

Objects (OOP)

Encoding objects in FbSR

- * probably: classes
- * behavior & data
- * inheritance, polymorphism, encapsulation
- * mutation[†]
- * self-referential

$$\langle S_0, e_1 \rangle \Rightarrow \langle S_1, v_1 \rangle \dots \langle S_{n-1}, e_n \rangle \Rightarrow \langle S_n, v_n \rangle$$

$$\langle S_0, \{l_1=e_1; \dots; l_n=e_n\} \rangle \Rightarrow \langle S_n, \{l_1=v_1; \dots; l_n=v_n\} \rangle$$

Let $r = \text{Ref } 0 \text{ In}$
 $\{a=!r; b=r:=3; c=!r\}$.

Simply Object

Let obj =
 { wins = Ref 0;
 losses = Ref 0;
 getTotal = Function this → !this.wins + !this.losses
 }

$\{a=S; b=True\}.a$

In

obj.getTotal obj obj.getTotal()

Object Extension

Let obj = ... In
 Let obj2 = { wins = obj.wins;
 losses = obj.losses;
 ties = Ref 0;
 getTotal = Function this → !this.ties + obj.getTotal
 }

In

obj2.getTotal obj2

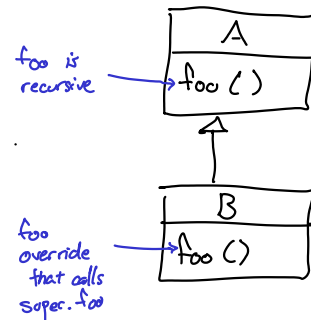
C++ → B.foo → A.foo
 ↙ virtual ↘

Java → B.foo A.foo
 Python
 ↙ ↘
 Dynamic Dispatch

static dispatch

obj
 this

dynamic dispatch



Classes

In a sense, a class is an object factory.

Let counter = {

new = Function this →

{ wins = Ref 0;

losses = Ref 0;

getTotal = Function this → ! this.wins + ! this.losses

}

}

In

Let obj = counter.new counter In

class Account:

nextId = 0

def __init__(self, username):

self.username = username

self.idnum = nextId

nextId += 1

the class itself