Programming Languages and Techniques (CIS120e)

Lecture 32
Nov. 29, 2010

Inner Classes

• Useful in situations where two objects require “deep access” to each other’s internals
• Replaces tangled workarounds like “owner object” (as in the drawing example)
  – Solution with inner classes is easier to read
  – No need to allow public access to instance variables of outer class

• Also called “dynamic nested classes”

Example

Key idea: Classes can be members of other classes...

```java
public class Outer {
    public int outerVar;
    public Outer () {
        outerVar = 6;
    }

    public class Inner {
        public int innerVar;
        public Inner (int z) {
            innerVar = outerVar + z;
        }
    }
}
```

--

```java
... Outer o = new Outer();
Outer.Inner i = o.new Inner();...
```

Scope

• Inner classes can refer to the instance variables and methods of the outer class

The type of inner-class instances

Reference from inner class to variable bound in outer class
### Anonymous Inner Classes

- Remember OCaml’s “anonymous functions”?
  ```ocaml
  map (fun x -> x*3) [1;2;3]
  ```

- Java has something similar: **anonymous inner classes**

- Same motivation in both cases:
  - local function definitions (in OCaml) and inner classes (in Java) are often used in only one place
  - ... so it’s silly to give them a name – better just to just directly write the function/class *itself* at the point where it’s needed!
Better...

class AnonExample {
    public AnonExample() {
        final JButton button = new JButton("On/Off");
        
        Timer timer = new Timer(1000, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Color f = button.getBackground();
                Color b = button.getForeground();
                button.setForeground(f);
                button.setBackground(b);
            }
        });
        timer.start();
    }
    ...
}

Points to note...

- Syntax:
  
  new interfacename() {
      method definitions
  }

  - Other forms also available, but this is enough for present purposes

- Local variables (like button here) that are referenced from the inner class must be declared final
  
  - reasons have to do with the fact that inner classes were added to Java late in its development

Back to the demos...

See DrawingExample2.java and friends