program   → funDecl* EOF ;

funDecl   → "fun" IDENTIFIER "("
                  (IDENTIFIER ":" type ("," IDENTIFIER ":" type)* ","?)?
                  ")" ":" type block ;

statement   → varDecl
           | exprStmt
           | ifStmt
           | putStmt
           | returnStmt
           | whileStmt
           | block ;

varDecl   → "var" IDENTIFIER ":" type "=" expression ";" ;

exprStmt   → expression ";" ;
ifStmt   → "if" "(" expression ")" statement
           ("else" statement)? ;
putStmt   → "put" expression ";" ;
returnStmt   → "return" expression? ";" ;
whileStmt   → "while" "(" expression ")" statement ;
block   → "{" statement* "}" ;

expression   → assignment ;

assignment   → IDENTIFIER "=" assignment
           | disjunction ;

disjunction → conjunction ("or" conjunction)* ;
conjunction → equality ("and" equality)* ;
equality   → comparison (("!=" | "==") comparison)* ;
comparison   → term ((">" | ">=" | "<" | "<=") term)* ;
term   → factor (("-" | "+") factor)* ;
factor   → unary ("*" unary)* ;

unary   → ("!" | "-") unary | call | primary ;
call   → IDENTIFIER "(" (expression (","expression)* ","?)? ")" ;
primary   → TRUE | FALSE | GET
           | NUMBER | IDENTIFIER | "(" expression ")"

```
type              → "Int"
                  | "Bool"
                  | "Void"
```