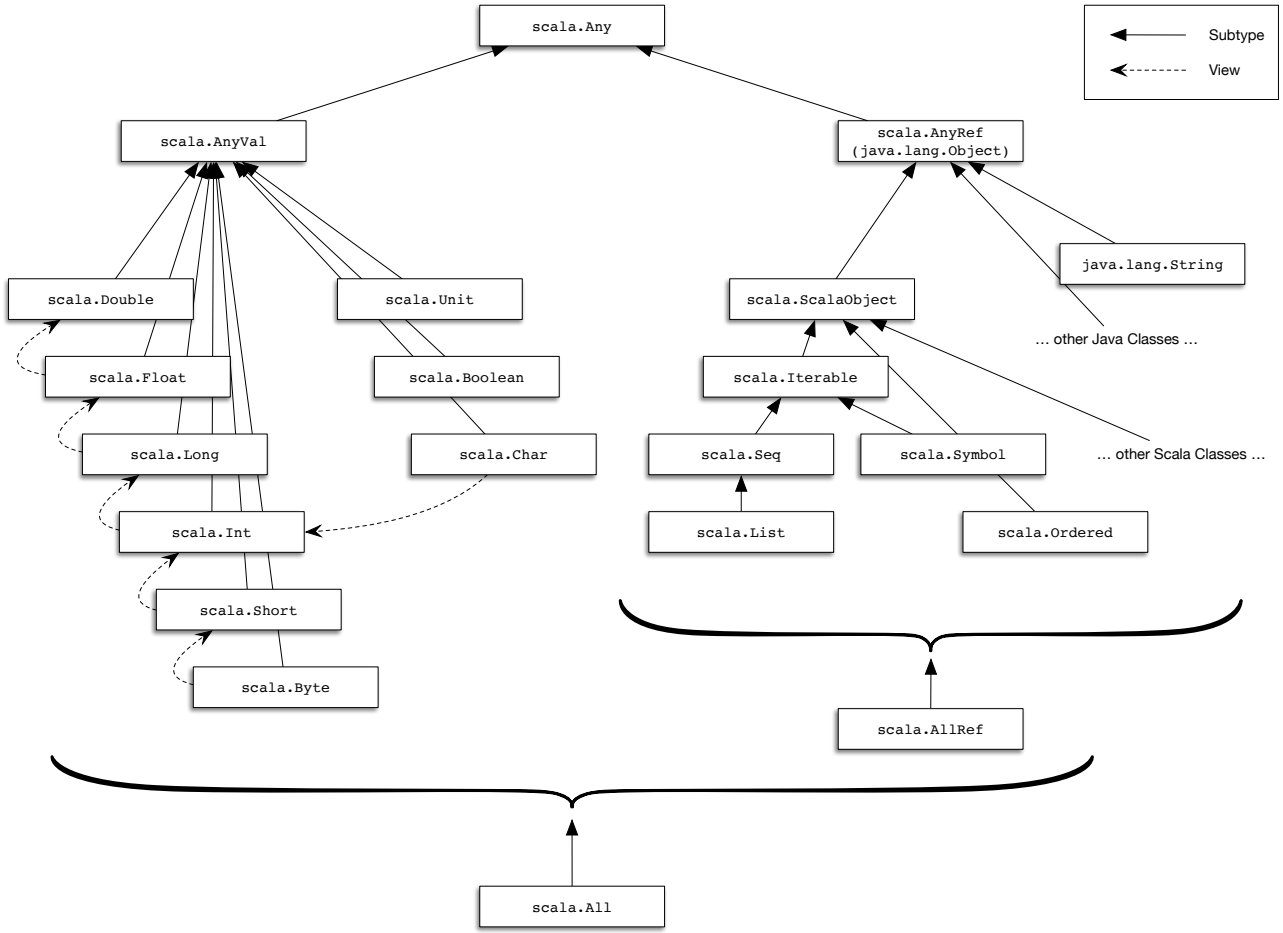

Figure: The Scala class hierarchy.

Traits, Objects, and Classes


```
trait Nat {
  def isZero (): Boolean
  def pred (): Nat
  def succ (): Nat
}
```

Traits, Objects, and Classes


```
trait Nat {
  def isZero (): Boolean
  def pred (): Nat
  def succ (): Nat
}

object Z extends Nat {
  def isZero () = true
  def succ () = S(this)
  def pred () = this
}
```

Traits, Objects, and Classes


```
trait Nat {
  def isZero (): Boolean
  def pred (): Nat
  def succ (): Nat
}

object Z extends Nat {
  def isZero () = true
  def succ () = S(this)
  def pred () = this
}

case class S(v: Nat) extends Nat {
  def isZero () = false
  def succ () = S(this)
  def pred () = v
}
```

The nat Package


```
package nat

trait Nat {
  def isZero: Boolean
  def pred: Nat
  def succ: Nat
  def + (y: Nat): Nat = if (y.isZero) this else this.succ + y.pred
  def - (y: Nat): Nat = if (y.isZero) this else this.pred + y.pred
  def * (y: Nat): Nat = if (y.isZero) y else this + this * y.pred
  def ^ (y: Nat): Nat = if (y.isZero) y.succ else this * (this ^ y.pred)
  def == (y: Nat): Boolean = {
    if (this.isZero) y.isZero
    else if (y.isZero) false
    else this.pred == y.pred
  }
  def != (y: Nat): Boolean = !(this == y)
  def > (y: Nat): Boolean = (this.isZero, y.isZero) match {
    case (true, _) => false
    case (false, true) => true
    case (false, false) => this.pred > y.pred
  }
  def >= (y: Nat): Boolean = (this.isZero, y.isZero) match {
    case (_, true) => true
    case (true, false) => false
    case (false, false) => this.pred >= y.pred
  }
  def <= (y: Nat): Boolean = !(this > y)
  def < (y: Nat): Boolean = !(this >= y)
}
```