AnehtaLanguage 语法 BNF设计稿 (未定版)

**支持语法:  
    1.基本表达式运算(不支持位运算)  
        a.加法  
        b.减法  
        c.乘法  
        d.除法  
        e.自增  
        f.自减  
        g.复合加法  
        h.复合减法  
        i.复合乘法  
        j.复合除法  
        k.取幂  
        l.求模  
        m.取随机数**

**n.位运算  
    2.支持变量，函数定义与调用  
    3.支持流程结构(判断,循环,跳出)  
    4.支持递归  
    5.支持函数多返回值与多重赋值**  
  
**主语句BNF**  
<MainStatement>  := <Statement><TMP\_MainStatement>  
<TMP\_MainStatement> := EOF <Statement><TMP\_MainStatement>  
<Statement> := <FuncStatement>       |  
         <AssigmentStatement>     |  
         <IFStatement>           |  
         <VarStatement>           |  
         <BLockStatement>         |  
         <CallFuncStatement>      |

<ForStatement> |  
         EOF  
  
**函数声明语句 func(xxx,xxx,xxx)**  
<FuncStatement> := FUNC WORD LP <FuncStatement\_Define> RP <BlockStatement> CASTING

<FuncReturnType>

<FuncReturnType> := <Return\_Type\_Factor> COMMA <FuncReturnType> | <Return\_Type\_Factor>

<Return\_Factor> := WORD  
  
**函数参数声明 xxx,xxx,xxx**  
<FuncStatement\_Define> := <FuncStatement\_Define\_Factor> <TMP\_FuncStatement\_Define>

<TMP\_FuncStatement\_Define> := COMMA <FuncStatement\_Define\_Factor> <TMP\_FuncStatement\_Define>

<FuncStatement\_Define\_Factor> := VAR WORD CASTING WORD

**函数返回语句 return xxx,xxx,xxx**  
<FuncStatement\_Return>  := RETURN <Arithmetic\_Expression> <TMP\_FuncStatement\_Return>  
<TMP\_FuncStatement\_Return> := COMMA <Arithmetic\_Expression> <TMP\_FuncStatement\_Return>  
  
**赋值语句**  
<AssigmentStatement>:=WORD<TMP\_AssigmentStatement>ASSIGMENT< MoreArithmetic\_Expression>

<TMP\_AssigmentStatement> := COMMA WORD <TMP\_AssigmentStatement> |

e

**多重表达式语句 xxx,xxx,xxx,xxx**

<MoreArithmetic\_Expression>:=<Arithmetic\_Expression><TMP\_MoreArithmetic\_Expression\_Factor>

<TMP\_MoreArithmetic\_Expression>:=COMMA<Arithmetic\_Expression><TMP\_MoreArithmetic\_Expression\_Factor>|e

**判断语句**  
<IFStatement> := IF LP <Boolean\_Expression> RP <BlockStatement> <IFStatement\_ELSE>  
<IFStatement\_ELSE> := ELSE <BlockStatement>                    |  
       ELSEIF LP <Boolean\_Expression> RP <BlockStatement> <IFStatement\_ELSE>|

e

**变量声明语句**  
<VarStatement> := VAR WORD -> TYPE|  
          VAR <AssigmentStatement>  
  
**块语句**  
<BlockStatement> := LBRACE <Main\_Statement> RBRACE  
  
**函数调用语句**  
<CallFuncStatement>  := WORD LP <CallFuncStatement\_Arg> RP  
<CallFuncStatement\_Arg>  := <Arithmetic\_Expression> <TMP\_sCallFuncStatement\_Arg>  
<TMP\_CallFuncStatement\_Arg> := COMMA <Arithmetic\_Expression> <TMP\_CallFuncStatement\_Arg> |  
                e

**循环语句**

<ForStatement>:=FOR LP <ForStatement\_Assigment> SEMICOLON <Boolean\_Expression> SEMICOLON <ForStatement\_Assigment> <BlockStatement>

<ForStatement\_Assigment>:=<AssigmentStatement> |

<VarStatement>

**判断表达式**  
<Boolean\_Expression>  := <Boolean\_Expression\_Factor> <TMP\_Boolean\_Expression>  
  
<TMP\_Boolean\_Expression> := ALSO<Boolean\_Expression\_Factor> <TMP\_Boolean\_Expression>  |  
              PERHAPS <Boolean\_Expression\_Factor> <TMP\_Boolean\_Expression> |  
              e

<Boolean\_Expression\_Factor> := <Arithmetic\_Expression> GT <Arithmetic\_Expression> |  
                <Arithmetic\_Expression> LT <Arithmetic\_Expression> |  
                 <Arithmetic\_Expression> GTEQ <Arithmetic\_Expression>|  
                 <Arithmetic\_Expression> LTEQ <Arithmetic\_Expression>|  
                 LP <Boolean\_Expression> RP                                  
 **基本表达式**  
<Arithmetic\_Expression> := <Arithmetic\_Expression\_Term> <TMP\_Arithmetic\_Expression> |  
  
  
<TMP\_Arithmetic\_Expression>:=ADD <Arithmetic\_Expression\_Term> <TMP\_Arithmetic\_Expression> |  
               SUB <Arithmetic\_Expression\_Term> <TMP\_Arithmetic\_Expression> |  
               RAND <Arithmetic\_Expression\_Term> <TMP\_Arithmetic\_Expression> |  
               e  
<Arithmetic\_Expression\_Term>:=<Arithmetic\_Expression\_Factor> <TMP\_Arithmetic\_Expression\_Term>  
<TMP\_Arithmetic\_Expression>:=MUL <Arithmetic\_Expression\_Factor><TMP\_Arithmetic\_Expression\_Term>|  
 DIV<Arithmetic\_Expression\_Factor> <TMP\_Arithmetic\_Expression\_Term> |

POWER<Arithmetic\_Expression\_Factor><TMP\_Arithmetic\_Expression\_Term> |

MOD <Arithmetic\_Expression\_Factor><TMP\_Arithmetic\_Expression\_Term> |

RAND<Arithmetic\_Expression\_Factor><TMP\_Arithmetic\_Expression\_Term> |  
          e  
  
<Arithmetic\_Expression\_Factor> := NUM         |  
                   WORD         |  
                   TRUE         |  
                  FALSE        |

WORD ADDSELF    |  
             WORD SUBSELF |

LP <Arithmetic\_Expression> RP |  
            <CallFuncStatement>

完成日期:2015年9月19日星期六

作者:Anehta