# Lecture 17: More Inheritance

CS 51G
Spring 2018
Kim Bruce

# Test Programs

- One today in lab (upstairs in Edmunds 219)
  - Must turn in by 4 p.m.
- Two others available Monday
- No coverage of GUI components
  - Dragging and interacting w/objects
  - Designing classes
  - Animations

# Midterm

- Friday in class: 50 minutes
- Coverage: Chapters 1-9, 20, and possibly 16.

# More Example

- FallingObjectsWithTomatoes?
  - http://www.cs.pomona.edu/classes/cs051G/demos/FallingObjectsWithTomatoes/FallingObjectsWithTomatoes.grace
  - Differing behavior when hit the ground
  - Define method hitBottom to specify behavior after fall.
- FallingObject is "abstract class"
  - Only defined to inherit from!

# What is inheritance?

- If (child) object inherit from parent (*fresh object*)
    - Child gets all features (defs/vars/method) from parent
    - Add new features specified in child
    - Can “override” inherited features by new ones.
        - If need to use overridden method in defining new one, can give it an alias to use
    - Once all features in place, run initialization code in child object

# What about Types

- Can extend types by adding features to existing types.
- Simpler than inheritance:
  - type A = {m1(...) -> T1,...,mn(...): Tn}
  - type B = A & type {n1(...) -> U1,...}
  - Now B has everything in A plus new ones listed.

# Objectdraw Dialect

```
type Graphic = {
    location -> Point
    x -> Number
    y -> Number
    removeFromCanvas -> Done
    contains(location: Point) -> Boolean
    ...
}
```

# Objectdraw Dialect

```
type Graphic2D = Graphic & type {
   // dimensions of object
   width -> Number
   height -> Number
   size -> Point

   // Change dimensions of object
   size := (dimensions: Point) -> Done
   width := (width: Number) -> Done
   height := (height: Number) -> Done
}
```

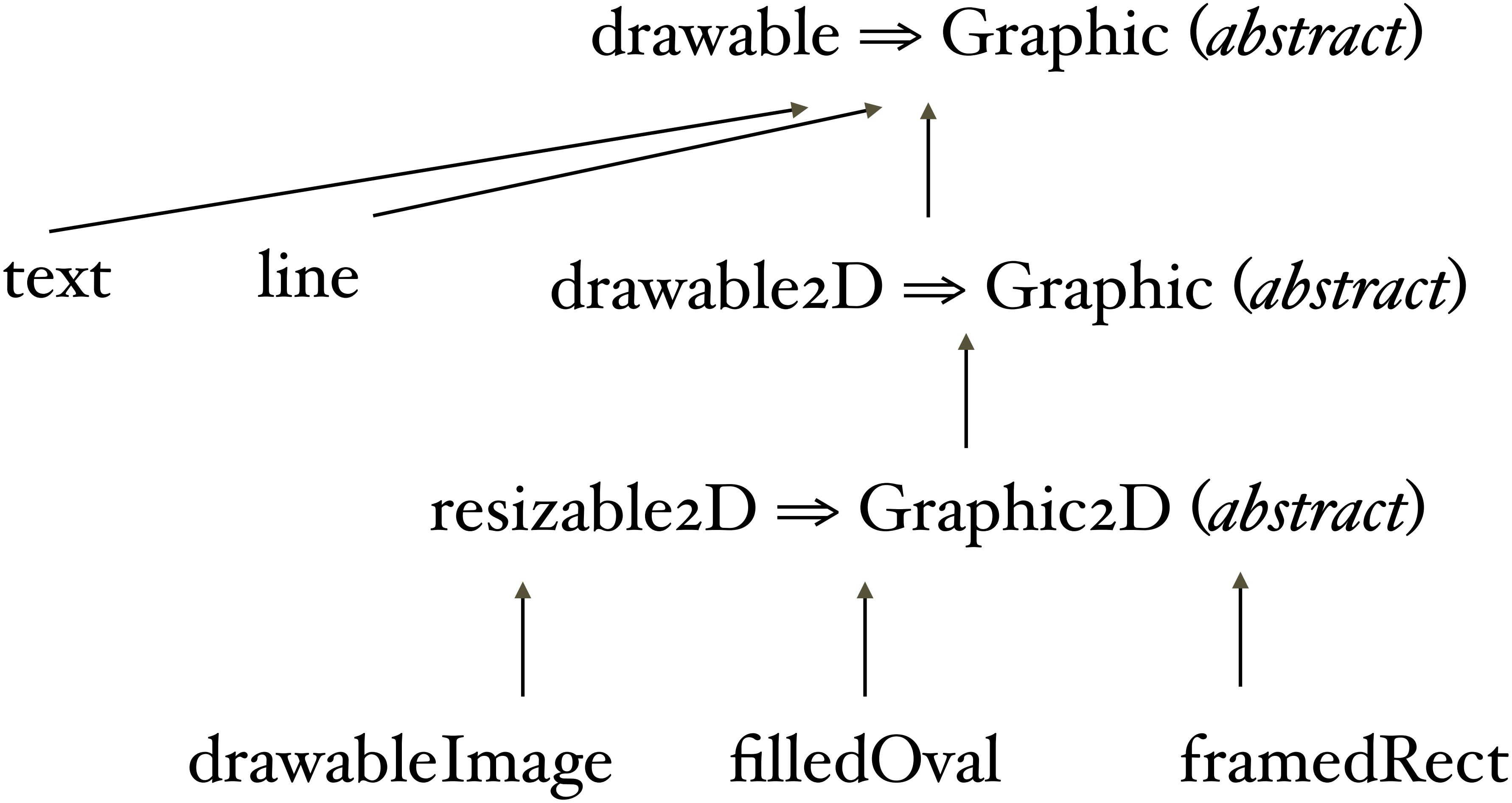
# Objectdraw Dialect

```
type Line = Graphic & type {
  // start and end of line
    start -> Point
    end -> Point

    // set start and end of line
    start := (start': Point) -> Done
    end := (end': Point) -> Done
    setEndPoints (start': Point, end': Point) -> Done

}
```

# Inheritance in Objectdraw



drawable ⇒ Graphic (*abstract*)

text

line

drawable2D ⇒ Graphic (*abstract*)

resizable2D ⇒ Graphic2D (*abstract*)

drawableImage

filledOval

framedRect

# Recursion

- Explain things naturally
- How to draw a target
  - If small enough, just draw bullseye
  - Otherwise draw outer ring and then draw smaller target inside
- Can write programs like that!
  - http://www.cs.pomona.edu/classes/cs051G/demos/TargetApp/TargetApp.grace

# Questions?