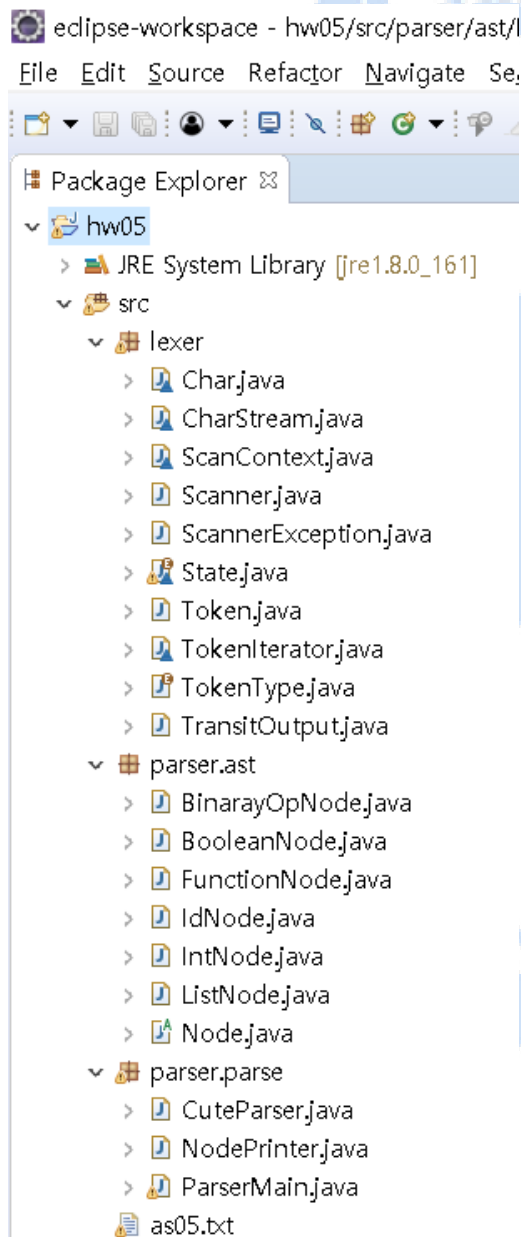


프로그래밍언어개론 보고서

컴퓨터공학과 201604140 박경수

PL Assignment #5: Cute18 Parser

전체 구조



3주차 과제 State 클래스 - START 수정

```
edipse-workspace - hw05/src/lexer/State.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

State.java
1 package lexer;
2
3 import static lexer.TokenType.*;
4
5
6
7 enum State {
8     START {
9         @Override
10        public TransitionOutput transit(ScanContext context) {
11            Char ch = context.getCharStream().nextChar();
12            char v = ch.value();
13            switch ( ch.type() ) {
14                case LETTER:
15                    context.append(v);
16                    return GOTO_ACCEPT_ID;
17                case DIGIT:
18                    context.append(v);
19                    return GOTO_ACCEPT_INT;
20                case SPECIAL_CHAR:
21                    Character cr = new Character(v); // toString()을 쓰기위해 cr을 생성해준다.
22                    context.append(v);
23                    if(v == 35) { // 첫 문자가 #일 경우.
24                        return GOTO_ACCEPT_ID; // 그 다음에 T나 F가 나올거니까 ID로 보내준다.
25                    }
26                    else return GOTO_SIGN; // 그 외 특수문자
27                case WS:
28                    return GOTO_START;
29                case END_OF_STREAM:
30                    return GOTO_EOS;
31                default:
32                    throw new AssertionError();
33            }
34        }
35    },
36 }
```

1. 3주차 과제에서 SPECIAL_CHAR 일경우에 무조건 GOTO_MATCHED로 보냈었는데, 이번과제에서 그렇게 하면 마지막 “)” 문자를 받았을 때 readToNextToken에서 FAILED상태로 가기 때문에 수정이 필요했다.
2. #일 경우를 제외하고는 append를 해준 후 GOTO_SIGN으로 리턴을 해줌.

3주차 과제 State 클래스 - SIGN 수정

```
edipse-workspace - hw05/src/lexer/State.java - Edipse
File Edit Source Refactor Navigate Search Project Run Window Help

State.java
83 SIGN {
84     @Override
85     public TransitionOutput transit(ScanContext context) {
86         Char ch = context.getCharStream().nextChar();
87         char v = ch.value();
88         switch ( ch.type() ) {
89             case LETTER:
90                 return GOTO_FAILED;
91             case DIGIT:
92                 context.append(v);
93                 return GOTO_ACCEPT_INT;
94             case WS:
95                 char a = context.getLexime().charAt(0);
96                 Character cr = new Character(a);
97                 String b = new String(cr.toString());
98                 return GOTO_MATCHED(TokenTypes.fromSpecialCharactor(a), b);
99             case END_OF_STREAM:
100                 char a2 = context.getLexime().charAt(0);
101                 Character cr2 = new Character(a2);
102                 String b2 = new String(cr2.toString());
103                 if(a2 == 41)
104                     return GOTO_MATCHED(fromSpecialCharactor(a2), b2); // 마지막에 ')'이 왔을 때 실패하지 않고 MATCHED로 가게끔
105                 return GOTO_FAILED;
106             default:
107                 throw new AssertionError();
108         }
109     }
110 }
```

1. SIGN에서 “)”이 왔을 때를 처리해야 하기 때문에, 스페이스바가 왔을 경우와 동일하게 END_OF_STREAM의 case에서도 context에 있는 “)”값을 받아와서 적절한 형태로 변환해준다.
2. GOTO_FAILED로 리턴 해주기 전에 “)”일 경우 GOTO_MATCHED로 리턴 해준다.

FunctionNode 클래스

edipse-workspace - hw05/src/parser/ast/FunctionNode.java - Edipse

File Edit Source Refactor Navigate Search Project Run Window Help

```
FunctionNode.java
1 package parser.ast;
2
3 import java.util.HashMap;
4
5
6
7
8 public class FunctionNode extends Node { //binaryOpNode클래스를 보고 참고해서 작성
9     enum FunctionType {
10         ATOM_Q { TokenType tokenType() {return TokenType.ATOM_Q;} },
11         CAR { TokenType tokenType() {return TokenType.CAR;} },
12         CDR { TokenType tokenType() {return TokenType.CDR;} },
13         COND { TokenType tokenType() {return TokenType.COND;} },
14         CONS { TokenType tokenType() {return TokenType.CONS;} },
15         DEFINE { TokenType tokenType() {return TokenType.DEFINE;} },
16         EQ_Q { TokenType tokenType() {return TokenType.EQ_Q;} },
17         LAMBDA { TokenType tokenType() {return TokenType.LAMBDA;} },
18         NOT { TokenType tokenType() {return TokenType.NOT;} },
19         NULL_Q { TokenType tokenType() {return TokenType.NULL_Q;} };
20
21         private static Map<TokenType, FunctionType> fromTokenType = new HashMap<TokenType, FunctionType>();
22
23         static {
24             for (FunctionType fType : FunctionType.values()){
25                 fromTokenType.put(fType.tokenType(), fType);
26             }
27         }
28         static FunctionType getfunctionType(TokenType tType){
29             return fromTokenType.get(tType);
30         }
31         abstract TokenType tokenType();
32     }
33
34     public FunctionType value;
35
36     @Override
37     public String toString(){
38         return value.name();
39     }
40     public void setValue(TokenType tType) {
41         FunctionType fType = FunctionType.getfunctionType(tType);
42         value = fType;
43     }
44 }
```

1. 각각의 functiontype에 대한 enum을 정의 해준다.
2. Map 컬렉션을 이용하여 TokenType(key)과 FunctionType(Value)으로 구성된 객체를 생성 후 추가를 해준다.
3. setValue()메소드는 Map에 저장된 타입을 가져와서 value값을 지정해준다.
4. toString()을 호출할 때 value의 name을 리턴해준다.

CuteParser 클래스 FunctionNode 키워드 부분

```
edipse-workspace - hw05/src/parser/parse/CuteParser.java - Edipse
File Edit Source Refactor Navigate Search Project Run Window Help

CuteParser.java
66      binaryOpNode binaryOpNode = new BinaryOpNode();
67      binaryNode.setValue(tType);
68      return binaryNode;
69
70      // FunctionNode 키워드가 FunctionNode에 해당
71      case ATOM_Q:
72      case CAR:
73      case CDR:
74      case COND:
75      case CONS:
76      case DEFINE:
77      case EQ_Q:
78      case LAMBDA:
79      case NOT:
80      case NULL_Q:
81          FunctionNode functionNode = new FunctionNode();
82          functionNode.setValue(tType);
83          return functionNode;
84
85      // BooleanNode
86      case FALSE:
87          BooleanNode falseNode = new BooleanNode();
88          falseNode.value = false;
89          return falseNode;
90
91      case TRUE:
92          BooleanNode trueNode = new BooleanNode();
93          trueNode.value = true;
94          return trueNode;
95      // case L_PAREN일 경우와 case R_PAREN일 경우
96      // L_PAREN일 경우 parseExprList()를 호출하여 처리
97      case L_PAREN:
98          ListNode listNode = new ListNode();
99          listNode.value = parseExprList();
100         return listNode;
101      case R_PAREN:
102         return null;
103      default:
104         // head의 next를 만들고 head를 반환하도록 작성
105         System.out.println("Parsing Error!");
106         return null;
107     }
108 }
```

1. FunctionNode 키워드에 대한 모든 case에 대해서,
2. functionNode 객체를 생성해준다.
3. setValue()를 통해 가져온 value값으로 functionNode의 value를 지정해준다.
4. functionNode를 리턴해준다.

실행결과

edipse-workspace - hw05/src/parser/parse/ParserMain.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

```
1 package parser.parse;
2
3 import java.io.File;
4
5
6
7 public class ParserMain {
8
9     public static final void main(String... args) throws Exception {
10         ClassLoader cloader = ParserMain.class.getClassLoader();
11         File file = new File(cloader.getResource("as05.txt").getFile());
12         CuteParser cuteParser = new CuteParser(file);
13         NodePrinter.getPrinter(System.out).prettyPrint(cuteParser.parseExpr());
14     }
15 }
```

Quick Access

Problems @ Javadoc Declaration Console

<terminated> ParserMain [Java Application] C:\Program Files\Java\jre1.8.0_161\bin#\n ([PLUS] ([MINUS] [INT: 3] [INT: 2]) [INT: -378])

1. 기존 file road 코드에서 변경하지 않음.