

Scala – Javas arvtager?

Eivind Barstad Waaler

BEKK/UiO

19. oktober 2009

Hva er Scala?

- Objektorientering + Funksjonell programmering
- Kompilert språk
- Statisk/sterkt typet
- Java bytecode → JVM
- Kort historie:
 - 2003 – Scala versjon 1.0
 - 2006 – Scala versjon 2.0
 - I dag – Scala versjon 2.7.7 → 2.8

Syntaks

```
// Typen skrives bak
val name: String = "Eivind"
// Eller oppdages automatisk av kompilator (inference)
val comp = "Bekk"

// val - immutable (konstant verdi)
val noChangy = 5
noChangy *= 2 // error: reassignment to val
// var - mutable (variabel verdi)
var changy = 3
changy *= 2 // 10

// Metoder med type og returverdi
def toUpper(s: String): String = {
    s.toUpperCase
}
```

Høyere-ordens funksjoner

```
// Funksjon som variabel
val doubleIt = (i: Int) => i * 2

// Funksjon som parameter
val intArr = Array(1, 2, 3, 4, 5)
val doubledArr = intArr.map(doubleIt)

// Funksjon som returverdi
def intFunc: (Int => Int) = (i: Int) => i * 2
val func = intFunc
val four = func(2)
```

Anonyme metoder & Closures

```
val arr = Array(5, 4, 3, 2, 1)

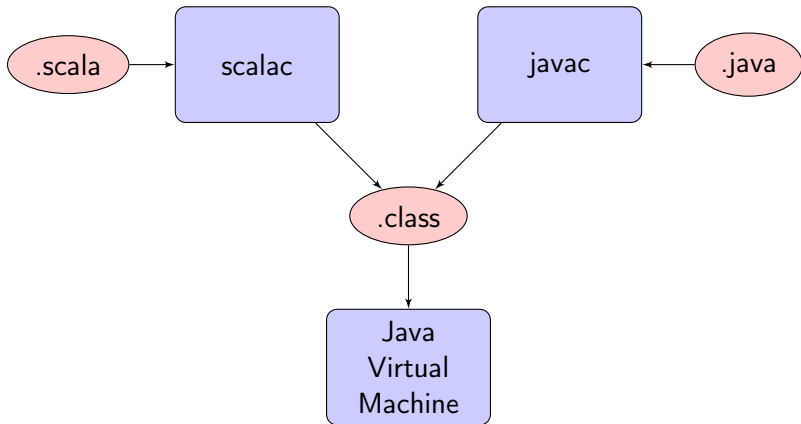
// Finn partall - anonym funksjon
val even = arr.filter(_ % 2 == 0)

// Closure - anonym funksjon med variabel
var threshold = 3
val func = (i: Int) => i < threshold
arr.filter(func) // Array(1, 2)
threshold = 4
arr.filter(func) // Array(1, 2, 3)
```

Traits

```
trait Greetable {  
  def sayHi(s: String) = "Hi " + s  
}  
  
trait Leavable {  
  def sayBye(s: String) = "Bye bye, " + s  
}  
  
class Person extends Greetable with Leavable
```

Scala & Java



Oppsummering

- Funksjoner og traits – bare toppen av kaka
- Statisk eller dynamisk typing best?
- Kompakt kode – 100% Java-kompatibel
- Javas arvtager?

Mer informasjon:

<http://www.scala-lang.org/>