Types in languages: Smalltalk

Amer Diwan

What is Smalltalk

• Very different from Modula-3 and Java
  – Everything is an object (including classes and control structures)
  – All type checking happens at run time
• Similar to Modula-3 and Java
  – Garbage collection
  – Safe from unchecked type errors
Run-time “type compatibility”

- Defined by the set of methods that may be invoked
- Consider invoking method \( m \) of object \( o \)
  - If \( o \) has an \( m \) method, then it is invoked
  - Otherwise a “method not understood” message
- A “method not understood” is a kind of a type failure at run time
An example

```
c1 ← C1 new
c2 ← C2 new
c3 ← oc3 new
oc3 ← C3 new
oc5 ← C5 new
t ← oc3 or oc5
t f
t g
t h
t i
```

Classes are objects too

- Members in classes are like static fields
- Methods in classes are available to all instances
  - class Point
    : class members
    allPoints
    : class methods
    newPoint |t|
    t ← Point new
    allPoints add: newPoint
    ^t
Metaclasses: the class of a class

• If classes are objects too, what is the class of a class?
  – It's metaclass
  – Provides a reflection mechanism
    • A program can inspect its own types and manipulate them

Discussion: how does it compare to Java etc.?

• Advantages?
• Disadvantages?
Next lecture: Polymorphism

• Polymorphism:
  – Kinds of polymorphism
• Reading: Cardelli and Wegner (follow link on syllabus page)
  – A dense paper: give yourself some time!