```
<program> ::= <statement> <program_tail>
<program_tail> ::= <statement> <program_tail> | epsilon
<statement> ::= <listener_definition> | <trigger_call> | <if_statement> | <for_loop> |
<input_operation> | <send_command>
<listener_definition> ::= "on" "(" <string> "," <lambda_expr> "," <lambda_expr> ")"
<trigger_call> ::= <string> "->" <expression>
<if_statement> ::= "if" <expression> ":" <block> "end"
<for_loop> ::= "for" <identifier> "in" <expression> ":" <block> "end"
<block> ::= <statement> <block_tail>
<block_tail> ::= <statement> <block_tail> | epsilon
<lambda_expr> ::= "lambda" <identifier> ":" <lambda_body> "end"
<lambda_body> ::= <expression> | <statement> <block_tail>
<input_operation> ::= <identifier> "=" "input" "(" ")"
<send_command> ::= "send_command" "(" <string> "," <string> ")"
<expression> ::= <boolean_expr> | <arithmetic_expr> | <function_call> | <string> | <identifier> |
<number>
<boolean_expr> ::= <expression> <boolean_op> <expression> | "True" | "False"
<boolean_op> ::= "and" | "or" | "not" | "==" | "!=" | "<" | ">" | "<=" | ">="
<arithmetic_expr> ::= <expression> <arith_op> <expression>
<arith_op> ::= "+" | "-" | "*" | "/" | "%"
<function_call> ::= <identifier> "(" <argument_list> ")"
<argument_list> ::= <expression> <argument_list_tail> | epsilon
<argument_list_tail> ::= "," <expression> <argument_list_tail> | epsilon
<string> ::= '"' <string_characters> '"'
<string_characters> ::= <string_character> <string_characters> | epsilon
<string_character> ::= <letter> | <digit>
<identifier> ::= <letter> <identifier_tail>
<identifier_tail> ::= <letter_or_digit_or_underscore> <identifier_tail> | epsilon
<letter_or_digit_or_underscore> ::= <letter> | <digit> | "_"
<letter> ::= "a" | … | "z" | "A" | … | "Z"
<digit> ::= "0" | … | "9"
<number> ::= <digit> <number_tail>
<number_tail> ::= <digit> <number_tail> | epsilon
```