

15조-코드 난독화 도구 제작

ANTLR & Listener

종합설계1

컴퓨터융합학부
조인우 | 손지웅 | 노형우
지도 교수 : 조은선

목차 LIST

01 ANTLR

02 Listener

03 구현 과정

01 ANTLR



- 문법 기반 파서 자동 생성 도구
- 컴파일러, 언어 처리기 제작 등에 활용

01 ANTLR



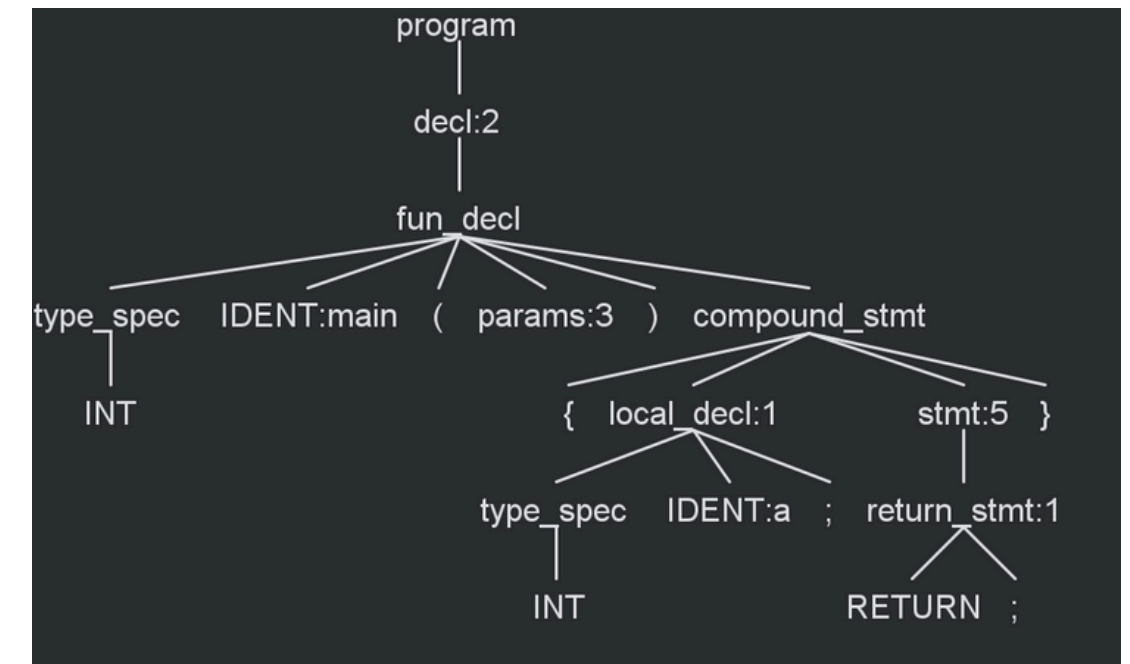
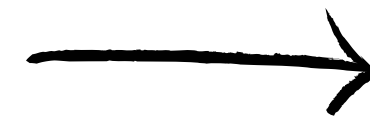
ANTLR

```
program : decl*  
decl   : var_decl  
       | fun_decl ;  
var_decl : type_spec IDENT ';' ;  
         | type_spec IDENT '=' LITERAL ';' ;  
         | type_spec IDENT '[' LITERAL ']' ';' ;  
type_spec : VOID  
         | INT ;  
fun_decl : type_spec IDENT '(' params ')' compound_stmt ;  
params   : param (',' param)*  
         | VOID ;  
param    : type_spec IDENT  
         | type_spec IDENT '[' ']' ;  
stmt     : expr_stmt  
         | compound_stmt  
         | if_stmt  
         | while_stmt  
         | return_stmt ;  
expr_stmt : expr ';' ;  
while_stmt : WHILE '(' expr ')' stmt ;  
compound_stmt : '{' local_decl* stmt* '}' ;  
local_decl  : type_spec IDENT ';' ;  
             | type_spec IDENT '=' LITERAL ';' ;  
             | type_spec IDENT '[' LITERAL ']' ';' ;  
if_stmt     : IF '(' expr ')' stmt  
             | IF '(' expr ')' stmt ELSE stmt ;  
return_stmt : RETURN ';' ;  
             | RETURN expr ';' ;  
expr       : LITERAL
```

grammar.g4

```
int main()  
{  
    int a;  
  
    return;  
}
```

test.c



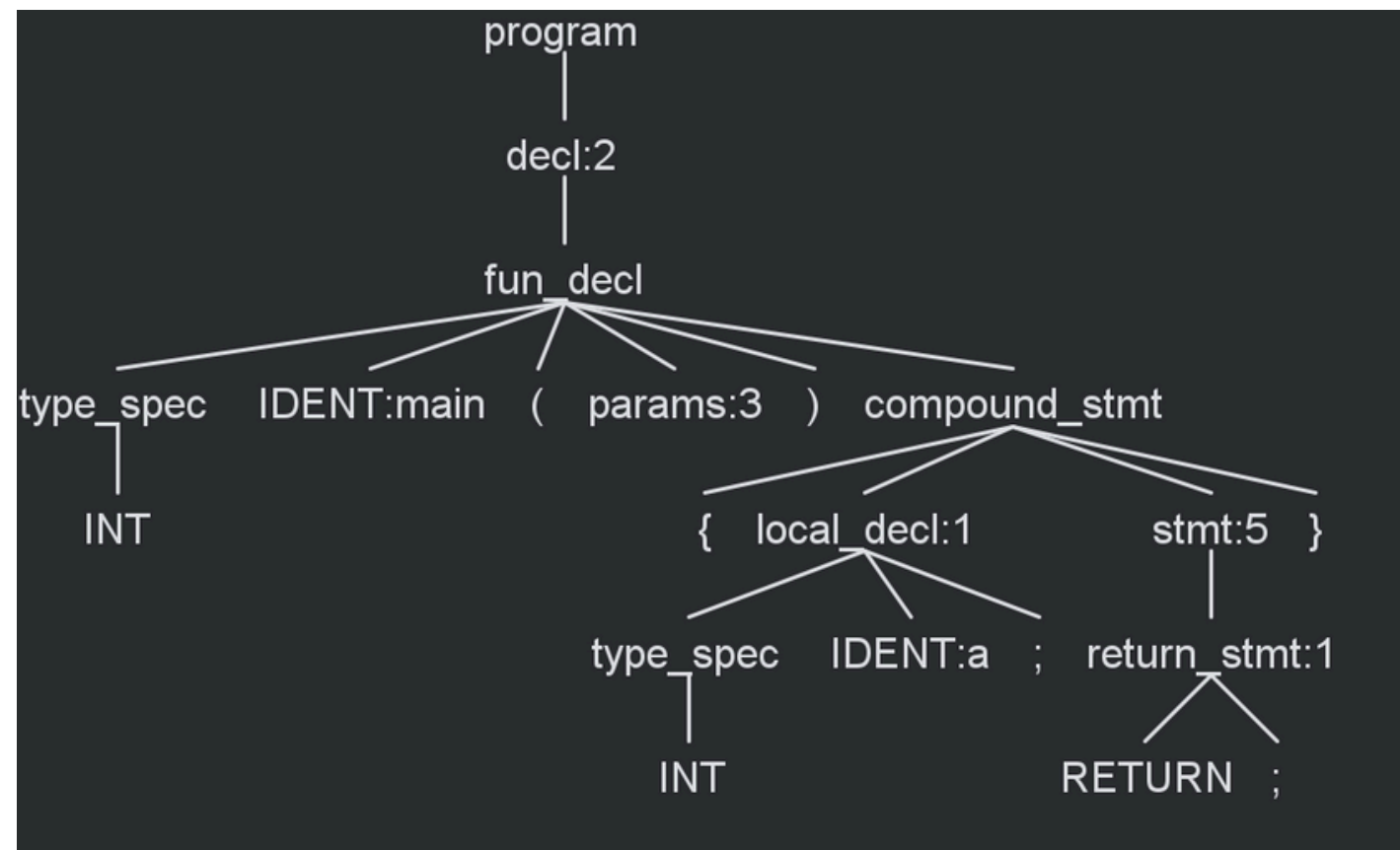
파스 트리(Parse Tree)

02 Listener

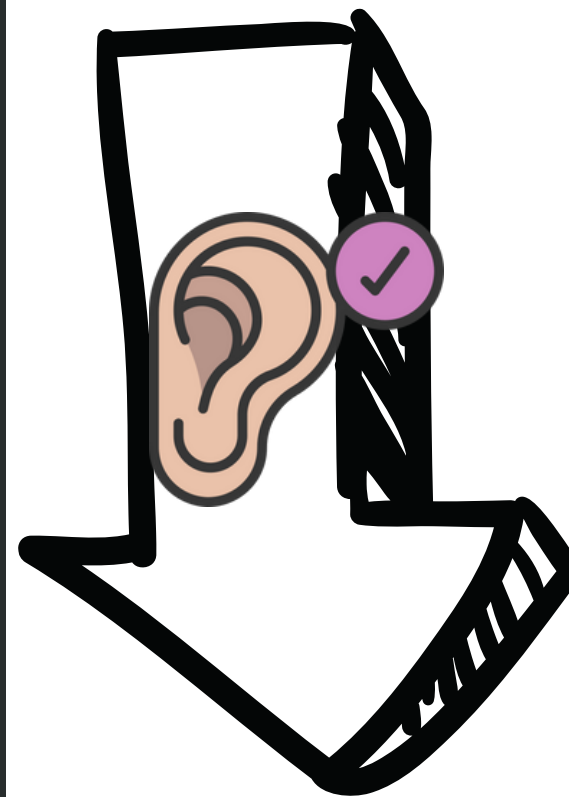


- 파스 트리의 노드를 순회하며 각 노드에 대한 동작 정의
- 자동 생성되는 BaseListener 클래스를 상속하여 구현

02 Listener



Parse Tree



- 깊이 우선 탐색(DFS)
- 이벤트 콜백 방식(exit, enter)

02 Listener

```
@Override public void exitType_spec(MiniCParser.Type_specContext ctx) {  
    String result = "";  
  
    if(ctx.VOID() != null)  
        result = "void";  
    else  
        result = "int";  
  
    cTree.put(ctx, result);  
}
```

Listener 함수 예시

03 구현 과정 예시 : PRETTY PRINT

- PRETTY PRINT

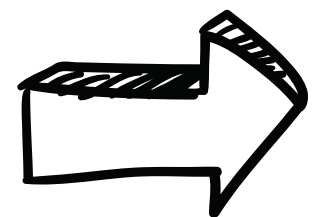
소스 코드를 사람이 읽기 좋게 Formatting

```
test.c x
1 int main() {
2 int a = 5;int b = 4;int c;c = a+b;
3 }
4 |
```



```
/opt/homebrew/Cellar/openjdk/23.0.2/libexec/o
int main() {
    int a = 5;
    int b = 4;
    int c;
    c = a + b;
}
```

ANTLR의 파싱 기능과 Listener를 활용해 Pretty Print를 구현



위와 같은 방식으로 함수를 수정하여 코드 난독화 도구 구현

질문과 답변 Q & A

궁금한 점이 있다면 자유롭게 질문 바랍니다 !

THANK YOU

감사합니다