



Laboratory of Software Analysis

Lezione 7

Filippo Ricca

ITC-Irst

Istituto per la ricerca
Scientifica e Tecnologica

ricca@itc.it

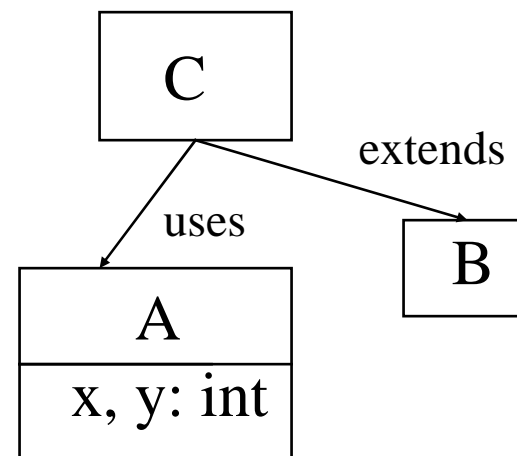
Exercise

Given a Java program build the **Class Diagram** (associations, extends, attributes) of it.

Class A
int x, y

Class B

Class C extend B
A a





Exercise: step by step

1. Give a look at <http://www.graphviz.org/> (see gallery)
2. Using dotty
3. Download the java grammar at <http://www.txl.ca>
4. Test it with small examples (parsing)
5. Implement in TXL the “getClassNames.txl” (print class names)
6. Implement in TXL the “TransformClassNamesInDotty.txl” (implement dot.Grm)
7. Adding to it **associations, extends and attributes.**
8. Test it.

```
redefine program
  ... | [dot_graph]
end redefine

define dot_graph
  'digraph [id] '{
    'graph '['
      'fontsize '= [number]
      'fontname '= [stringlit]
      'fontcolor '= [id]
      'color '= [id]
      'ratio '= [id]
    '];
    [repeat node]
    [repeat edge]
  '}'
end define

define base_node
  [class_name]
end define

define node
  [base_node] [opt labels];
end define
```

```
define edge
  [base_node] '-> [base_node] [opt labels];
end define

define labels
end define

define label
end define
```

DOT GRAMMAR

```
rule buildClassDiagram
```

```
  replace [program]
```

```
    P [program]
```

```
  deconstruct P
```

```
    PH [opt package_header]
```

```
    ID [repeat import_declaration]
```

```
    TD [repeat type_declaration]
```

```
  construct class_nodes [repeat node]
```

```
    using ... TD
```

```
  construct rel_uses [repeat edge]
```

```
    using ... TD
```

```
  construct rel_inheritance [repeat edge]
```

```
    using ... TD
```

```
  by
```

```
    'digraph 'g '{
```

```
      'graph '['
```

```
        'fontsize = 12
```

```
        'fontname = "Times-Roman
```

```
        'fontcolor = black
```

```
        'color = black
```

```
        'ratio = auto
```

```
      '];
```

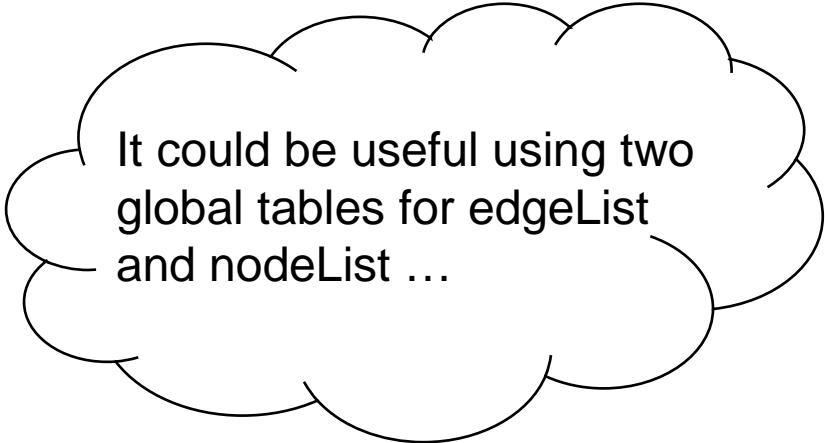
```
      class_nodes
```

```
      rel_uses[. rel_inheritance]
```

```
    }
```

```
  end rule
```

Build Class diagram



It could be useful using two global tables for edgeList and nodeList ...

