

고급프로그래밍언어론 과제1

(ANTLR를 이용한 계산기)

소 속 : 충남대학교 컴퓨터공학과
학 번 : 201760406
이 름 : 정 재 엽
연 락 처 : 010-9412-9756

1 Program List

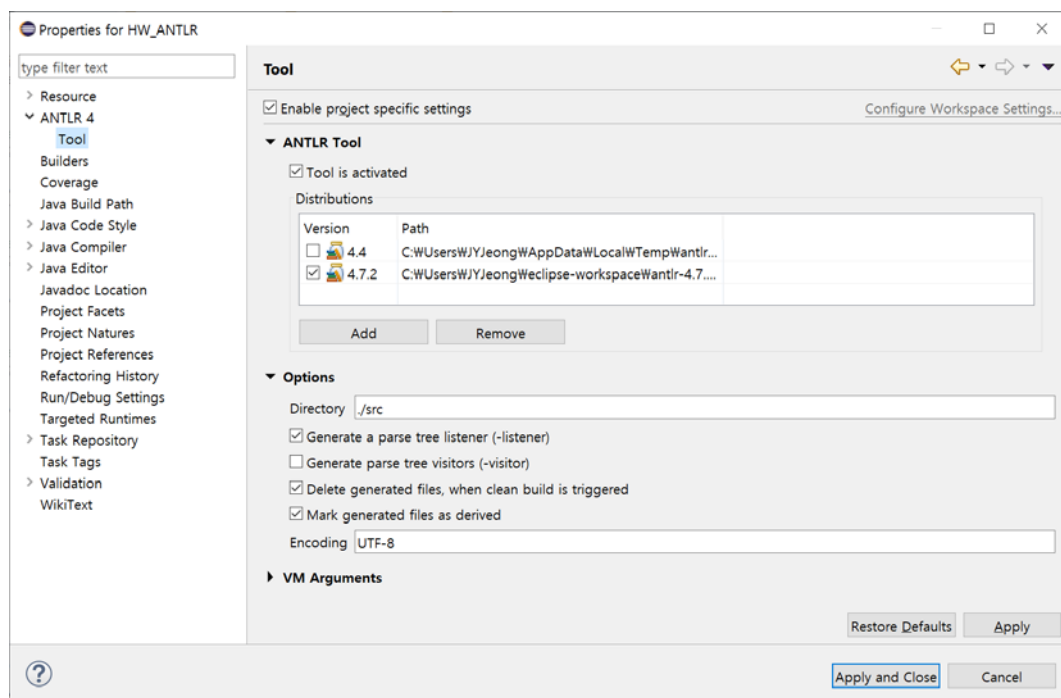
ANTLR를 이용하여 계산기를 작성하기 위해 Eclipse for JAVA, ANTLR 4, Facted Project Framework를 사용하였으며, 각 프로그램의 버전은 아래와 같습니다.

Tools	Version	Description
Eclipse	2019-09 R (4.13.0)	Build ID: 20190917-1200
Java	JDK-12.0.2	JDK 설치 및 경로 설정
ANTLR 4 SDK	0.3.6.201611052310	eclipse plug-in으로 설치
ANTLR 4	4.7.2	ANTLR 홈페이지에서 java를 위한 최신 jar파일 다운로드
Eclipse Facted Project Framework	3.7.2.v201903222010	ANTLR 프로젝트에 Java Project를 통합하기 위해 사용
Eclipsed Facted Project Framework JDT Enablement	3.12.0.v201906031755	

2 Eclipse Project for ANTLR4

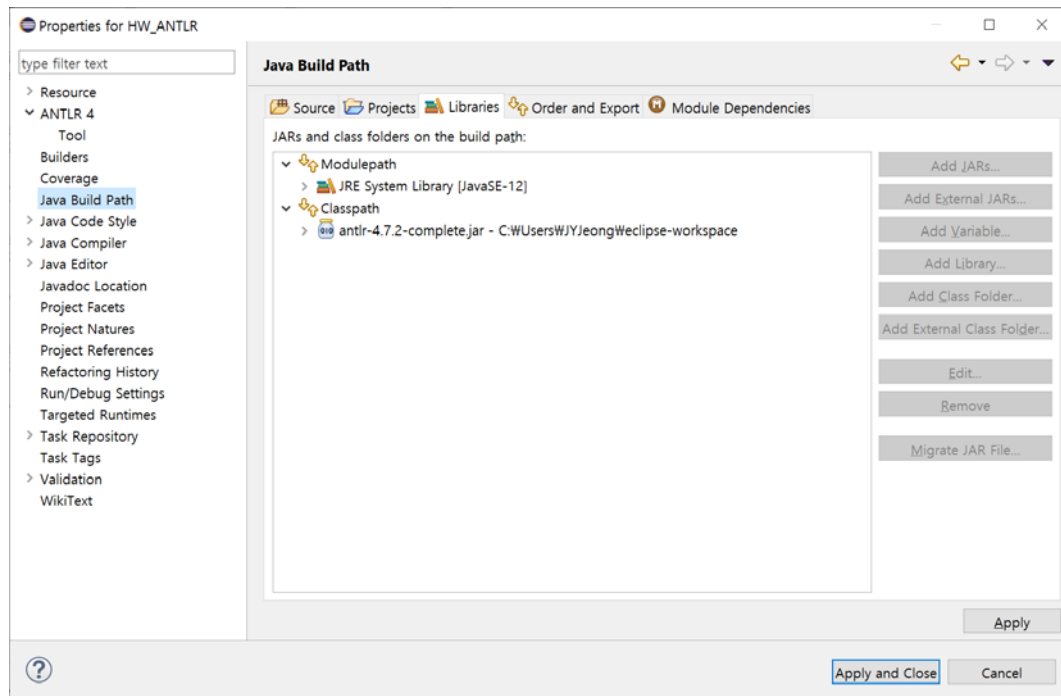
아래와 같은 절차대로 ANTLR4 프로젝트를 생성하였습니다.

1. New → Project → General → ANTLR 4 Project
2. Project Properties → Project Facets 에서 Java language로 Application을 작성할 수 있도록 설정
3. 2번을 수행하고 Project Properties를 다시 확인하면 JAVA와 관련된 설정이 가능하도록 됨
4. Project Properties → ANTLR 4 → Tool 설정
 - A. Enable project specific setting을 체크하고, ANTLR Tool에 antlr-4.7.2-complete.jar 파일을 추가하고 사용하도록 체크함. 기존에 추가되어 있는 4.4는 사용하지 않음
 - B. Options → Directory를 ./src로 설정하여 WHILE.g4 실행 시 생성되는 파일들이 ./src/generated에 추가될 수 있도록 함



5. Project Properties → Java Build Path 설정

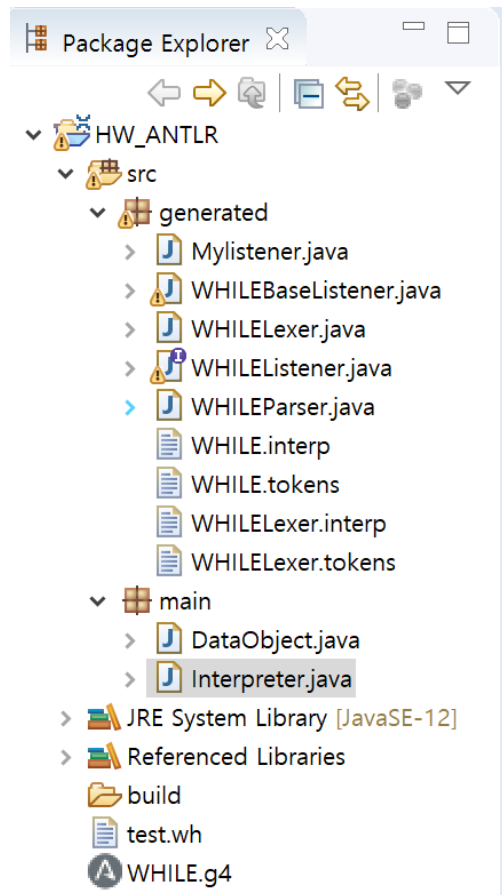
A. Libraries → Add External JARs을 이용하여 Classpath에 antlr-4.7.2-complete.jar 파일 추가



3 Project File List

ANTLR를 이용하여 계산기를 작성하기 위해 아래 파일들을 제작하였으며, 용도는 아래와 같습니다.

File Name	Note
WHILE.g4	<ul style="list-style-type: none"> - WHILE grammar를 이용하여 N, IDENT, *, +, -에 대한 Rule과 Action을 정의 - Action후에 # 명령을 추가함으로써, WHILEBaseListener.java에 각 Operation에 대한 Enter/Exit Function이 추가될 수 있도록 함 - IDENT 값을 DataObject Class에서 얻어오기 위해 해당 함수를 import함 (import static main.DataObject.GetIdentValue)
Interpreter.java	<ul style="list-style-type: none"> - Lexer, Token, Parser, Tree는 제공된 코드 사용 - 결과 출력을 위해 Listener Object를 생성하고 ParseTreeWalker를 이용하여 수행
DataObject.java	<ul style="list-style-type: none"> - Identifier x를 static으로 정의하고 해당 값을 return하는 GetIdentValue() 함수를 static으로 작성함 - Identifier는 x밖에 없으며, 다른 문자가 들어올 경우 0을 리턴하도록 함
Mylistener.java	<ul style="list-style-type: none"> - WHILEBaseListener class를 상속받아 각 연산에 대한 Exit Function을 Override함 - 계산 절차를 확인하기 위해 간단히 ctx.sv 값을 출력하고, 최종 결과를 출력하기 위해 Stack에 push함 - exitProgram() 에서 입력으로 받은 수식을 출력하고 최종 결과 값을 Stack에서 pop 하여 출력 함



4 실행결과

test.wh파일에 다음과 같은 수식을 넣어 Parsing하도록 하였습니다. $(7+4+3*3-3*2-x)$, $x=100$
 Multiply의 연산에 우선순위를 두기 위해 WHILE.g4 작성시 Multiply를 상위에 배치하였습니다.
 수행결과 : $7+4+(3*3)-(3*2)-x = -86$

