The Abstract Syntax Tree (AST) is made of object of the class TreeNode. A TreeNode has the following definition. Note that you must use this definition if you want to use my code generator.

```java
public class TreeNode {
    public int lineNumber; //Line in program where this construct is found
    public int nValue; //Numerical value of a number
    public String sValue; //Lexeme or string value of an identifier
    public int nodeType; //PROGRAM, DECLARATION, etc.
    public int typeSpecifier; //VOID or INT
    public String rename; //Used by the Semantic Analyzer
    public boolean visited; //Initialized to false, used for traversals

    public TreeNode C1; //Pointer to Child 1
    public TreeNode C2; //Pointer to Child 2
    public TreeNode C3; //Pointer to Child 3
    public TreeNode sibling; //Pointer to Sibling
}
```

The structure of the tree is the following. The root of the tree is a PROGRAM node. The sibling field of the PROGRAM node is a linked list of three types of nodes: a VARIABLE node, an ARRAY node (where the size of the array is stored in the nValue field), or a FUNCTION node.

For a FUNCTION node, the C1 field points to a PARAMETER_LIST node and the C2 field points to a COMPOUND node.

A PARAMETER_LIST node either has a typeSpecifier field of VOID or the sibling field points to a linked list of VARIABLE and/or ARRAY nodes.

For a COMPOUND node, the C1 field points to a DECLARATION node and the C2 field points to a STATEMENT_LIST node.

A DECLARATION node is a sibling field linked list of either VARIABLE or ARRAY nodes. Of course, the sibling field could be null if there are no declarations.

A STATEMENT_LIST node is a sibling linked list of several types of nodes. The nodes in the list may represent an expression, or a COMPOUND node, or an IF statement, a WHILE statement, a RETURN statement, a READ statement, a WRITE statement, or a CALL statement.

For the various types of expression nodes, the nodeType contains an arithmetic or a relational operator, C1 points to the left side of the expression, which may be a VARIABLE, ARRAY, or another expression, and C2 points to the right side of the expression.

When an array is being dereferenced, that is a particular array location is being used as a variable—for example x[2]—then the C1 field of the ARRAY node points to the expression that evaluates to the array location.
For an IF statement node, C1 points to the test expression, C2 points to the true statement, and C3 points to the false statement. Both the true and false statements may be COMPOUND nodes.

For a WHILE statement node, C1 points to the test expression and C2 points to the statement list.

For a RETURN node, C1 points to expression to be returned.

For a READ node, C1 points to an ARRAY or VARIABLE node that is receiving the value being read.

For a WRITE node, C1 points to the expression to be written.

For a CALL node, the sValue field contains the name of the function being called, the typeSpecifier field contains INT or VOID depending of the return type of the function, and C1 points to an ARGUMENTS node.

An ARGUMENTS node is a sibling link list of NUMBER, VARIABLE, or ARRAY nodes that are the arguments being passed to the function.

As an example, the program below is essentially the Selection Sort program from pages 496-497 of the text book. It has been changed to accommodate the read and write statements. Below the program is a copy of the syntax tree that is produced from this program. Most of the constructs mentioned above appear in this syntax tree.

In the printout of the syntax tree, the first item identifies which pointer is represented: Sibling, C1, C2, or C3. If a pointer is null, however, it is not shown. The Node Type is the nodeType field, the Name is the sValue field, the Value is the nValue field, and the Data Type is the typeSpecifier field. The Rename field is so that every variable can have a unique name. If you look carefully, you will see a change that I made in the Selection Sort program. In the function sort() there are two variables called t. However, they are in different scopes, so they are actually different variables. If you find them in the syntax tree, you will see that each has a unique rename value: one is tmpVar0000012 and the other is tmpVar0000013.
/***********************************************
*  A program to perform selection sort on a 10
*  element array.
*
***********************************************/

int x[ 10 ];

int minloc ( int a[], int low, int high ) {
    int i; int x; int k;

    k = low;
    x = a[ low ];
    i = low + 1;

    while( i < high ) {  /* do swap */
        if( a[ i ] < x ) {
            x = a[ i ];
            k = i;
        }
        i = i + 1;
    }

    return k;
}

void sort( int a[], int low, int high ) {
    int i; int k; int t;

    i = low;
    while( i < high - 1 ) {
        int t;
        k = minloc( a, i, high );
        t = a[ k ];
        a[ k ] = a[ i ];
        a[ i ] = t;
        i = i + 1;
    }
}

void main( void ) {
    int i;
    i = 0;

    while( i < 10 ) {
        read x[ i ];
    }
}
i = i + 1;
}

sort( x, 0, 10 );
i = 0;
while( i < 10 ) {
    write( x[ i ] );
    i = i + 1;
}
}

Node Type: Program
Line Number: 9
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: Array
Line Number: 9
Name: x
Value: 10
Data Type: int
Rename: tmpVar000000

Sibling
Node Type: Function
Line Number: 11
Name: minloc
Value: 0
Data Type: int
Rename: null

C1
Node Type: Parameter List
Line Number: 11
Name: null
Value: 0
Data Type: unknown
Rename: unknown

Sibling
Node Type: Array
Line Number: 11
Name: a
Value: 0
Data Type: int
Rename: tmpVar000001
Sibling
Node Type: Variable
Line Number: 11
Name: low
Value: 0
Data Type: int
Rename: tmpVar000002

Sibling
Node Type: Variable
Line Number: 11
Name: high
Value: 0
Data Type: int
Rename: tmpVar000003

C2
Node Type: Compound Statement
Line Number: 11
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 12
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: Variable
Line Number: 12
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004

Sibling
Node Type: Variable
Line Number: 12
Name: x
Value: 0
Data Type: int
Rename: tmpVar000005

Sibling
Node Type: Variable
Line Number: 12
Name: k
Value: 0
Data Type: int
Rename: tmpVar000006
C2
Node Type: Statement List
Line Number: 14
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: =
Line Number: 14
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 14
Name: k
Value: 0
Data Type: int
Rename: tmpVar000006

C2
Node Type: Variable
Line Number: 14
Name: low
Value: 0
Data Type: int
Rename: tmpVar000002

Sibling
Node Type: =
Line Number: 15
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 15
Name: x
Value: 0
Data Type: int
Rename: tmpVar000005

C2
Node Type: Array
Line Number: 15
Name: a
Value: 0
Data Type: int
Rename: tmpVar000001
C1
Node Type: <
Line Number: 18
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 18
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004

C2
Node Type: Variable
Line Number: 18
Name: high
Value: 0
Data Type: int
Rename: tmpVar000003

C2
Node Type: Compound Statement
Line Number: 18
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 19
Name: null
Value: 0
Data Type: unknown
Rename: null

C2
Node Type: Statement List
Line Number: 19
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: if
Line Number: 19
Name: null
Value: 0
Data Type: unknown
Rename: null
C1
Node Type: <
Line Number: 19
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Array
Line Number: 19
Name: a
Value: 0
Data Type: int
Rename: tmpVar000001

C1
Node Type: Variable
Line Number: 19
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004

C2
Node Type: Variable
Line Number: 19
Name: x
Value: 0
Data Type: int
Rename: tmpVar000005

C2
Node Type: Compound Statement
Line Number: 19
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 20
Name: null
Value: 0
Data Type: unknown
Rename: null

C2
Node Type: Statement List
Line Number: 20
Name: null
Value: 0
Data Type: unknown
Rename: null
Sibling
Node Type: =
Line Number: 20
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 20
Name: x
Value: 0
Data Type: int
Rename: tmpVar000005

C2
Node Type: Array
Line Number: 20
Name: a
Value: 0
Data Type: int
Rename: tmpVar000001

C1
Node Type: Variable
Line Number: 20
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004

Sibling
Node Type: =
Line Number: 21
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 21
Name: k
Value: 0
Data Type: int
Rename: tmpVar000006

C2
Node Type: Variable
Line Number: 21
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004
Sibling
Node Type: =
Line Number: 23
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 23
Name: i
Value: 0
Data Type: int
Rename: tmpVar000004

C2
Node Type: +
Line Number: 23
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: return
Line Number: 26
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 26
Name: k
Value: 0
Data Type: int
Rename: tmpVar000006
Sibling
Node Type: Function
Line Number: 29
Name: sort
Value: 0
Data Type: void
Rename: null

C1
Node Type: Parameter List
Line Number: 29
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: Array
Line Number: 29
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007

Sibling
Node Type: Variable
Line Number: 29
Name: low
Value: 0
Data Type: int
Rename: tmpVar000008

Sibling
Node Type: Variable
Line Number: 29
Name: high
Value: 0
Data Type: int
Rename: tmpVar000009

C2
Node Type: Compound Statement
Line Number: 29
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 30
Name: null
Value: 0
Data Type: unknown
Rename: null
Sibling
Node Type: Variable
Line Number: 30
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

Sibling
Node Type: Variable
Line Number: 30
Name: k
Value: 0
Data Type: int
Rename: tmpVar0000011

Sibling
Node Type: Variable
Line Number: 30
Name: t
Value: 0
Data Type: int
Rename: tmpVar0000012

C2
Node Type: Statement List
Line Number: 32
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: =
Line Number: 32
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 32
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

C2
Node Type: Variable
Line Number: 32
Name: low
Value: 0
Data Type: int
Rename: tmpVar0000008
Sibling
Node Type: while
Line Number: 33
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: <
Line Number: 33
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 33
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

C2
Node Type: -
Line Number: 33
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 33
Name: high
Value: 0
Data Type: int
Rename: tmpVar000009

C2
Node Type: a number
Line Number: 33
Name: null
Value: 1
Data Type: int
Rename: null

C2
Node Type: Compound Statement
Line Number: 33
Name: null
Value: 0
Data Type: unknown
Rename: null
C1  
Node Type: Declaration  
Line Number: 34  
Name: null  
Value: 0  
Data Type: unknown  
Rename: null

Sibling  
Node Type: Variable  
Line Number: 34  
Name: t  
Value: 0  
Data Type: int  
Rename: tmpVar0000013

C2  
Node Type: Statement List  
Line Number: 35  
Name: null  
Value: 0  
Data Type: unknown  
Rename: null

Sibling  
Node Type: =  
Line Number: 35  
Name: null  
Value: 0  
Data Type: unknown  
Rename: null

C1  
Node Type: Variable  
Line Number: 35  
Name: k  
Value: 0  
Data Type: int  
Rename: tmpVar0000011

C2  
Node Type: Call  
Line Number: 35  
Name: minloc  
Value: 0  
Data Type: int  
Rename: null

C1  
Node Type: Arguments  
Line Number: 35  
Name: null  
Value: 0  
Data Type: unknown  
Rename: null
Sibling
Node Type: Array
Line Number: 35
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007

Sibling
Node Type: Variable
Line Number: 35
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

Sibling
Node Type: Variable
Line Number: 35
Name: high
Value: 0
Data Type: int
Rename: tmpVar000009

Sibling
Node Type: =
Line Number: 36
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 36
Name: t
Value: 0
Data Type: int
Rename: tmpVar0000013

C2
Node Type: Array
Line Number: 36
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007

C1
Node Type: Variable
Line Number: 36
Name: k
Value: 0
Data Type: int
Rename: tmpVar0000011
Sibling
Node Type: =
Line Number: 37
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Array
Line Number: 37
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007

C1
Node Type: Variable
Line Number: 37
Name: k
Value: 0
Data Type: int
Rename: tmpVar0000011

C2
Node Type: Array
Line Number: 37
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007

C1
Node Type: Variable
Line Number: 37
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

Sibling
Node Type: =
Line Number: 38
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Array
Line Number: 38
Name: a
Value: 0
Data Type: int
Rename: tmpVar000007
C1
Node Type: Variable
Line Number: 38
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

C2
Node Type: Variable
Line Number: 38
Name: t
Value: 0
Data Type: int
Rename: tmpVar0000013

Sibling
Node Type: =
Line Number: 39
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 39
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

C2
Node Type: +
Line Number: 39
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 39
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000010

C2
Node Type: a number
Line Number: 39
Name: null
Value: 1
Data Type: int
Rename: null
Sibling
Node Type: Function
Line Number: 43
Name: main
Value: 0
Data Type: void
Rename: null

C1
Node Type: Parameter List
Line Number: 43
Name: null
Value: 0
Data Type: void
Rename: null

C2
Node Type: Compound Statement
Line Number: 43
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 44
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: Variable
Line Number: 44
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: Statement List
Line Number: 45
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: =
Line Number: 45
Name: null
Value: 0
Data Type: unknown
Rename: null
C1
Node Type: Variable
Line Number: 45
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: a number
Line Number: 45
Name: null
Value: 0
Data Type: int
Rename: null

Sibling
Node Type: while
Line Number: 47
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: <
Line Number: 47
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 47
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: a number
Line Number: 47
Name: null
Value: 10
Data Type: int
Rename: null

C2
Node Type: Compound Statement
Line Number: 47
Name: null
Value: 0
Data Type: unknown
Rename: null
C1
Node Type: Declaration
Line Number: 48
Name: null
Value: 0
Data Type: unknown
Rename: null

C2
Node Type: Statement List
Line Number: 48
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: read
Line Number: 48
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Array
Line Number: 48
Name: x
Value: 0
Data Type: int
Rename: tmpVar000000

C1
Node Type: Variable
Line Number: 48
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

Sibling
Node Type: =
Line Number: 49
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 49
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014
C2
Node Type: +
Line Number: 49
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 49
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

c2
Node Type: a number
Line Number: 49
Name: null
Value: 1
Data Type: int
Rename: null

Sibling
Node Type: Call
Line Number: 52
Name: sort
Value: 0
Data Type: void
Rename: null

C1
Node Type: Arguments
Line Number: 52
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: Array
Line Number: 52
Name: x
Value: 0
Data Type: int
Rename: tmpVar000000

Sibling
Node Type: a number
Line Number: 52
Name: null
Value: 0
Data Type: int
Rename: null
Sibling
Node Type: a number
Line Number: 52
Name: null
Value: 10
Data Type: int
Rename: null

Sibling
Node Type: =
Line Number: 54
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 54
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: a number
Line Number: 54
Name: null
Value: 0
Data Type: int
Rename: null

Sibling
Node Type: while
Line Number: 55
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: <
Line Number: 55
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 55
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014
C2
Node Type: a number
Line Number: 55
Name: null
Value: 10
Data Type: int
Rename: null

C2
Node Type: Compound Statement
Line Number: 55
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Declaration
Line Number: 56
Name: null
Value: 0
Data Type: unknown
Rename: null

C2
Node Type: Statement List
Line Number: 56
Name: null
Value: 0
Data Type: unknown
Rename: null

Sibling
Node Type: write
Line Number: 56
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Array
Line Number: 56
Name: x
Value: 0
Data Type: int
Rename: tmpVar000000

C1
Node Type: Variable
Line Number: 56
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014
Sibling
Node Type: =
Line Number: 57
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 57
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: +
Line Number: 57
Name: null
Value: 0
Data Type: unknown
Rename: null

C1
Node Type: Variable
Line Number: 57
Name: i
Value: 0
Data Type: int
Rename: tmpVar0000014

C2
Node Type: a number
Line Number: 57
Name: null
Value: 1
Data Type: int
Rename: null