Where we are & where we’re going

general-purpose
- library
  - discover nouns & verbs
- API
  - add fluency

domain-specific
- internal
  - remove host flavor
- external
  - tools & environments

general-purpose?
Use case classes to combine structures

abstract class Description(val name: Char) {
    override def toString = name.toString
}

object Open extends Description(' ')
object Blocked extends Description('▌')
object Anything extends Description('*')

class Surroundings(val north: Description, val east: Description,
                    val south: Description, val west: Description)

US

case class Surroundings(north: Description, east: Description,
                         south: Description, west: Description)
Scala generates lots of useful code for case classes

```scala
val trapped1 = Surroundings(Blocked, Blocked, Blocked, Blocked)
val trapped2 = Surroundings(Blocked, Blocked, Blocked, Blocked)

scala> trapped1.north
res11: Description = □

scala> trapped1 == trapped2
res12: Boolean = true

scala> println(trapped1)
Surroundings(█,█,█,█)

scala> trapped1 match {
  |   case Surroundings(Open, _, _, _) ⇒ println("north is free")
  |   case Surroundings( Blocked, _, _, _) ⇒ println("north is blocked")
  | }
north is blocked
```

- value parameters
- equals method
- toString method
- pattern matching (i.e., extractors)
CS 111 Fall ’15: The Moving Bears
The Expression Problem

There are multiple formulations. This is one of them.

- $\exists$ a library $L$ of nouns (data) and verbs (behaviors)
- Person A extends $L$ to add a new noun (data)
- Person B extends $L$ to add a new verb (behavior)
- Person C wants to *safely* combine A and B’s extensions

*User-defined Types and Procedural Data Structures as Complementary Approaches to Data Abstraction*, John Reynolds, 1975

*Object-Oriented Programming Versus Abstract Data Types*, William R. Cook, 1990

*The Expression Problem*, Philip Wadler, 1998

*Independently Extensible Solutions to the Expression Problem*, Matthias Zenger & Martin Odersky, 2005

*Data types à la carte*, Wouter Swierstra, 2008