program

class Program { declaration declaration declaration declaration declaration declaration declaration declaration declaration declaration declaration }

structDeclaration

struct A { varDeclaration }

varType a ;

int

structDeclaration

struct B { varDeclaration varDeclaration }

varType b [5] ;

int

varType c ;

struct A

varDeclaration

varType y ;

struct A

varDeclaration

varType z ;

struct A

varDeclaration

varType m ;

struct B

methodDeclaration

methodType error ( ) block

void { varDeclaration statement statement statement statement statement statement statement statement }

varType c ;

char

assignStmt

location = expression

m . location

b [expression]

literal

int_literal

10

literal

int_literal

5

; assignStmt

location = expression

b . location

c . location

a

literal

char_literal

'c'

; ifStmt

if (expression) block else block

literal

int_literal

1

{ statement statement }

assignStmt

location = expression

a literal

int_literal

4

;

{ statement statement }

assignStmt

location = expression

m literal

int_literal

3

;

assignStmt

location = expression

c methodCall

factorial (arg)

expression

location

c

; returnStmt

return expression ;

location

m

methodDeclaration

methodType factorial (parameter) block

int parameterType n

int

{ statement }

ifStmt

if (expression) block else block

expression eq_op expression

location

n

== literal

int_literal

0

{ statement }

returnStmt

return expression ;

literal

int_literal

1

{ statement }

returnStmt

return expression ;

expression higher_arith_op expression

location

n

* methodCall

factorial (arg)

expression

expression arith_op expression

location

n

- literal

int_literal

1

methodDeclaration

methodType OutputInt (parameter) block

void parameterType n

int

{ }

methodDeclaration

methodType InputInt ( ) block

int { statement }

returnStmt

return expression ;

literal

int_literal

0

methodDeclaration

methodType ReturnNumber ( ) block

int { statement }

returnStmt

return expression ;

location

z . location

a

methodDeclaration

methodType main ( ) block

void { varDeclaration varDeclaration varDeclaration varDeclaration statement statement statement statement statement statement statement }

varType y [5] ;

struct B

varType i ;

int

varType j ;

int

varType k ;

int

assignStmt

location = expression

i literal

int_literal

0

; assignStmt

location = expression

j literal

int_literal

0

; assignStmt

location = expression

z . location

a

literal

int_literal

3

; whileStmt

while (expression) block

expression rel_op expression

location

i

<= literal

int_literal

10

{ statement statement statement statement statement statement statement statement }

assignStmt

location = expression

y [expression] . location

location

j

b [expression]

literal

int_literal

0

methodCall

InputInt ( )

; ifStmt

if (expression) block

expression eq_op expression

location

y [expression] . location

location

j

b [expression]

literal

int_literal

0

== literal

int_literal

5

{ statement statement statement statement statement }

assignStmt

location = expression

y [expression] . location

location

j

b [expression]

literal

int_literal

0

location

z . location

a

; assignStmt

location = expression

k methodCall

factorial (arg)

expression

methodCall

ReturnNumber ( )

; methodCall ;

OutputInt (arg)

expression

location

k

assignStmt

location = expression

y [expression] . location

location

j

c . location

a

methodCall

factorial (arg)

expression

location

y [expression] . location

location

j

b [expression]

literal

int_literal

0

; methodCall ;

OutputInt (arg)

expression

location

y [expression] . location

location

j

c . location

a

assignStmt

location = expression

i expression arith_op expression

location

i

+ literal

int_literal

1

;