Atitit 语法分析 递归下降法原理状态机原理状态模式

每个token+state 对应一个class、method

双case模式 token》state 双case算法

package com.attilax.core;

import java.util.List;

import com.alibaba.fastjson.JSON;

import com.attilax.ast.ClassInstanceCreation;

import com.attilax.ast.Expression;

import com.attilax.ast.MethodInvocation;

import com.attilax.ast.SimpleName;

import com.attilax.collection.listBuilder;

import com.attilax.parser.Token;

import com.attilax.str.strService;

import com.google.common.collect.Lists;

public class AstBuilderV3s528 {

public static void main(String[] args) {

String s = "invokeMethod com.attilax.core.methodRunner \"string\" contstuParamVal . methDync \"string\" haha";

String[] a = s.split(" ");

List li = listBuilder.$(a).trimElement().delEmptyElement().build();

com.attilax.ast.Expression buildAstV3s528 = new AstBuilderV3s528().buildAstV3s528(li);

System.out.println(JSON.toJSONString(buildAstV3s528,true) );

}

List<String> tokens;

int token\_index = 0;

int paramIdx = 0;

String stat = "ini";

private Expression Expression;

public Expression buildAstV3s528(List<String> tokens) {

this.tokens = tokens;

System.out.println("tokenindex" + token\_index);

if (token\_index == 4)

System.out.println("dbg");

if (token\_index >= tokens.size())

return this.Expression;

String cur\_token = tokens.get(token\_index);

if (token\_index == 0 && stat.equals("ini")) {

tokenIndexZeroStat(cur\_token);

} else if (stat.equals("startgetClassname")) {

startgetClassnameState(cur\_token);

} else if (stat.equals("ClassInstanceCreationParamType")) {

ClassInstanceCreationParamTypeState(cur\_token);

}

else if (!cur\_token.equals(".") && stat.equals("ClassInstanceCreationParamStart")) {

ClassInstanceCreationParamStartStat(cur\_token);

}

else if (cur\_token.equals(".") && stat.equals("ClassInstanceCreationParamStart")) {

this.stat = "ClassInstanceCreationParamEnd";

} else if (stat.equals("ClassInstanceCreationParamEnd")) {

ClassInstanceCreationParamEndStat(cur\_token);

} else if (stat.equals("MethodInvocationType")) {

this.stat = "MethodInvocationParamStart";

this.Expression.paramtypes = cur\_token.split(",");

} else if (stat.equals("MethodInvocationParamStart")) {

MethodInvocation cic = (MethodInvocation) this.Expression;

cic.arguments.add(cur\_token);

}

token\_index++;

return buildAstV3s528(tokens);

}

private void ClassInstanceCreationParamEndStat(String cur\_token) {

MethodInvocation mi = new MethodInvocation();

mi.jsonname="MethodInvocation";

mi.Exp = this.Expression;

mi.Name = cur\_token;

this.Expression = mi;

this.stat = "MethodInvocationType";

}

private void ClassInstanceCreationParamStartStat(String cur\_token) {

ClassInstanceCreation cic = (ClassInstanceCreation) this.Expression;

Object params = getArg(cur\_token, paramIdx, this.Expression.paramtypes);

cic.arguments.add(cur\_token);

paramIdx++;

}

private void ClassInstanceCreationParamTypeState(String cur\_token) {

this.stat = "ClassInstanceCreationParamStart";

this.Expression.paramtypes = cur\_token.split(",");

}

private void startgetClassnameState(String cur\_token) {

if (this.Expression instanceof ClassInstanceCreation) {

ClassInstanceCreation cic = (ClassInstanceCreation) this.Expression;

cic.name = cur\_token;

} else // static class simplename

{

SimpleName sn = (SimpleName) this.Expression;

sn.IDENTIFIER = cur\_token;

}

this.stat = "ClassInstanceCreationParamType";

}

private void tokenIndexZeroStat(String cur\_token) {

if (cur\_token.equals("invokeMethod")) {

ClassInstanceCreation cic = new ClassInstanceCreation();

// cic.arguments = getArgs(tokens, m\_index);

this.Expression = cic;

this.Expression.jsonname = "ClassInstanceCreation";

this.stat = "startgetClassname";

//

}

if (cur\_token.equals("invokeStaticMethod")) {

this.Expression = new SimpleName(cur\_token);

this.Expression.jsonname = "ClassInstanceCreation";

this.stat = "startgetClassname";

}

}

private Object getArg(String cur\_token, int paramIdx2, String[] paramtypes) {

String ptype = paramtypes[paramIdx2];

if (ptype.equals("int"))

return Integer.parseInt(cur\_token.toString());

return cur\_token;

}

}