```java
class ErrorExample {
    int x
    float y = 2.5;

    void main() {
        int a = 5
        if a > 3) {
            print(a);
        } else {
            print("Error");

        while (a > 0 {
            a = a - 1;
        }

        for (int i = 0; i < 10; i++) {
            if (i % 2 == 0 {
                print(i);
            }
        }
    }

    int calculate(int x int y) {
        return x * y;
    }
}
```

Salida esperada para inputErrors.txt

```
-- class
--- IDENTIFIER: ErrorExample
---- {
----- RULE_TYPE
----- TYPE: int
--- IDENTIFIER: x
Syntax Error 3 at line 2, token: float
---- ;
Recovered: Found opening brace after missing semicolon
----- RULE_TYPE
----- TYPE: float
--- IDENTIFIER: y
--- =
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 2.5
-- ;
----- RULE_METHODS
----- RULE_TYPE
----- TYPE: void
----- IDENTIFIER: main
----- (
------ RULE_PARAMS
----- )
----- {
-- RULE_BODY
--- RULE_VARIABLE
----- RULE_TYPE
----- TYPE: int
--- IDENTIFIER: a
--- =
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 5
Syntax Error 16 at line 6, token: if
-- ;
Recovered: Found opening parenthesis after missing semicolon
--- RULE_IF
--- if
```

Syntax Error 403 at line 6, token: a
Recovered: Missing opening parenthesis in if condition
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: a
------ >
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 3
---)
---- RULE_STATEMENT_BLOCK
---- {
-- RULE_BODY
--- RULE_CALL_METHOD
--- IDENTIFIER: print
--- (
---- RULE_PARAM_VALUES
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: a
---)
-- ;
---- }
--- else
---- RULE_STATEMENT_BLOCK
---- {
-- RULE_BODY
--- RULE_CALL_METHOD
--- IDENTIFIER: print
--- (
---- RULE_PARAM_VALUES
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: "Error"
---)
-- ;
Syntax Error 53 at line 12, token: while

```
---- }
Recovered: Found closing brace for statement block

--- RULE_WHILE
--- while
--- (
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: a
------ >
------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 0
Syntax Error 29 at line 12, token: {
Recovered: Missing closing parenthesis in while condition
---- RULE_STATEMENT_BLOCK
---- {
-- RULE_BODY
--- RULE_ASSIGNMENT
--- IDENTIFIER: a
--- =
--- RULE_EXPRESSION
------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: a
------- -
------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 1
-- ;
---- }

--- RULE_FOR
--- for
--- (
--- RULE_VARIABLE
----- RULE_TYPE
----- TYPE: int
--- IDENTIFIER: i
```

```
--- =
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 0
-- ;
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: i
------ <
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 10
-- ;
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: i
------ ++
--- )
---- RULE_STATEMENT_BLOCK
---- {
-- RULE_BODY
--- RULE_IF
--- if
--- (
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: i
------ %
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- LITERAL: 2
------ ==
-------- RULE_A
--------- RULE_B
```

---------- RULE_C
---------- LITERAL: 0
Syntax Error 402 at line 16, token: {
Recovered: Missing closing parenthesis in if condition
---- RULE_STATEMENT_BLOCK
---- {
-- RULE_BODY
--- RULE_CALL_METHOD
--- IDENTIFIER: print
--- (
---- RULE_PARAM_VALUES
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B
---------- RULE_C
---------- IDENTIFIER: i
---)
-- ;
---- }
---- }
----- }

----- RULE_METHODS
----- RULE_TYPE
----- TYPE: int
----- IDENTIFIER: calculate
----- (
------ RULE_PARAMS
----- RULE_TYPE
----- TYPE: int
------ IDENTIFIER: x
Syntax Error 15 at line 22, token: int
Recovered: Missing comma in parameter list
----- RULE_TYPE
----- TYPE: int
------ IDENTIFIER: y
-----)
----- {
-- RULE_BODY
--- RULE_RETURN
--- return
--- RULE_EXPRESSION
-------- RULE_A
--------- RULE_B

---------- RULE_C
---------- IDENTIFIER: x
-------- *
-------- RULE_A
---------- RULE_B
---------- RULE_C
---------- IDENTIFIER: y
-- ;
----- }
---- }
Parsing completed with recovery.
Total errors found: 7