

这个扩展功能的pl0编译器花了我一个礼拜的时间，代码总共编写了5000行左右(代码全部自己完成，无任何抄袭)，如果你想使用我的代码，请注明出处。请尊重我，同时我也相信你作为一个程序员的良知。

本人是：华东理工大学 计072班 刘斌(kingrl)

邮箱：tryanswer@gmail.com

chinaUnix博客：<http://blog.chinaunix.net/u3/113325/>

pl0编译器功能：词法分析、语法分析、生成基于java虚拟机的字节码(.class)的目标代码文件、java \*.class 运行

开发环境:ubuntu9.10 eclipse

这个Pl0扩展编译器，支持for while repeat if-else function procedure ...

并且支持char | boolean | integer | real四种类型

以下是我写的测试用例：

注意：

程序的结构是：

```
program program_name;
    .... //将以下测试代码加入到这里
. //.是结束符号，结束符不能紧随代码，这里我没控制好
```

不足之处：

符号表我只建了一个，所以整个源程序代码中的变量声明不能相同。当时自己考虑到了这点，可是自己却犯了迷糊，居然给忘了。整个程序文件里不支持全局变量。

//const定义

```
const const_a=12,const_b=1.2,const_ch='c',const_bool=true;
```

//声明变量

```
var var_a:char,var_b:integer,var_c:real,var_d:boolean;
```

//赋值语句

```
var var_a:char,var_b:integer,var_c:real,var_d:boolean;
```

```
var_a := 'c'
```

```
var_b := 12
```

```
var_c := 1.2
```

```
var_d := true
```

```
write(var_a,var_b,var_c,var_d)
```

//begin语句

```
var var_a:char,var_b:integer,var_c:real,var_d:boolean;
```

```
begin
```

```
    var_a := 'c';
```

```
    var_b := 12;
```

```
    var_c := 1.2;
```

```
    var_d := true;
```

```
    write(var_a,var_b,var_c,var_d)
```

```
end
```

//write

```
var enter:char;
```

```
enter:='10'
```

```
write(1,enter,1.2,enter,'A',enter,true,enter)
```

```
var var_a:char,var_b:integer,var_c:real,var_d:boolean;
```

```
var_a := 'c'
```

```
var_b := 12
```

```
var_c := 1.2
```

```
var_d := true
```

```
write(enter,var_a,enter,var_b,enter,var_c,enter,var_d,enter)
```

```
write(enter,var_b*2,enter,var_a+var_b,enter,var_c-var_a,enter)
```

//read

```
var enter:char;
```

```
enter:='10'
```

```
var var_a:char,var_b:integer,var_c:real,var_d:boolean;
```

```
read(var_a,var_b,var_c,var_d)
```

```
write(enter,var_a,enter,var_b,enter,var_c,enter,var_d,enter)
```

//定义procedure子程序

```
procedure helloworld();
    var h:char,l:char,w:char,e:char,o:char,r:char,d:char;
    begin
        h:='h';
        l:='l';
        w:='w';
        o:='o';
        e:='e';
        r:='r';
        d:='d';
        write(h,e,l,l,o,w,o,r,l,d)
    end
;

procedure println(var var_a:char,var var_b:integer,var var_c:real,var var_d:boolean);
    var enter:char;
    enter:='10'
    write(enter,var_a,enter,var_b,enter,var_c,enter,var_d,enter)
;
call println('A',123,12.2,true)

function func(var var_a:char,var var_b:integer,var var_c:real,var var_d:boolean):real;
    var var_e:real;
    var_e := var_a+var_b+var_c+var_d
;
return var_e
var var_f:real;
var_f := call func('A',1,1.2,true) //函数返回值可以赋值给变量
write(var_f)
```

//求和子程序

```
function Sum(var fromV:integer,var toV:integer):integer;
    var sum:integer;
    sum := 0
    var i:integer;
    //for 循环
    for i:=fromV to toV do //增加了变量赋值
    begin
        sum := sum + i
    end;
;
return sum
var var_sum:integer;
var_sum := call Sum(1,101)
write(var_sum)
```

//NB 的 for 循环

/\* 1. \*/

```
var space:char;
space:='32'
var i:integer;
for i:=1 to 10 do
begin
    write(i,space)
end;
```

/\* 2. \*/

```
var space:char;
space:='32'
var i:integer;
var fromV:integer,toV:integer;
fromV:=1
toV:=10
for i:=fromV to toV do
begin
    write(i,space)
```

```

end;

/* 3. */
var space:char;
space:='32'
var i:integer;
var fromV:integer,toV:real;//toV是实型
fromV:=1
toV:=7.2
for i:=fromV to toV do
begin
    write(i,space)
end;

/* 4. */
var space:char;
space:='32'
var i:real;
var fromV:real,toV:real;//fromV toV是实型
fromV:=1.2
toV:=7.2
for i:=fromV to toV do
begin
    write(i,space)
end;
//在for循环结构中, 只要不是将real 赋值给int就可以实现循环, 类型是自动转换的

//if 语句
/* 1*/
if odd true then
    write(true)
else
    write(false)
/* 2. */
if odd -1 then //非0即为真
    write(true)
else
    write(false)
/* 3. */
var a:integer;
var b:integer;
a:= 10
b:= 1
if a>b then
    write(a)
else
    write(b)
//交换a b 的子程序
procedure swap(var a:integer,var b:integer);
var temp:integer;
var space:char;
space:='32'
if a>b then
begin
    temp := a;
    a := b;
    b := temp
end
else
begin
    write(space);//嵌套的If语句
    if a=b then
        write(1,space)
    else
        write(0,space)
    end
end
write(a,space,b)

```

```

;
call swap(3,2)
call swap(2,3)
//if 不——列举

//while 语句
//打印abc子程序
procedure printABC();
    var a:char;
    a:='A'
    while a<>'Z' do
        begin
            write(a);
            a:=a+1
        end;
    call printABC()
//嵌套while子程序
procedure emWhile();
    var i:integer;
    i:=1
    var space:char;
    space:='32'
    var t:integer;
    t:=-1
    while i<10 do
        begin
            write(i);
            write(space);
            i:=i+1;
            while i<5 do
                begin
                    write(space);
                    write(t);
                    i:=i+1
                end
            end
        end
    ;
    call emWhile()

//repeat 语句
var a:integer;
var b:integer;
a := 2
b := 10
repeat
    begin
        write(a);
        a:=a+1
    end
until a>b

//子程序的嵌套调用
function Sum(var fromV:integer,var toV:integer):integer;
    var sum:integer;
    sum := 0
    var i:integer;
    //for 循环
    for i:=fromV to toV do //增加了变量赋值
        begin
            sum := sum + i
        end;
    ;
    return sum
procedure printABC();
    var a:char;
    a:='A'

```

```

    while a<>'Z' do
        begin
            write(a);
            a:=a+1
        end
    var var_sum:integer;
    var_sum := call Sum(1,101)
    write(var_sum)
;
call printABC()

//表达式支持复杂的运算，条件运算也支持复杂的运算
var a:real;
var b:integer;
var c:char;
b := 1
c := 'A'
a := (b*2)/2+c-2*c+1
write(a)

//计算1*2*3*4....
function I(var n:integer):integer;
var i:integer;
var sum:integer;
sum := 1
n := n+1
for i:=1 to n do
    begin
        sum := sum * i
    end;
;
return sum
var s:integer;
s := call I(4)//1*2*3*4
write(s)

//递归子程序求1*2*3*4
function II(var n:integer):integer;
var i:integer;
var sum:integer;
var temp:integer;
var subn:integer;
if n = 1 then
    sum := 1
else
    begin
        sum := n;
        temp := n-1;
        temp := call II(temp);
        sum := sum * temp
    end
;
return sum
var s:integer;
s := call II(4) // 1*2*3*4
write(s)

//打印1 1 2 3 5 8 13 21 34 55 ..
procedure III(var n:integer);
var i1:integer;
var i2:integer;
i1 := 0
i2 := 1
var f:integer;
var i:integer;
f:=1
var space:char;

```

```
space := 32
for i:=1 to n do
begin
    write(f);
    write(space);
    f := i1+i2;
    i1 := i2;
    i2 := f
end;
;
var s:integer;
s := 10
call III(s)

// 输入控制分支
procedure printInput();
var y:char,o:char,u:char,i:char,n:char,p:char,t:char,space:char;
y:='y'
o:='o'
u:='u'
i:='i'
n:='n'
p:='p'
u:='u'
t:='t'
space:=32
write(y,o,u,space,i,n,p,u,t,space)
;
var x:integer;
read(x)
call printInput()
if x <> 'n' then
    write('n')
else
    write('y')
```

欢迎大家修改我的代码，完善它的功能，使用请注明出处。