

Διάσχιση AST δένδρου

Από τη γραμματική στις τάξεις (I)

- Για κάθε **τερματικό** και **μη-τερματικό** σύμβολο της γραμματικής, ο SableCC φτιάχνει την αντίστοιχη Java τάξη στον κατάλογο `<package_name>/node`.
- **Tokens**

```
boolean = 'boolean';  
assign = '=' ;  
identifier = letter(letter|digit|'_')*;  
number = digit+;
```



TBoolean.java
TAssign.java
TIdentifier.java
TNumber.java



Χρήσιμες Μέθοδοι:

- toString()
- getLine()
- getPos()



extends Token.java

Από τη γραμματική στις τάξεις (II)

- Κανόνες AST (για κάθε alternative)

```
exp = {and}      [l]:exp [r]:exp  
      | {plus}   [l]:exp [r]:exp  
      | ...  
      | {number} number  
      | {id} identifier;
```



AAndExp.java
APlusExp.java
ANumberExp.java
AIdExp.java



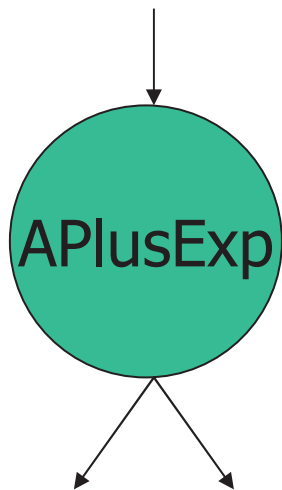
extends **Pexp[.java]**

program = main_class;  AProgram[.java] extends PProgram[.java]

- Μέθοδοι Τάξεων, π.χ. APlusExp.java:
 - public **PExp** getL() { ... }
 - public **PExp** getR() { ... }
- AIdExp.java:
 - public **TIdentifier** getIdentifier() { ... }

Επισκέπτες κόμβων AST δένδρου

- Στον κατάλογο `<package_name>/analysis`
 - `DepthFirstAdapter` extends `AnalysisAdapter`

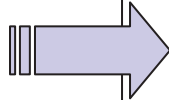


```
...  
public void inAPlusExp (APlusExp node){...}  
  
public void outAPlusExp (APlusExp node){...}  
  
public void caseAPlusExp (APlusExp node){  
    inAPlusExp (node);  
    if(node.getL() != null){  
        node.getL().apply(this);  
    }  
    if(node.getR() != null){  
        node.getR().apply(this);  
    }  
    outAPlusExp (node);  
}  
...
```

Κόμβοι AST δένδρου

- Αρχικός κόμβος: Start

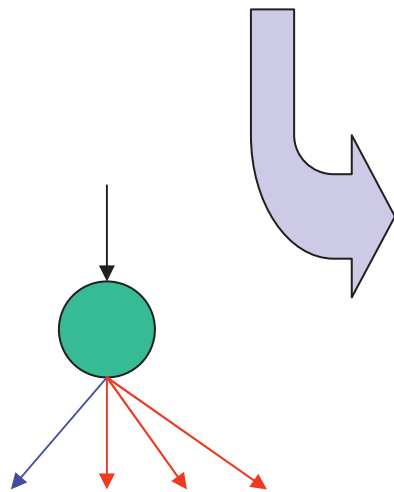
`program = main_class;`



```
public void caseStart(Start node) {  
    inStart(node);  
    node.getPPProgram().apply(this);  
    node.getEOF().apply(this);  
    outStart(node);  
}
```

- Διάσχιση Λίστας

`main_class = identifier statement* ;`



```
public void caseAMainClass(AMainClass node) {  
    inAMainClass(node);  
    if(node.getIdentifier() != null) {  
        node.getIdentifier().apply(this);  
    }  
    {  
        Object temp[] = node.getStatement().toArray();  
        for(int i = 0; i < temp.length; i++) {  
            ((PStatement) temp[i]).apply(this);  
        }  
    }  
    outAMainClass(node);  
}
```

Κατασκευή δικού σας επισκέπτη

- YourVisitor.java
 - extends DepthFirstAdapter
- Μπορείτε να κάνετε **override** όποιες μεθόδους του DepthFirstAdapter θέλετε:
 - στις μεθόδους **inXxxx** και **outXxxx** γράφετε κατευθείαν τον κώδικά σας
 - στις μεθόδους **caseXxxx** θα πρέπει να αντιγράψετε τον κώδικά τους από το αρχείο DepthFirstAdapter.java και να προσθέσετε όπου θέλετε τον κώδικά σας, μιας και στις μεθόδους αυτές περιλαμβάνεται και ο κώδικας διάσχισης των υποδένδρων
- Οι μέθοδοι caseXxxx γίνονται override όταν χρειάζεται να κάνουμε κάτι ανάμεσα στις επισκέψεις των υποδένδρων του κόμβου

Compiler.java

```
import java.io.*;
import package.lexer.Lexer;
import package.parser.Parser;
import package.node.Start;

public class Parser {
    public static void main(String[] args) {
        try {
            Parser parser = new Parser(
                new Lexer(
                    new PushbackReader(
                        new FileReader(args[0].toString()), 1024));
            Start ast = parser.parse();

            ast.apply(new YourVisitor());

        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
```