Atitit..状态机与词法分析 通用分词器 分词引擎的设计与实现 attilax总结

[1. 状态机 理论参考 1](#_Toc5706)

[2. 词法分析理论 1](#_Toc32358)

[3. 词法分析实例 2](#_Toc1163)

[4. ---code fsm 状态机通用实现 2](#_Toc3811)

[4.1. Context 2](#_Toc20725)

[4.2. public abstract class State { 3](#_Toc288)

[4.3. public class Token { 4](#_Toc111)

[4.4. public class FinishState extends State { 4](#_Toc14912)

[5. 扫描器 5](#_Toc28671)

[5.1. public class Scanner { 5](#_Toc17888)

# 状态机 理论参考

Atitit. 有限状态机 fsm 状态模式 - attilax的专栏 - 博客频道 - CSDN.NET.htm

Atitit. 有限状态机 fsm 状态模式 - attilax的专栏 - 博客频道 - CSDN.NET.htm

# 词法分析理论

atitit.词法分析的实现token attilax总结 - attilax的专栏 - 博客频道 - CSDN.NET.htm

atitit.自己动手开发编译器and解释器(1) ------词法分析--attilax总结 - attilax的专栏 - 博客频道 - CSDN.NET.htm

作者:: 老哇的爪子 Attilax 艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 词法分析实例

atitit..sql update语法的词法分析，与语法ast构建 - attilax的专栏 - 博客频道 - CSDN.NET.htm

atitit.java解析sql语言解析器解释器的实现 - 其他综合 - 红黑联盟.htm

# ---code fsm 状态机通用实现

package com.attilax.fsm;

import java.util.ArrayList;

import java.util.List;

import com.attilax.lang.text.strUtil;

## Context

public class Context {

public State state;

public List<Token> tokenList=new ArrayList <Token>();

public Token curToken=new Token();

public int curcharIndex;

public char[] sa;

public State getState() {

return state;

}

public void setState(State state) {

this.state = state;

}

public void request(String s) {

if(sa==null)

sa=s.toCharArray();

state.handle(s, this);

}

}

package com.attilax.fsm;

import java.util.List;

import com.attilax.fsm.camelScanner.iniState;

**package** com.attilax.fsm;

## public abstract class State {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

}

**public** **boolean** MoveNextisEnd(Context context) {

context.curcharIndex++;

**if**(context.curcharIndex>=context.sa.length)

{

addCurToken(context );

context.state=**new** FinishState();

**return** **true**;

}

**return** **false**;

}

**public** **void** addCurToken(Context context) {

context.tokenList.add(context.curToken);

}

**public** **void** newToken(Context context , **char** curChar) {

Token token = **new** Token();

token.value= token.value+String.*valueOf*(curChar);

context.curToken=token;

}

**public** **void** addCurTokenNnewToken(Context context, **char** curChar) {

addCurToken(context);

newToken(context,curChar);

}

**public** **abstract** **void** handle(String sampleParameter, Context context) ;

}

**package** com.attilax.fsm;

## public class Token {

**public** String value="";

**public** String getValue() {

**return** value;

}

**public** **void** setValue(String value) {

**this**.value = value;

}

}

**package** com.attilax.fsm;

## public class FinishState extends State {

@Override

**public** **void** handle(String sampleParameter, Context context) {

// **TODO** Auto-generated method stub

}

}

# 扫描器

## public class Scanner {

public static void main(String[] args) {

// TODO Auto-generated method stub

String s = "fld1=1,fld2='at''t,lax',fld3='val3'";

s="aaaaCamelJSONObject123forMac";

}

public List getTokenList(String s) {

// DslPaserContext context = new DslPaserContext();

Context context = new Context ();

context.setState(new iniState());

int n = 0;

while (!(context.state instanceof FinishState)) {

// System.out.println(n);

// ����

context.request(s);

n++;

if (n > 200)

break;

}

for (Token tk : context.tokenList) {

// if(tk.value.trim().length()>0)

System.out.println(tk.value + "===");

}

return (List) context.tokenList;

}

}

**public** List<String> getTokenList\_retStr(String s)

{

**if**(s.length()==0)

**return** **new** ArrayList<String>();

List<String> li\_r=**new** ArrayList<String>();

List<Token> li=getTokenList(s);

**for** (Token token : li) {

li\_r.add(token.value);

}

**return** li\_r;

}