

## Plan for Today

### Tree traversals

- depth-first traversal
- pre-order
- post-order

### Visitor Design Pattern

- why?
- what?
- traversing trees with a visitor design pattern

### SableCC's Visitor Design Pattern

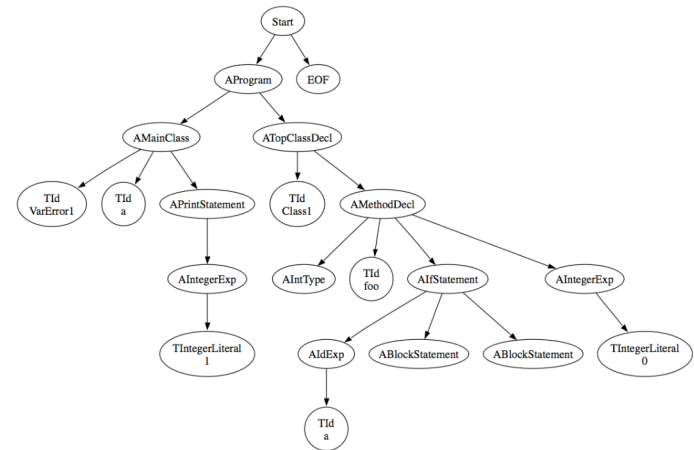
- terminology
- generated code
- using it

CS453 Lecture

Visitor Design Pattern

1

## Example for Tree Traversals



## Visitor Design Pattern

### Situation

- Want to perform some processing on all items in a data structure
- Will be adding many different ways to process items, different features
- Will not be changing the classes of the data structure itself much

### Possibilities

- For each functionality add a method to all of the classes
  - Example of this in PA5, outputDot
  - Each new functionality is spread over multiple files
  - Sometimes can't do it
- Use a large if-then-else statement in visit method
  - pro: keeps all the code for the feature in one place
  - con: can be costly and involve lots of casting
- Visitor design pattern

CS453 Lecture

Visitor Design Pattern

3

## Visitor Design Pattern from PA1

```
public void outputDot( Stm s )
{
    java.io.PrintWriter out = new PrintWriter(System.out);
    DotVisitor dotvisitor = new DotVisitor(out);
    s.accept(dotvisitor);
}
...
class CompoundStm extends Stm {
    Stm stm1, stm2;
    CompoundStm(Stm s1, Stm s2) {stm1=s1; stm2=s2;}
    void accept(Visitor v) {
        v.visitCompoundStm(this);
    }
}
...
// In class DotVisitor
public void visitCompoundStm(CompoundStm s) {
    ...
    s.stm1.accept(this);
    s.stm2.accept(this);
    ...
}
```

CS453 Lecture

Visitor Design Pattern

4

## SableCC Visitor Design Pattern

```
BuildSymTable buildSTvisitor = new BuildSymTable(linesToNodes);
ast.apply(buildSTvisitor);
SymTable globalST = buildSTvisitor.getSymTable();
...
// in class AAndExp
public void apply(Switch sw)
{
    ((Analysis) sw).caseAAndExp(this);    }
...
public void inAAndExp(AAndExp node) { defaultIn(node);    }
public void outAAndExp(AAndExp node) { defaultOut(node);    }

public void caseAAndExp(AAndExp node) {
    inAAndExp(node);
    if(node.getLExp() != null) {    node.getLExp().apply(this);    }
    if(node.getRExp() != null) {    node.getRExp().apply(this);    }
    outAAndExp(node);    }
```

## FAQ

**How do I associate data with a node in the AST if I can't add fields to the node classes?**

**What if I want to do the same thing on each node?**

**What if I only need to do something on certain nodes?**

**Shouldn't the visit/case methods have return values?**