**CmsLatte**

**목적**

Page소스를 생성하기 위해서 작성함.

게시판에 파생된 필드를 생성할 수 있도록 함

**특징**

* + Php와 유사하게 텍스트속에 포함된 <@...@> 사이의 부분을 프로그램의 소스로 생각하고 해석한다. Interpreter식으로 한 라인식 해석해서 결과를 만든다.
  + 각 문장을 해석한 결과를 모두 가지고 있다가 나중에 한번에 string형태로 제공한다.
  + 데이터타입은 자동으로 판별되며 명확하게 기술하지 않는다.
  + 데이터타입에 Array와 Table을 기본으로 제공한다
  + 사용자 정의 함수를 제공하지 않는다.
  + 객체지향언어가 아님, 단순 스크립트 언어임

**DataType**

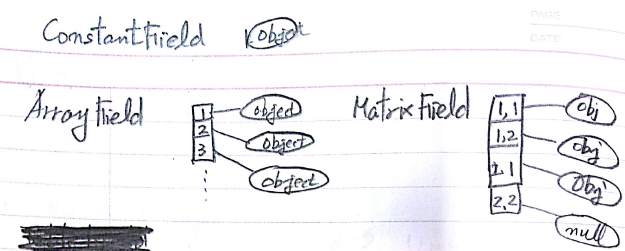
* DataType 7가지 DISBD,Array,Matrix(Table)
* Boolean,Integer,Double,String,Date,Array,Table
* 주의)Char,byte없음 그래서 0x0A나 ‘a’ 와 같은 표현은 없음

**토큰(Token)**

토큰은 각 소스부분을 해석해야하는 단위로 나눈 것이다. TokenType으로 구분한다.

**Field**

* Field는 varaible을 의미한다.
* 3가지 종류가 있다.
  + Constant, Array, Table
* Constant는 단일 Data를 가진다.
* Array는 Array형태의 Data를 가진다.
* Matrix는 Table형태의 Data를 가진다.



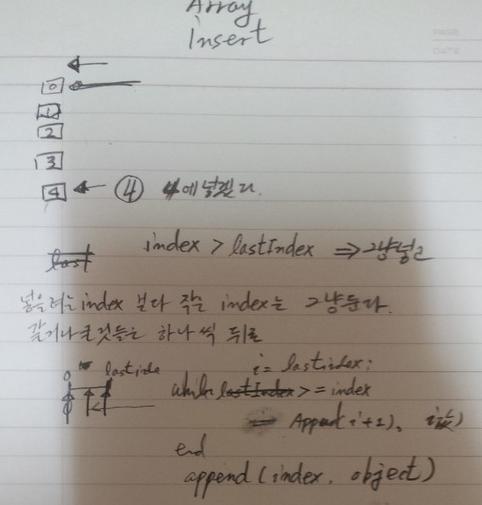
Array와 Matrix는 map형태의 자료구조를 가지고 운용하며 Array의 키는 integer형이고 Matrix의 키는 row 정수 + “,” + col 정수의 String형을 키로 한다.

**Array**

Array 란 DISBD의 타입들만 인자로 갖는다. Array나 Table등은 인자로 갖을 수 없다

Array + Array,DISBD 가 가능하다

LinkedHashMap<Integer,Object>로 바꾸다

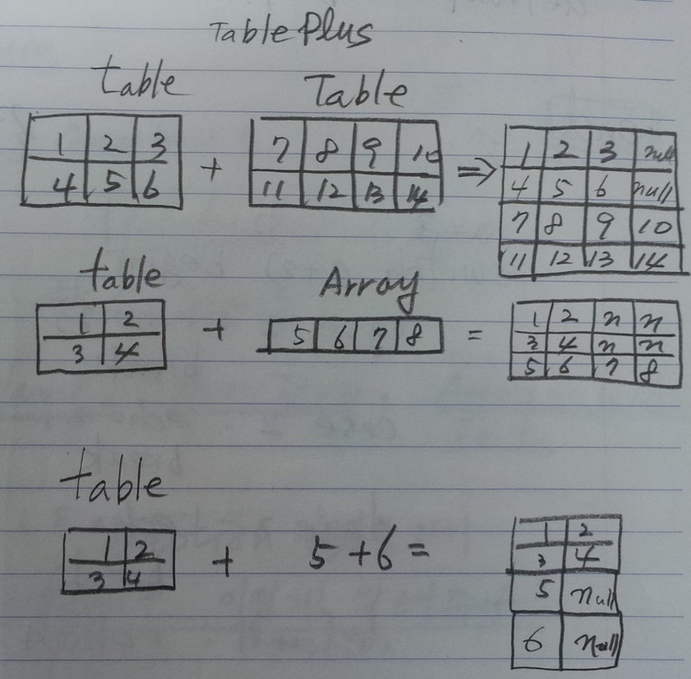


**Matrix**

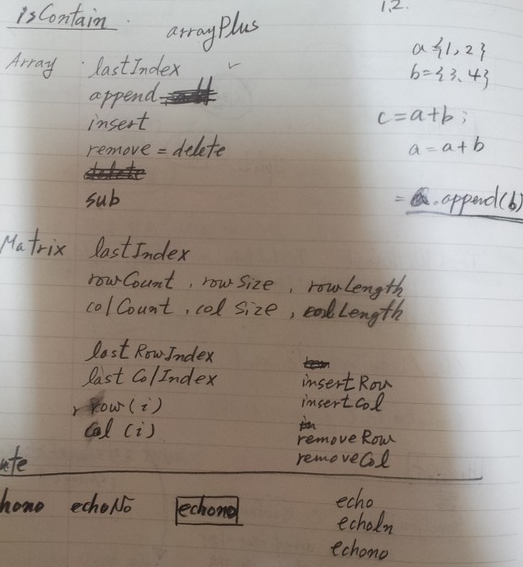
LinkedHashMap<Integer,ArrayField>로 바꾸다

Matrix + Matrix 는 큰 column에 맞춰진다.

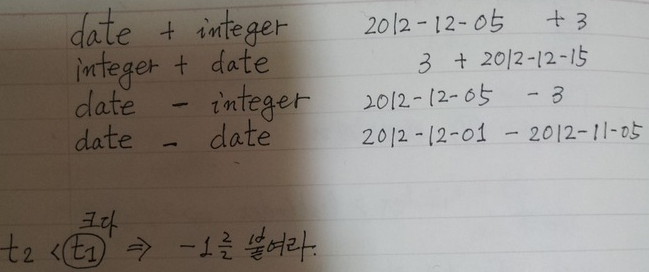
Matrix + array, Matrix + DISBD 를 지원한다



**Array 와 Matrix 함수들**

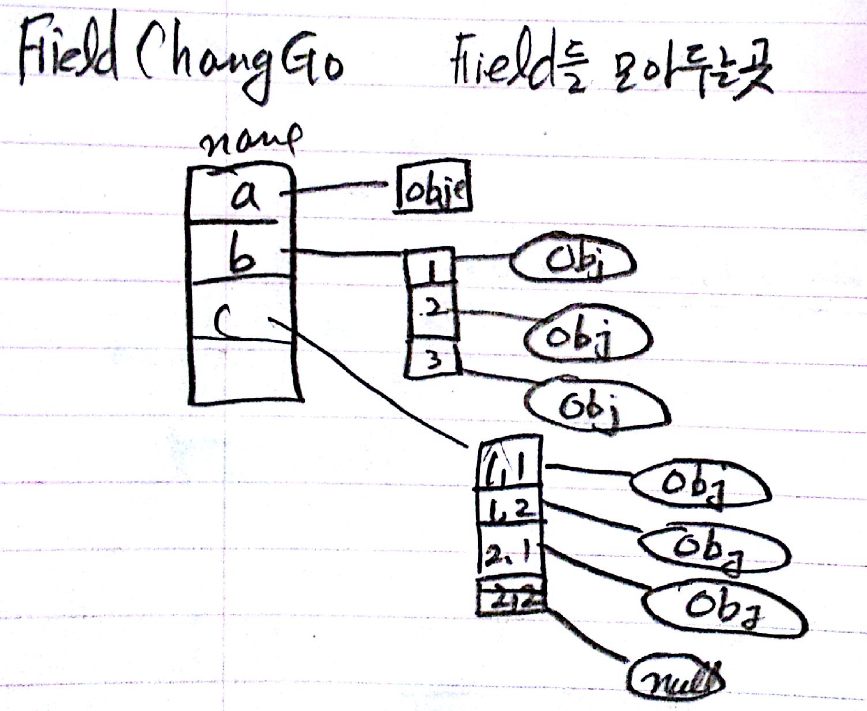


**Date**

****

**FieldChanggo**

Constant,Array,Matrix형태의 Field(Variable)을 관리한다.

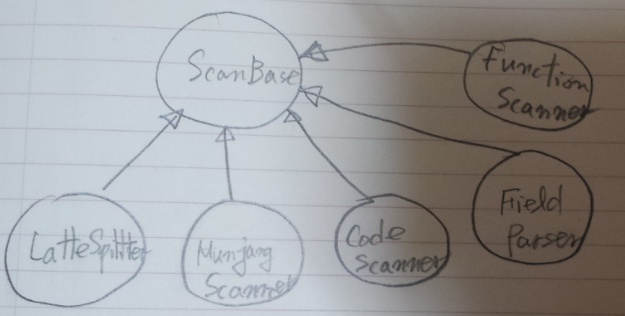


같은 이름의 변수를 저장시 기존에 있던 변수는 삭제된다.

AddOrReplace형태로 저장된다.

**Scanner들(**

Scan=parser=Splitter 모두 문자열을 char 단위로 검사하는 기능을 수행하는 클래스를 의미한다



**ScannerBase**

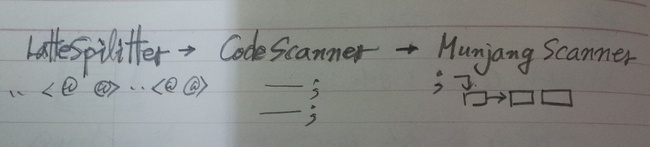
Scanner의 부모클래스로 추상클래스이며 scan시 필요한 함수들과 데이터를 가지고있다.

Index : 주어진 문자열를 char단위로 조사할때의 index

Buffer: Stringbuilder로 char단위로 문자열을 저장하는 임시버퍼의 역할을 한다.

srcArray : char[] 로 주어진 문자열을 char 배열로 가지고 있다.

* + LatteSpilitter
    1. Template의 소스부분과 CmsLatte가 해석해야할 부분(Code)을 나눈다.
    2. LatteItem class각 각 나누어진 부분을 담고 있는 클래스이다.
  + CodeScanner
    1. LatteSplitter가 나눈Code부분을 다시 각 문장단위로 나눈다. 문장이란 CmsLatte가 해석할 수 있는 1개의 의미있는 토큰들의 리스트라고 할 수 있다.
    2. Ex) a = 10;
    3. 각 문장은 ‘;’ 로 분리되어 있다.
  + MunjangScanner
    1. 1개의 문장을 해석하여 결과로 TokenStack을 리턴한다.
  + FunctionScanner
    1. CmsLatte가 가지고 있는 predefined function들을 해석하여 수행할 수 있도록 한다
  + FieldParser
    1. CmsLatte는 특별한 Field명명법을 사용하는데 그것을 지원할 수 있도록 파싱하여 FieldName class를 리턴한다.



**Munjang Scanner의 역할**

한문장을 스캔한다.

문장이란 ...; ';'로 끊어지는 cmsLatte program code의 1개의 문장을 의미함

ex) a = b+3; if a==3 ; begin; end; a[i,4] = b[4].substring(3,5);

[ ] 안에는 integer or integer constant field만 들어갈 수 있다.

이때 수식에서 해석할 수 있는 단위로 끊어내야한다.

a[1] = b[2] + c[3].substring(1,2) 일경우 토큰은 5개가 되어야한다.

a[1] , =, b[2] , + , c[3].substring(1,2) 와 같이 끊어져야한다.

허용하지 않는 문법

1. a [1]; 고 같이 a 다음에 스페이스가 있고 [ 가 나오는 것을 허용치 않음.

예) abc [3] 인경우 에러이다.

2. double형 숫자를 쓸 때 앞에 스페이스가 있으면 에러이다.

예) 1.5는 허용, .5는 허용치 않음

A = 1.5 ; //O , A = .5 //X

**Field명명법**

Field(variable)은 다음과 같은 명명법을 사용한다

prefix**\_**name[susik,susik]**.**subfunctionDesc

prefix : 사용할 수도 있고 아닐 수도 있다. Prefix는 1개의 문자를 의미하며 표현될 부가 함수이다

Name: 필수(알파벳과 숫자로 100자까지)

Susik : array나 matrix 필드의 row,col으로 수식을 사용가능하다

subfunctionDesc : field가 fieldType에 따라서 사용가능한 함수들

**Exception**

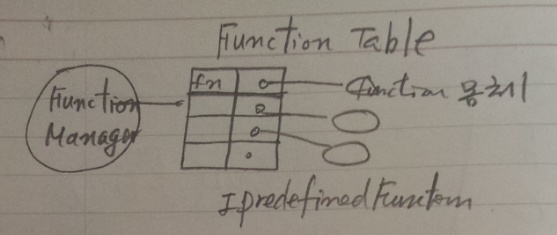
2개의 exception을 제공한다.

CmsLattException과 하위클래스 CmsLatteFunctionException

CmsLatteException은 전반적으로 사용되면 CmsLatteFunctionException은 predefeined function를 작성할 때 사용한다

**Functions**

CmsLatte는 predefined function을 갖는다. 이 함수들은 IPredefinedFunction을 구현한 것이어야한 것



IPredefinedFunction의 프로토타입은 다음과 같다

***Object* invoke*(Object... args) throws CmsLatteFunctionException;***

FunctionManager는 모든 function들을 관리한다.

싱글레톤으로 FunctionTable을 가지고 있다. FunctionTable은 functionName을 키로

predefinedFunction을 갖고 있는 map을 가지고 있다.

FunctionManager를 CmsLatte의 다른 프로세스(Susik)이 사용하여 함수를 수행하는데 필요한 인자들을 FunctionArguments클래스에 담아서 넘겨받아 수행하는 execute함수를 가지고 있다.

각 기능별로 함수들을 모아 놓는다.

파일명은 Predefined로 시작하며 상위 추상클래스 PredefinedFunctionBase로 상속받아 생성한다.

|  |  |
| --- | --- |
| 파일명 | 주요기능 |
| PredefinedCommonFunctions | 기본적이면서 각 Data Type에서 공통으로 사용하는 함수들 |
| PredefinedStringFunctions | String관련 함수들 |

Function은 subfunction이 수행되고 난 뒤에 prefix 가 수행된다.

u\_s(“abcdef”).subStr(0,3) -> subStr 수행후 -> toUpperCase가 수행된다.

**수식처리기susik**

수식 처리기는 사칙연산을 비롯한 연산동작을 수행한다

연산의 결과를 저장하기 위한 imsiChangGo를 가지고 있다.

수식자체가 문장이 되는 경우 echo를 수행한다.

수식은 scan함수에 의해서 TokenStack을 받아서 수행되며 결과는 토큰으로 돌려준다.

DISBD는 토큰의 value에 담아서 리턴한다

**Command해석기(CommandSuHangGi)**

fieldChanggo를 가지고 있다.

**문장의 종류들**

**수행**

String src="abc<@ a=10; echoln a;@>def";

CmsLatte cmsLatte = **new** CmsLatte(src);

String output = cmsLatte.createPage();

*assertEquals*(output,"abc10def");

**CmsLatte의 수행**

1. LatteSplitter가 소스코드와 일반텍스트부분을 분리한다.
2. LatteSplitter는 내부적으로 CodeScanner를 호출한다.
3. CodeScanner는 <@와 @> 사이의 문자열을 받아서 문장단위로 분리한다.
4. 결과적으로 justtext + code + code + … 와 같은 리스트가 생긴다.
5. latte함수에서 하나씩 수행하면서 output을 만든다

**echo**

echo는 DISBD는 string으로 변환하여 출력한다.

Array,matrix는 toString함수를 통해서 출력한다

수식은 결과를 echo로 처리한다

<@;(1+2);@> 인 경우 echo (1+2)와 같이 동작한다.

echoln : output + “\n”을 write한다

echono : 수행만 하고 결과를 write하지 않는다. Array or table에서 자기 자신에게만 적용되고 끝나는 함수의 수행시 필요하다

echono array1.insert(“b”)와 같은 경우

**흐름제어**

If문, for문,while문

각 문장에 맞게 Controller클래스를 둔다.

흐름을 조정하기 위해서 Controller를 stack에 담아서 현재의 문장이 무엇인지 판별한다.

**If 문**

조건이 맞는지 먼저 if에서 체크하고 나중에 end에서 체크한다.

문법

If 조건식 begin

…

End elseif 조건식 begin

….

End else begin

…

End;

**For 문**

for 변수또는 대입식 to 수식 begin

…

End;

Example)

For i=1 to 10 begin

End;

For I to 10 begin

End;

**While문**

While 조건식 begin

End;

**Switch 문**

Switch 수식 begin

Case 변수:

Break;

Default:

End;

**QuestionIf**

조건식 ? 문장1개: 문장1개;

1==1?echo 1: echo2;

문장은 echo, assign, goto, continue, break, 만 가능하다

**Break문**

looping문들 for, foreach,while, 에서 벗어난다

**Continue문**

For,foreach,while문에서 루프의 처음으로 가서 다시 조건을 검사한다

**Clear**

Clear variables|output;

clear단독으로 사용되면 output을 의미한다.

CmsLatte는 출력되는 내용을 최종적으로 담고 있다가 나중에 리턴해 줍니다. 그 출력되는 내용을 모두 clear합니다.

<@

Echo 1; clear; echo 2;

@>

2를 출력합니다. 1이 output 출력버퍼에 담겨 있는 상황에서 clear에 의해서 명령이

**Save**

결과 output을 파일 또는 필드에 넣는다.

save to file 파일명 with 인코딩 override boolean변수

example)

save output to file c:/1.txt with “utf-8” override true; //파일에 저장

save output to tmp; //변수에 저장

save variable to file c:/1.txt with “utf-8” override true; //파일에 저장

save variable to tmp;

**loadvar**

변수들을 저장해 둔 xml파일을 읽어들여서 창고에 넣는다

xml파일의 형식은 부록3과 같다.

loadvar 수식

example) loadvar c:/temp/1.xml

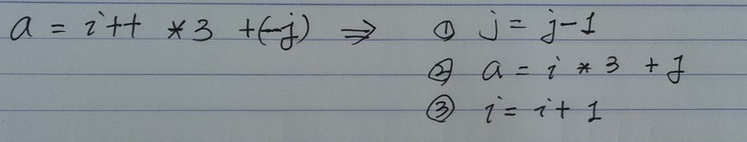
**PluseAssign, MinusAssign처리**

MunjangScan에서 += 를 만나면 = + 로 token을 만들어서 다시 넣는다.

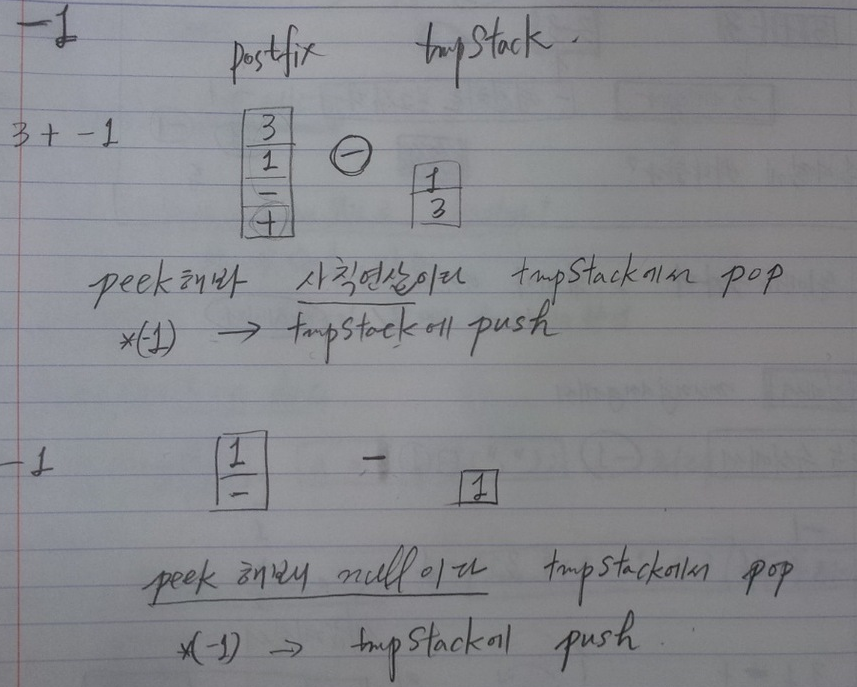
A += 3 --🡪 a = a + 3 으로 stack을 만든다

**++와 –의 처리**

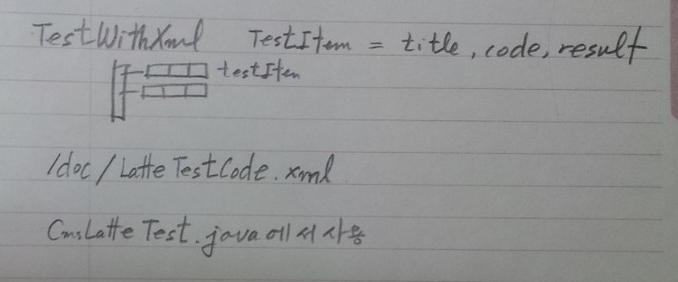
A = i++ \* 3 과 같은 경우 latte수행시 토큰 스택에서 ++가 있는지 판단하고 ++(또는 --)가 필드의 전인지 후인지 판단하여 pasaengcommand체 token stack을 채운다



**-1의처리**



**Xml에 코드와 결과를 담아서 테스트**



**원칙**

변수의 타입은 들어가는 값에 의해서 결정된다

Int a; 라는 식으로 a가 int인지 먼저 정하지 않는다

Var a : 라는 식으로 a가 변수라고 먼저 정하지 않는다.

A=10;

A=”ABC”;

A=true;

와 같이 사용한다. 이미 변수가 선언되어 있다면 덮어쓰기를 한다.

**배열의 초기화**

A={1,2,3};

A={};

**테이블의 초기화**

T={{}}; //빈 테이블

T={$2,3${}}; //row 2 , col 3인 테이블을 초기화한다

**Prefix functions**

String[] source = new String[]{

"<@ s=\"abc\"; echo u\_s;@>",

"<@ s=\"abc Def\"; echo l\_s;@>",

"<@ s=\"abc\"; echo a\_s;@>",

"<@ s=\"ABcdEfG\"; echo z\_s;@>",

"<@ s=\"ABcdEfG\"; echo c\_s;@>",

"<@ s=\"abc def ghi\"; echo c\_s;@>",

"<@ s=\"abc def ghi\"; echo p\_s;@>",

"<@ s=\"WCM\_USER\"; echo p\_s;@>",

"<@ s=\"WCM\_USER\"; echo s\_s;@>",

"<@ s=\"WCM\_USER\"; echo d\_s;@>",

};

String[] result = new String[]{

"ABC",

"abc def",

"Abc",

"aBcdEfG",

"Abcdefg",

"Abc Def Ghi",

"abcDefGhi",

"wcmUser",

"'WCM\_USER'",

"\"WCM\_USER\"",

}

**외부함수의 load**

Local(운영사이트)에서 필요한 동작을 java 클래스 만들어서 jar로 만들고 cmslatte에서 사용할 수있도록 load할 수 있는 기능을 한다.

외부함수의 작성은 먼저 CmsLatte.jar를 classpath에 추가하고 annotation CmsLatteFunction을 기술하여야한다.

함수는 FunctionManager의 loadExternalLib에서 수행한다

load className(수식) of jarfile(수식); 이렇게 cmslatte의 명령으로 load를 두는 것이 실패다

왜냐하면 javalatte와 달리 처음에 한꺼번에 모두 파싱해서 저장해 두기 때문에.

load문이 실행되기전 parsing을 해서 함수인지 판별하는데, 그때는 load가 되지 않아서 함수로 인식하지 않고, field로 인식한다.

**결국 FunctionManage가 생성될 때 config파일을 읽어서 거기서 load해야한다는 뜻**

FunctionManager가 생성될 때 1번 수행한다

1. **CmsLatteExternalFunctions.xml**을 classPath에서찾아서 parsing한다.
2. Jar와 그 안의 class를 기술하는 방법과
3. class만을 기술하는 방법 두가지가 있다.

/CmsLatteExternalFunctions.xml을 어디에 두어야 웹에서 사용가능한가?

**부록1-Predefined functions**

|  |  |
| --- | --- |
| Prefix | |
| u | 대문자로 만든다 |
| l | 소문자로 만든다 |
| z | 첫글짜만 소문자로 만든다 |
| a | 첫글짜만 대문자로 만든다 |
| c | 단어들의 첫글자만 대문자로 만든다Capital |
| p | Property 문자로 만든다. |
| s | Single quotation을 붙인다 |
| d | Double quotation을 붙인다 |

|  |  |
| --- | --- |
| **Common** (alias이름) 함수 | |
| length(size) |  |
| typeOf | String=typeOf(anyType) |
| toString |  |
| toDouble |  |
| toInteger |  |
| toDate |  |
| toBoolean |  |
| toArray |  |
| toTable |  |
| functionExist | 함수가 존재하는지 여부 |
| ifEmpty | 오라클 nvl |

|  |  |
| --- | --- |
| **String** (alias이름) 함수 | |
| trim |  |
| leftTrim, rightTrim (=ltrim,rtrim) |  |
| Substring |  |
| subStr |  |
| toUpperCase,toLowerCase |  |
| singleQuotation,doubleQuotation |  |
| firstCharLower,firstCharUpper |  |
| capital |  |
| propertyString |  |
| Pad |  |
| format |  |
| indexOf |  |
| Replace |  |
| Split |  |
| **Array** (alias이름) 함수 | |
| lastIndex |  |
| append |  |
| insert |  |
| remove |  |
| Sub |  |

|  |  |
| --- | --- |
| **Matrix** (alias이름) 함수 | |
| lastRowIndex,lastColIndex |  |
| rowCount,colCount(rowSize,colSize,  rowLength,colLength) |  |
| col(column),row |  |
| insertRow,insertCol |  |
| removeRow,removeCol |  |

|  |  |
| --- | --- |
| **Date** (alias이름) 함수 | |
| now |  |
| date |  |
| dayOfWeek |  |
| dayToArray |  |
| dayAdd |  |
| monthAdd |  |
| daysBetween |  |

**부록2 – 외부함수의 작성**

package kr.kalpa.co.cmslib;

import kr.dcos.cmslatte.annotation.CmsLatteFunction;

import kr.dcos.cmslatte.exception.CmsLatteFunctionException;

import kr.dcos.cmslatte.functions.LatteFunctionBase;

public class Reservation extends **LatteFunctionBase**{

private static String getNamesFromDb(){

return "abc";

}

@CmsLatteFunction(anotherName="hide4Char",desc="hide 4 char of name",subApply="string",backArg="false")

public static String hideName(Object... args) throws CmsLatteFunctionException{

String functionName = "hideName";

String protoType = String.format("string=%s(string[,markstring])",functionName);

// argument length check

if (argsLength(args.length, 1,2) == false) {

throw new CmsLatteFunctionException(functionName, protoType,

ERROR\_ARGS\_LENGTH);

}

if(!isString(args[0]) || (args.length==2 && !isString(args[1]))){

throw new CmsLatteFunctionException(functionName, protoType, ERROR\_ARGTYPE\_MISMATCH);

}

String name = (String)args[0];

String hideChar = "\*";

if(args.length==2){

hideChar = (String)args[1];

}

return new String(new char[4]).replace("\0", hideChar) + name.substring(4);

}

}

부록3 XML에 변수들 기술

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Variables>

<Constant>

<var id=*"a"* type=*"String"*>

<![CDATA[

<p>The pug snoring on the couch next to me is

<em>extremely</em> cute</p>

]]>

</var>

<var id=*"b"* type=*"Integer"*><![CDATA[13]]></var>

</Constant>

<Array>

<var id=*"array1"*>

<value index=*"0"* type=*"Integer"*><![CDATA[13]]></value>

<value index=*"1"* type=*"String"*><![CDATA[13]]></value>

<value index=*"2"* type=*"Date"*><![CDATA[2012-11-30]]></value>

<value index=*"3"* type=*"Double"*><![CDATA[2.3]]></value>

<value index=*"4"* type=*"Boolean"*><![CDATA[false]]></value>

</var>

<var id=*"array2"*>

<value index=*"0"* type=*"Integer"*><![CDATA[13]]></value>

<value index=*"1"* type=*"String"*><![CDATA[13]]></value>

</var>

</Array>

<Table>

<var id=*"table1"*>

<value row=*"0"* col=*"0"*><![CDATA[7]]></value>

</var>

<var id=*"table2"*>

<value row=*"1"* col=*"1"*><![CDATA[713]]></value>

<value row=*"10"* col=*"10"*><![CDATA[813]]></value>

</var>

</Table>

</Variables>

**주의**

table에서 <value> 태그에 type 속성이 빠져있다. type속성은 default로 String형이다.

또한 row,col,index의 default는 0 이다.

Integer,String..등 알려지지 않는 타입명은 String으로 변환한다

**주의**

Varibles,Array,Contant,Table의 첫글짜는 대문자이고, var,와 value는 모두 소문자이다

**부록4 CmsLatteExternalFunctions.xml**

<?xml version="1.0" encoding="UTF-8"?>

<CmsLatte-External-Functions>

<in-jar>

<jar file="c:/1.jar">

<class>kr.co.kalpa.A</class>

<class>kr.co.kalpa.B</class>

<class>kr.co.kalpa.C</class>

</jar>

<jar file="c:/2.jar">

<class>kr.co.kalpa.A1</class>

<class>kr.co.kalpa.B1</class>

<class>kr.co.kalpa.C1</class>

</jar>

</in-jar>

<classes>

<class>kr.co.dcos.A</class>

<class>kr.co.dcos.B</class>

<class>kr.co.dcos.C</class>

</classes>

</CmsLatte-External-Functions>