# 目录

[1 目录 1](#_Toc513585177)

[2 语义分析测试 3](#_Toc513585178)

[2.1 测试环境与计划 3](#_Toc513585179)

[2.1.1 测试环境与依赖 3](#_Toc513585180)

[2.1.2 测试计划 3](#_Toc513585181)

[2.2 测试非数组变量引用 3](#_Toc513585182)

[2.2.1 变量未定义 3](#_Toc513585183)

[2.2.2 变量引用过程名错误 3](#_Toc513585184)

[2.2.2.1 测试1 3](#_Toc513585185)

[2.2.2.2 测试2 4](#_Toc513585186)

[2.2.2.3 测试3 5](#_Toc513585187)

[2.2.3 变量引用函数名，形式参数个数不为0 5](#_Toc513585188)

[2.2.3.1 测试1 5](#_Toc513585189)

[2.2.3.2 测试2 6](#_Toc513585190)

[2.2.3.3 测试3 7](#_Toc513585191)

[2.2.3.4 测试4 7](#_Toc513585192)

[2.2.3.5 测试5 8](#_Toc513585193)

[2.2.4 无法被引用的标识符被用作变量引用 9](#_Toc513585194)

[2.2.4.1 测试1 9](#_Toc513585195)

[2.2.4.2 测试2 9](#_Toc513585196)

[2.2.4.3 测试3 10](#_Toc513585197)

[2.3 测试数组变量引用 10](#_Toc513585198)

[2.3.1 符号表中记录的不是数组 10](#_Toc513585199)

[2.3.2 数组下标维数不对 13](#_Toc513585200)

[2.3.3 数组下标类型检查和越界检查 14](#_Toc513585201)

[2.4 测试repeat、while、if语句的条件表达式的类型检查 15](#_Toc513585202)

[2.4.1 测试1 15](#_Toc513585203)

[2.4.2 测试2 16](#_Toc513585204)

[2.5 测试for语句相关类型检查 17](#_Toc513585205)

[2.5.1 测试1 17](#_Toc513585206)

[2.5.2 测试2 18](#_Toc513585207)

[2.6 赋值语句相关类型检查（包括以赋值语句形式给出的函数返回值语句） 19](#_Toc513585208)

[2.7 测试过程调用相关语义检查 21](#_Toc513585209)

[2.7.1 测试1 21](#_Toc513585210)

[2.7.2 测试2 22](#_Toc513585211)

[2.7.3 测试3 23](#_Toc513585212)

[2.7.4 测试4 25](#_Toc513585213)

[2.7.5 测试5 25](#_Toc513585214)

[2.7.6 测试6 25](#_Toc513585215)

[2.7.7 测试7 26](#_Toc513585216)

[2.7.8 测试8 28](#_Toc513585217)

[2.7.9 测试9 29](#_Toc513585218)

[2.7.10 测试10 30](#_Toc513585219)

[2.8 测试子程序定义的宏观检查 30](#_Toc513585220)

[2.9 测试函数返回值语句存在性检查 32](#_Toc513585221)

[2.9.1 测试1 32](#_Toc513585222)

[2.9.2 测试2 32](#_Toc513585223)

[2.9.3 测试3 33](#_Toc513585224)

[2.9.4 测试4 34](#_Toc513585225)

[2.9.5 测试5 35](#_Toc513585226)

[2.9.6 测试6 36](#_Toc513585227)

[2.10 测试变量相关检查 37](#_Toc513585228)

[2.10.1 测试1 37](#_Toc513585229)

[2.10.2 测试2 38](#_Toc513585230)

[2.11 测试常量相关检查 39](#_Toc513585231)

[2.11.1 测试1 39](#_Toc513585232)

[2.11.2 测试2 40](#_Toc513585233)

[2.11.3 测试3 41](#_Toc513585234)

[2.11.4 测试4 42](#_Toc513585235)

[2.11.5 测试5 43](#_Toc513585236)

[2.12 测试函数调用相关检查 44](#_Toc513585237)

[2.12.1 测试1 44](#_Toc513585238)

[2.12.2 测试2 46](#_Toc513585239)

[2.12.3 测试3 46](#_Toc513585240)

[2.12.4 测试4 47](#_Toc513585241)

[2.12.5 测试5 48](#_Toc513585242)

[2.12.6 测试6 49](#_Toc513585243)

[2.12.7 测试7 50](#_Toc513585244)

[2.13 测试表达式相关检查 51](#_Toc513585245)

[2.13.1 测试1 51](#_Toc513585246)

[2.13.2 测试2 52](#_Toc513585247)

[2.13.3 测试3 54](#_Toc513585248)

[2.13.4 测试4 57](#_Toc513585249)

[2.13.5 测试5 59](#_Toc513585250)

[2.13.6 测试6 60](#_Toc513585251)

[2.14 测试库程序、主程序名、主程序参数相关检查 61](#_Toc513585252)

[2.14.1 测试1 61](#_Toc513585253)

[2.14.2 测试2 62](#_Toc513585254)

[2.14.3 测试3 62](#_Toc513585255)

[2.14.4 测试4 62](#_Toc513585256)

[2.15 综合测试 64](#_Toc513585257)

# 语义分析测试

## 测试环境与计划

### 测试环境与依赖

* 测试在visual studio搭建的工程中进行
* 需要词法分析程序能够处理无词法错误的程序并输出记号序列
* 需要语法分析程序能够处理无语法错误的记号序列，并输出抽象语法树

### 测试计划

* 有针对性地测试每一种成分的语义分析
* 编写包含各种语义错误的综合程序，进行最终测试，并检查语义分析遇到错误时的恢复情况

## 测试非数组变量引用

### 变量未定义

* 测试样例

1. program test(input,output);
2. begin
3. a:=a+1;//变量未定义
4. end.

* 预期结果

报变量未定义的错误

* 测试结果

1. [Undefined identifier!] <Line 3> a has not been defined.
2. [Undefined identifier!] <Line 3> a has not been defined.

* 分析

报错结果符合预期，语义分析程序继续运行

### 变量引用过程名错误

#### 测试1

* 测试样例

1. program test(input,output);
2. var a:integer;
3. begin
4. test:=a; //错误地引用了主程序名
5. a:=test; //错误地引用了主程序名
6. end.

* 预期结果

报过程名test不能被引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference] <Line 4> Procedure name "test" can't be referenced
2. [Assign statement type mismatch!] <Left at line 4, right at line 4> Left "test" type is error **while** right "a" type is integer.
3. [Invalid reference] <Line 5> Procedure name "test" can't be referenced
4. [Assign statement type mismatch!] <Left at line 5, right at line 5> Left "a" type is integer **while** right "test" type is error.

* 分析

由于过程名不能被变量引用，因此过程的type为error。报错结果符合预期，语义分析程序继续运行

#### 测试2

* 测试样例

1. program test(input,output);
2. var a:integer;
3. procedure pro;
4. begin
5. pro:=a; //在子过程的定义中错误地引用了该子过程名
6. a:=pro; //在子过程的定义中错误地引用了该子过程名
7. end;
8. begin
9. pro;
10. end.

* 预期结果

报过程名pro不能被引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference] <Line 5> Procedure name "pro" can't be referenced
2. [Assign statement type mismatch!] <Left at line 5, right at line 5> Left "pro" type is error **while** right "a" type is integer.
3. [Invalid reference] <Line 6> Procedure name "pro" can't be referenced
4. [Assign statement type mismatch!] <Left at line 6, right at line 6> Left "a" type is integer **while** right "pro" type is error.

* 分析

由于过程名不能被变量引用，因此过程的type为error。报错结果符合预期，语义分析程序继续运行

#### 测试3

* 测试样例

1. program test(input,output);
2. var a:integer;
3. procedure pro;
4. begin
5. a:=1;
6. end;
7. begin
8. a:=pro;//在子过程外错误地引用了子过程名
9. writeln(a);
10. end.

* 预期结果

报过程名pro不能被引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference!] <Line 8> "pro" is a procedure, it can't be referenced.
2. [Assign statement type mismatch!] <Left at line 8, right at line 8> Left "a" type is integer **while** right "pro" type is error.

* 分析

由于过程名不能被变量引用，因此过程的type为error。报错结果符合预期，语义分析程序继续运行

### 变量引用函数名，形式参数个数不为0

#### 测试1

* 测试用例

1. program test(input,output);//斐波那契数列，递归调用，正确的程序
2. var a:integer;
3. function fib(i:integer):integer;
4. begin
5. **if** i=0 then
6. fib:=1
7. **else**
8. begin
9. **if** i=1 then
10. fib:=1
11. **else**
12. fib:=fib(i-1)+fib(i-2);
13. end;
14. end;
15. begin
16. writeln(fib(5));
17. end.

* 预期结果

该测试样例没有语义错误，不报错

* 测试结果

语义分析程序没有报错

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试2

* 测试用例

1. program test(input,output);//斐波那契数列，递归调用，错误的程序
2. var a:integer;
3. function fib(i:integer):integer;
4. begin
5. **if** i=0 then
6. fib:=1
7. **else**
8. begin
9. **if** i=1 then
10. fib:=1
11. **else**
12. fib:=fib(i-1)+fib;//缺少实参
13. end;
14. end;
15. begin
16. writeln(fib(5));
17. end.

* 预期结果

报第12行函数调用缺少实参的错

* 测试结果

1. [Function parameter number mismatch!] <Line 12> Function "fib" should have 1 but not 0 parameters.
2. please correct your semantic error

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试3

* 测试用例

1. program test(input,output); //在子函数外出现对函数名的左值引用
2. var a:integer;
3. function fib(i:integer):integer;
4. begin
5. **if** i=0 then
6. fib:=1
7. **else**
8. begin
9. **if** i=1 then
10. fib:=1
11. **else**
12. fib:=fib(i-1)+fib(i-2);
13. end;
14. end;
15. begin
16. fib:=a; //在子函数外出现对函数名的左值引用
17. end.

* 预期结果

报子函数名不能作为左值引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference!] <Line 16> function name "fib" can't be referenced as l-value.
2. [Assign statement type mismatch!] <Left at line 16, right at line 16> Left "fib" type is error **while** right "a" type is integer.

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试4

* 测试用例

1. program test(input,output); //函数调用缺少实参(非递归)
2. var a:integer;
3. d:**char**;
4. function fun(b:integer;c:boolean):**char**;
5. begin
6. **if** c then
7. begin
8. writeln(b);
9. fun:='y'
10. end
11. **else**
12. begin
13. fun:='n';
14. end;
15. end;
16. begin
17. d:=fun; //函数调用缺少实参(非递归)
18. end.

* 预期结果

报函数调用缺少实参的错误

* 测试结果

1. [Function parameter number mismatch!] <Line 17> Function "fun" should have 2 but not 0 parameters.
2. please correct your semantic error

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试5

* 测试用例

1. program test(input,output); //正确的函数调用
2. var a:integer;
3. d:**char**;
4. e:boolean;
5. function fun(b:integer;c:boolean):**char**;
6. begin
7. **if** c then
8. begin
9. writeln(b);
10. fun:='y'
11. end
12. **else**
13. begin
14. fun:='n';
15. end;
16. end;
17. begin
18. read(a);
19. e:=a>1;
20. d:=fun(a,e);
21. writeln(d);
22. end.

* 预期结果

该测试用例正确，不报错

* 测试结果

语音分析程序没有报错

* 分析

报错结果符合预期，语义分析程序继续运行

### 无法被引用的标识符被用作变量引用

#### 测试1

* 测试样例

1. program test(input,output);//左值引用主程序参数
2. begin
3. input:=1;
4. end.

* 预期结果

报参数无法被引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference!] <Line 3> "input" is a parameter of program, it can't be referenced.
2. [Assign statement type mismatch!] <Left at line 3, right at line 3> Left "input" type is error **while** right "1" type is integer.

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试2

* 测试样例

1. program test(input,output);//右值引用主程序参数
2. begin
3. input:=output;
4. end.

* 预期结果

报参数无法被引用的错误

* 测试结果

1. [Invalid reference!] <Line 3> "input" is a parameter of program, it can't be referenced.
2. [Invalid reference!] <Line 3> "output" is a parameter of program, it can't be referenced.

* 分析

报错结果符合预期，语义分析程序继续运行

#### 测试3

* 测试样例

1. program test(input,output);
2. var a:array[1..6] of integer;
3. b:integer;
4. begin
5. a:=1;//左值不带下标引用数组
6. b:=a;//右值不带下标引用数组
7. end.

* 预期结果

报数组名无法被引用和左右类型不匹配的错误

* 测试结果

1. [Invalid reference!] <Line 5> "a" is a array, it can't be referenced.
2. [Assign statement type mismatch!] <Left at line 5, right at line 5> Left "a" type is error **while** right "1" type is integer.
3. [Invalid reference!] <Line 6> "a" is a array, it can't be referenced.
4. [Assign statement type mismatch!] <Left at line 6, right at line 6> Left "b" type is integer **while** right "a" type is error.

* 分析

报错结果符合预期，语义分析程序继续运行

## 测试数组变量引用

### 符号表中记录的不是数组

* 测试样例

1. program test(input,output);
2. var a:integer;
3. b:array[0..5] of integer;
4. function fun:integer;
5. begin
6. fun:=1;
7. a:=fun[1]; //错把函数名当数组
8. fun[1]:=a; //错把函数名当数组
9. end;
10. procedure pro;
11. begin
12. a:=pro[1]; //错把过程名当数组
13. pro[1]:=a; //错把过程名当数组
14. writeln('y');
15. end;
16. procedure pro2(var d:integer;e:**char**);
17. begin
18. a:=d[1]; //错把引用参数当数组
19. a:=e[1]; //错把传值参数当数组
20. a:=pro2[1]; //错把过程名当数组
21. d[1]:=a; //错把引用参数当数组
22. e[1]:=a; //错把传值参数当数组
23. pro2[1]:=a; //错把过程名当数组
24. end;
25. begin
26. a:=test[1]; //错把主程序名当数组
27. a:=input[1]; //错把主程序参数当数组
28. a:=fun[1]; //错把函数名当数组
29. a:=pro[1]; //错把过程名当数组
30. a:=b[1]; //正确的数组引用
31. test[1]:=a; //错把主程序名当数组
32. output[1]:=a; //错把主程序参数当数组
33. fun[1]:=a; //错把函数名当数组
34. pro[1]:=a; //错把过程名当数组
35. b[1]:=a; //正确的数组引用
36. end.

* 预期结果

报把函数、过程、非数组的函数参数当成数组的错误，以及左右类型不匹配的错误

* 测试结果

1. [Symbol kinds mismatch!] <Line 7> "fun" defined at line 4 is a (sub)program name but not a array.
2. [Assign statement type mismatch!] <Left at line 7, right at line 7> Left "a" type is integer **while** right "fun" type is error.
3. [Symbol kinds mismatch!] <Line 8> "fun" defined at line 4 is a (sub)program name but not a array.
4. [Assign statement type mismatch!] <Left at line 8, right at line 8> Left "fun" type is error **while** right "a" type is integer.
5. [Symbol kinds mismatch!] <Line 12> "pro" defined at line 10 is a (sub)program name but not a array.
6. [Assign statement type mismatch!] <Left at line 12, right at line 12> Left "a" type is integer **while** right "pro" type is error.
7. [Symbol kinds mismatch!] <Line 13> "pro" defined at line 10 is a (sub)program name but not a array.
8. [Assign statement type mismatch!] <Left at line 13, right at line 13> Left "pro" type is error **while** right "a" type is integer.
9. [Symbol kinds mismatch!] <Line 18> "d" defined at line 16 is a var parameter but not a array.
10. [Assign statement type mismatch!] <Left at line 18, right at line 18> Left "a" type is integer **while** right "d" type is error.
11. [Symbol kinds mismatch!] <Line 19> "e" defined at line 16 is a value parameter but not a array.
12. [Assign statement type mismatch!] <Left at line 19, right at line 19> Left "a" type is integer **while** right "e" type is error.
13. [Symbol kinds mismatch!] <Line 20> "pro2" defined at line 16 is a (sub)program name but not a array.
14. [Assign statement type mismatch!] <Left at line 20, right at line 20> Left "a" type is integer **while** right "pro2" type is error.
15. [Symbol kinds mismatch!] <Line 21> "d" defined at line 16 is a var parameter but not a array.
16. [Assign statement type mismatch!] <Left at line 21, right at line 21> Left "d" type is error **while** right "a" type is integer.
17. [Symbol kinds mismatch!] <Line 22> "e" defined at line 16 is a value parameter but not a array.
18. [Assign statement type mismatch!] <Left at line 22, right at line 22> Left "e" type is error **while** right "a" type is integer.
19. [Symbol kinds mismatch!] <Line 23> "pro2" defined at line 16 is a (sub)program name but not a array.
20. [Assign statement type mismatch!] <Left at line 23, right at line 23> Left "pro2" type is error **while** right "a" type is integer.
21. [Symbol kinds mismatch!] <Line 26> "test" defined at line 1 is a (sub)program name but not a array.
22. [Assign statement type mismatch!] <Left at line 26, right at line 26> Left "a" type is integer **while** right "test" type is error.
23. [Symbol kinds mismatch!] <Line 27> "input" defined at line 1 is a parameter of program but not a array.
24. [Assign statement type mismatch!] <Left at line 27, right at line 27> Left "a" type is integer **while** right "input" type is error.
25. [Symbol kinds mismatch!] <Line 28> "fun" defined at line 4 is a function but not a array.
26. [Assign statement type mismatch!] <Left at line 28, right at line 28> Left "a" type is integer **while** right "fun" type is error.
27. [Symbol kinds mismatch!] <Line 29> "pro" defined at line 10 is a procedure but not a array.
28. [Assign statement type mismatch!] <Left at line 29, right at line 29> Left "a" type is integer **while** right "pro" type is error.
29. [Symbol kinds mismatch!] <Line 31> "test" defined at line 1 is a (sub)program name but not a array.
30. [Assign statement type mismatch!] <Left at line 31, right at line 31> Left "test" type is error **while** right "a" type is integer.
31. [Symbol kinds mismatch!] <Line 32> "output" defined at line 1 is a parameter of program but not a array.
32. [Assign statement type mismatch!] <Left at line 32, right at line 32> Left "output" type is error **while** right "a" type is integer.
33. [Symbol kinds mismatch!] <Line 33> "fun" defined at line 4 is a function but not a array.
34. [Assign statement type mismatch!] <Left at line 33, right at line 33> Left "fun" type is error **while** right "a" type is integer.
35. [Symbol kinds mismatch!] <Line 34> "pro" defined at line 10 is a procedure but not a array.
36. [Assign statement type mismatch!] <Left at line 34, right at line 34> Left "pro" type is error **while** right "a" type is integer.

* 分析

报错结果符合预期，语义分析程序继续运行

### 数组下标维数不对

* 测试样例

1. program test(input,output);
2. var a: array[0..5,6..10,11..15] of integer;
3. b: integer;
4. begin
5. a[0]:=b;
6. b:=a[0, 6];
7. a[0, 6, 11]:=b;
8. b:=a[0, 6, 11, 16];
9. end.

* 预期结果

报数组下标维数不对的错误

* 测试结果

1. [Array index number mismatch!] <Line 5> Array "a" should have 3 but not 1 indices.
2. [Array index number mismatch!] <Line 6> Array "a" should have 3 but not 2 indices.
3. [Array index number mismatch!] <Line 8> Array "a" should have 3 but not 4 indices.

* 分析

测试结果符合预期，语义分析程序继续运行

### 数组下标类型检查和越界检查

* 测试用例

1. program test(input,output);
2. **const** e=10;
3. f=20;
4. var a: array[0..5,6..10,11..15] of integer;
5. b,c: integer;
6. d: **char**;
7. begin
8. a[d,b>c,b+c]:=b;//表达式类型不为integer
9. b:=a[e-f, 6-3, 20];//下标越界 常量表达式也可以计算出结果
10. a[e+f, e\*f, e/f]:=b;
11. b:=a[e mod f, e div f, e+f\*e];
12. end.

* 预期结果

报数组下标类型非integer、数组下标越界的错误

* 测试结果

1. [Expression type error!] <Line 8> Expression "d" used **for** 1th index of array "a" should be integer but not **char**.
2. [Expression type error!] <Line 8> Expression "b > c" used **for** 2th index of array "a" should be integer but not boolean.
3. [Array range out of bound!] <Line 9> The value of expression "e - f" is -10, but the range of array "a" 0th index is 0 to 5.
4. [Array range out of bound!] <Line 9> The value of expression "6 - 3" is 3, but the range of array "a" 1th index is 6 to 10.
5. [Array range out of bound!] <Line 9> The value of expression "20" is 20, but the range of array "a" 2th index is 11 to 15.
6. [Array range out of bound!] <Line 10> The value of expression "e + f" is 30, but the range of array "a" 0th index is 0 to 5.
7. [Array range out of bound!] <Line 10> The value of expression "e \* f" is 200, but the range of array "a" 1th index is 6 to 10.
8. [Array range out of bound!] <Line 10> The value of expression "e / f" is 0, but the range of array "a" 2th index is 11 to 15.
9. [Array range out of bound!] <Line 11> The value of expression "e mod f" is 10, but the range of array "a" 0th index is 0 to 5.
10. [Array range out of bound!] <Line 11> The value of expression "e div f" is 0, but the range of array "a" 1th index is 6 to 10.
11. [Array range out of bound!] <Line 11> The value of expression "e + f \* e" is 210, but the range of array "a" 2th index is 11 to 15.

* 分析

测试结果符合预期，语义分析程序继续运行

## 测试repeat、while、if语句的条件表达式的类型检查

### 测试1

* 测试用例

1. program test(input,output);//正确的程序
2. var a,b:integer;
3. c:boolean;
4. begin
5. a:=1;
6. repeat
7. begin
8. writeln(a);
9. a:=a+1
10. end
11. until a=10;
12. **while** a<=20 **do**
13. begin
14. writeln(a);
15. a:=a+1
16. end;
17. **if** a=20 then
18. begin
19. a:=a+1;
20. writeln(a)
21. end
22. **else**
23. begin
24. a:=a+10;
25. writeln(a)
26. end;
27. end.

* 预测结果

这是一个正确的测试样例，不报错

* 测试结果

语义分析程序没有报错

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试2

* 测试用例

1. program test(input,output);//正确的程序
2. var a,b:integer;
3. c:boolean;
4. d:real;
5. e:**char**;
6. begin
7. a:=1;
8. repeat
9. begin
10. writeln(a);
11. a:=a+1
12. end
13. until a+b; //条件表达式类型错误
14. **while** e **do** //条件表达式类型错误
15. begin
16. writeln(a);
17. a:=a+1
18. end;
19. **if** c=d then //条件表达式类型错误
20. begin
21. a:=a+1;
22. writeln(a)
23. end
24. **else**
25. begin
26. a:=a+10;
27. writeln(a)
28. end;
29. end.

* 预测结果

报until、if、while条件表达式类型错误

* 测试结果

1. [Expression **type** error!] <Line 13> Expression "a + b" used **for** condition **of** **repeat**-**until** statement should be **boolean** but **not** **integer**.
2. [Expression **type** error!] <Line 14> Expression "e" used **for** condition **of** **while** statement should be **boolean** but **not** **char**.
3. [Operands expression **type** mismatch!] <Left at line 19, right at line 19> Left "c" **type** **is** **boolean** **while** right "d" **type** **is** **real**.
4. [Expression **type** error!] <Line 19> Expression "c = d" used **for** condition **of** **if** statement should be **boolean** but **not** error.

* 分析

测试结果符合预期，语义分析程序继续运行

## 测试for语句相关类型检查

### 测试1

* 测试用例

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. c:**char**;
4. **begin**
5. **for** a:=6 **to** 1+3 **do**
6. **writeln**(b);
7. **for** a:=c **to** a+b **do** //start表达式不是integer类型
8. **writeln**(a);
9. **for** b:=b **to** a>b **do** //end表达式不是integer类型
10. **writeln**(a+b)
11. **end**.

* 预期结果

报start表达式不是integer而是char、bool类型的错误

* 测试结果

1. [Expression **type** error!] <Line 7> Expression "c" used **for** start value **of** **for** statement should be **integer** but **not** **char**.
2. [Expression **type** error!] <Line 9> Expression "a > b" used **for** **end** value **of** **for** statement should be **integer** but **not** **boolean**.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试2

* 测试用例

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. c:**char**;
5. **function** fun:**integer**;
6. **begin**
7. **for** fun:=1 **to** 3 **do** //错把函数名当循环变量
8. writlen('y')
9. **end**;
11. **procedure** pro(**var** d:**integer**;e:**char**);
12. **begin**
13. **for** pro:=1 **to** 3 **do** //错把过程名当循环变量
14. **writeln**(a\*b);
15. **for** d:=1 **to** 3 **do** //正确
16. **writeln**(a/b);
17. **for** e:=1 **to** 3 **do** //错把char变量当循环变量
18. writlen(a **mod** b);
19. **for** fun:=1 **to** 3 **do** //错把函数名当循环变量
20. **writeln**(a **div** b)
21. **end**;
23. **begin**
24. **for** test:= 1 **to** 3 **do** //错把程序名当循环变量
25. **writeln**(a);
26. **for** input:=1 **to** 3 **do** //错把主程序参数当循环变量
27. **writeln**(b);
28. **for** c:=1 **to** 3 **do** //错把char变量当循环变量
29. **writeln**(c);
30. **for** fun:=1 **to** 3 **do** //错把函数名当循环变量
31. **writeln**(a+b);
32. **for** pro:=1 **to** 3 **do** //错把过程名当循环变量
33. writlen(a-b);
34. **for** d:=1 **to** 3 **do** //循环变量未定义
35. **writeln**(a\*b)
36. **end**.

* 预期结果

报把函数、过程、主程序、函数参数、字符当做循环变量以及循环变量未定义的错误

* 测试结果

1. [Symbol kinds mismatch!] <Line 7> "fun" defined at line 5 **is** a (sub)**program** name but **not** a value parameter, **var** parameter **or** normal variant.
2. [Symbol kinds mismatch!] <Line 13> "pro" defined at line 11 **is** a (sub)**program** name but **not** a value parameter, **var** parameter **or** normal variant.
3. [Usage **type** error!] <Line 17> "e" used **for** cyclic variable **of** **for** statement should be **integer** but **not** **char**.
4. [Symbol kinds mismatch!] <Line 19> "fun" defined at line 5 **is** a **function** but **not** a value parameter, **var** parameter **or** normal variant.
5. [Symbol kinds mismatch!] <Line 24> "test" defined at line 1 **is** a (sub)**program** name but **not** a value parameter, **var** parameter **or** normal variant.
6. [Symbol kinds mismatch!] <Line 26> "input" defined at line 1 **is** a parameter **of** **program** but **not** a value parameter, **var** parameter **or** normal variant.
7. [Usage **type** error!] <Line 28> "c" used **for** cyclic variable **of** **for** statement should be **integer** but **not** **char**.
8. [Symbol kinds mismatch!] <Line 30> "fun" defined at line 5 **is** a **function** but **not** a value parameter, **var** parameter **or** normal variant.
9. [Symbol kinds mismatch!] <Line 32> "pro" defined at line 11 **is** a **procedure** but **not** a value parameter, **var** parameter **or** normal variant.
10. [Undefined identifier!] <Line 34> d has **not** been defined.

* 分析

测试结果符合预期，语义分析程序继续运行

## 赋值语句相关类型检查（包括以赋值语句形式给出的函数返回值语句）

* 测试用例

1. **program** test(input,output);
2. **var** a:**integer**;
3. b:**real**;
4. c:**boolean**;
5. d:**char**;
6. **function** fib(i:**integer**):**integer**;
7. **begin**
8. **if** i=0 **then**
9. fib:='a' //表达式类型为char，但是返回值类型为integer
10. **else**
11. **begin**
12. **if** i=1 **then**
13. fib:=1 //正确
14. **else**
15. fib:=fib(i-1)+fib(i-2); //正确
16. **end**;
17. **end**;
18. **begin**
19. a:=a\*2; //正确
20. a:=b; //左值类型为integer，右值类型为real，错误
21. b:=a; //左值类型为real, 右值类型为integer，正确
22. a:=a+b; //左值类型为integer，右值类型为real，错误
23. b:=a+b; //左值类型为real，右值类型为integer，正确
24. a:=c; //左值类型为integer，右值类型为boolean，错误
25. c:=a; //左值类型为boolean，右值类型为integer，错误
26. a:=d; //左值类型为integer，右值类型为char，错误
27. d:=a; //左值类型为char，右值类型为integer，错误
28. b:=c; //左值类型为real，右值类型为boolean，错误
29. c:=b; //左值类型为boolean，右值类型为real，错误
30. b:=d; //左值类型为real，右值类型为char，错误
31. d:=b; //左值类型为char，右值类型为real，错误
32. c:=d; //左值类型为boolean，右值类型为char，错误
33. d:=c; //左值类型为char，右值类型为boolean，错误
34. **end**.

* 预期结果

报函数返回值类型和表达式类型不匹配、赋值语句左右表达式类型不匹配的错误

* 测试结果

1. [Return **type** **of** funciton mismatch!] <Line 9> The **type** **of** return expression **is** **char** ,but **not** **integer** **as** **function** "fib" defined.
2. [Assign statement **type** mismatch!] <Left at line 20, right at line 20> Left "a" **type** **is** **integer** **while** right "b" **type** **is** **real**.
3. [Assign statement **type** mismatch!] <Left at line 22, right at line 22> Left "a" **type** **is** **integer** **while** right "a + b" **type** **is** **real**.
4. [Assign statement **type** mismatch!] <Left at line 24, right at line 24> Left "a" **type** **is** **integer** **while** right "c" **type** **is** **boolean**.
5. [Assign statement **type** mismatch!] <Left at line 25, right at line 25> Left "c" **type** **is** **boolean** **while** right "a" **type** **is** **integer**.
6. [Assign statement **type** mismatch!] <Left at line 26, right at line 26> Left "a" **type** **is** **integer** **while** right "d" **type** **is** **char**.
7. [Assign statement **type** mismatch!] <Left at line 27, right at line 27> Left "d" **type** **is** **char** **while** right "a" **type** **is** **integer**.
8. [Assign statement **type** mismatch!] <Left at line 28, right at line 28> Left "b" **type** **is** **real** **while** right "c" **type** **is** **boolean**.
9. [Assign statement **type** mismatch!] <Left at line 29, right at line 29> Left "c" **type** **is** **boolean** **while** right "b" **type** **is** **real**.
10. [Assign statement **type** mismatch!] <Left at line 30, right at line 30> Left "b" **type** **is** **real** **while** right "d" **type** **is** **char**.
11. [Assign statement **type** mismatch!] <Left at line 31, right at line 31> Left "d" **type** **is** **char** **while** right "b" **type** **is** **real**.
12. [Assign statement **type** mismatch!] <Left at line 32, right at line 32> Left "c" **type** **is** **boolean** **while** right "d" **type** **is** **char**.
13. [Assign statement **type** mismatch!] <Left at line 33, right at line 33> Left "d" **type** **is** **char** **while** right "c" **type** **is** **boolean**.

* 分析

real类型不能隐式转换为integer，而integer可以隐式转换为real；integer、real和bool类型不存在隐式转换。测试结果符合预期，语义分析程序继续运行

## 测试过程调用相关语义检查

### 测试1

* 测试用例：将非过程作为过程调用，或者过程未定义

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. c:**char**;
4. f:**array**[1..5] **of** **integer**;
5. **function** fun:**integer**;
6. **begin**
7. fun:=1;
8. **end**;
9. **procedure** pro2(**var** d:**real**; e:**boolean**);
10. **begin**
11. d; //错把引用参数当成函数调用
12. e; //错把传值参数当成函数调用
13. pro2(d,e); //正确
14. **end**;
15. **begin**
16. pro; //过程未定义
17. pro2(a, a>b); //正确
18. test; //将主程序名错当为过程调用
19. input; //将主程序参数错当为过程调用
20. a; //将integer变量错当成过程调用
21. c; //将char变量错当成过程调用
22. fun; //将函数错当成过程调用
23. f; //将数组错当成过程调用
24. **end**.

* 预期结果

报将非过程错当为过程调用和过程未定义的错误

* 测试结果

1. [Symbol kinds mismatch!] <Line 11> "d" defined at line 9 **is** a **var** parameter but **not** a **procedure**.
2. [Symbol kinds mismatch!] <Line 12> "e" defined at line 9 **is** a value parameter but **not** a **procedure**.
3. [Undefined identifier!] <Line 16> pro has **not** been defined.
4. [Expression **type** error!] <Line 17> Expression "a" used **for** 1th actual parameter **of** **procedure** call **of** "pro2" should be **real** but **not** **integer**.
5. [Symbol kinds mismatch!] <Line 18> "test" defined at line 1 **is** a (sub)**program** name but **not** a **procedure**.
6. [Symbol kinds mismatch!] <Line 19> "input" defined at line 1 **is** a parameter **of** **program** but **not** a **procedure**.
7. [Symbol kinds mismatch!] <Line 20> "a" defined at line 2 **is** a normal variant but **not** a **procedure**.
8. [Symbol kinds mismatch!] <Line 21> "c" defined at line 3 **is** a normal variant but **not** a **procedure**.
9. [Symbol kinds mismatch!] <Line 22> "fun" defined at line 5 **is** a **function** but **not** a **procedure**.
10. [Symbol kinds mismatch!] <Line 23> "f" defined at line 4 **is** a **array** but **not** a **procedure**.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试2

* 测试用例：exit语句相关检查

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. c:**char**;
4. e:**real**;
5. f:**array**[1..5] **of** **integer**;
6. **function** fun:**real**;
7. **begin**
8. exit(f[1]); //正确，integer到real的隐式类型转换
9. exit(a); //正确，integer到real的隐式类型转换
10. exit(c); //错误，定义的返回值类型为real，而返回值表达式的类型为char
11. exit(e); //正确，返回值类型一直
12. exit; //错误，缺少返回值表达式
13. exit(a,b); //错误，返回值表达式多余
14. **end**;
15. **procedure** pro;
16. **begin**
17. exit(1); //错误，返回值表达式多余
18. exit; //正确，过程不需要返回值
19. **end**;
20. **begin**
21. exit(1); //错误
22. exit; //正确，主程序不需要返回值
23. **end**.

* 预期结果

报函数返回值类型错误和过程、主程序不需要返回值的错误

* 测试结果

1. [Return **type** **of** funciton mismatch!] <Line 10> The **type** **of** return expression **is** **char** ,but **not** **real** **as** **function** "fun" defined.
2. [Return value missing!] <Line 12> Number **of** return value **of** **function** must be 1, that **is**, exit must have 1 actual parameters.
3. [Return value redundancy!] <Line 13> Number **of** return value **of** **function** must be 1, that **is**, exit must have 1 actual parameters.
4. [Return value redundancy!] <Line 17> Number **of** return value **of** **procedure** must be 0, that **is**, exit must have no actual parameters.
5. [Return value redundancy!] <Line 21> Number **of** return value **of** **procedure** must be 0, that **is**, exit must have no actual parameters.

* 分析

exit语句是PASCAL\_S支持的库函数之一，用于返回函数的返回值，返回的表达式类型错误、多余、缺少等都会报错。测试结果符合预期，语义分析程序继续运行

### 测试3

* 测试用例：read语句相关检查 read的参数只能是变量或数组元素

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. c:**char**;
4. e:**real**;
5. f:**array**[1..5] **of** **integer**;
6. **function** fun:**integer**;
7. **begin**
8. fun:=1;
9. **end**;
10. **procedure** pro(**var** d:**integer**;h:**char**);
11. **var** i:**real**;
12. j:**boolean**;
13. **begin**
14. read(d,h,i,j); //正确
15. read(a,c,e,f[1]); //正确
16. read(f); //错误，数组名不能作为read的参数
17. read(test); //错误，主程序名不能作为read的参数
18. read(input); //错误，主程序参数不能作为read的参数
19. read(pro); //错误，过程名不能作为read的参数
20. read(fun); //错误，函数名不能作为read的参数
21. **end**;
22. **begin**
23. read(pro); //错误，过程名不能作为read的参数
24. read(fun); //错误，函数名不能作为read的参数
25. read(f); //错误，数组名不能作为read的参数
26. read(a,b,c,e,f[1]); //正确
27. **end**.

* 预期结果

报非变量、非数组元素不能作为read函数参数的错误

* 测试结果

1. [Read **boolean** error!] <Line 14> The 4th actual parameter **of** read "j" **is** **boolean**, it can't be read.
2. [Invalid reference!] <Line 16> "f" **is** a **array**, it can't be referenced.
3. [Actual parameter **of** read **procedure** **type** error!] <Line 16> "read" 1th expression parameter "f" **is** **not** a variant **or** an **array** element.
4. [Invalid reference] <Line 17> Procedure name "test" can't be referenced
5. [Actual parameter **of** read **procedure** **type** error!] <Line 17> "read" 1th expression parameter "test" **is** **not** a variant **or** an **array** element.
6. [Invalid reference!] <Line 18> "input" **is** a parameter **of** **program**, it can't be referenced.
7. [Actual parameter **of** read **procedure** **type** error!] <Line 18> "read" 1th expression parameter "input" **is** **not** a variant **or** an **array** element.
8. [Invalid reference] <Line 19> Procedure name "pro" can't be referenced
9. [Actual parameter **of** read **procedure** **type** error!] <Line 19> "read" 1th expression parameter "pro" **is** **not** a variant **or** an **array** element.
10. [Actual parameter **of** read **procedure** **type** error!] <Line 20> "read" 1th expression parameter "fun" **is** **not** a variant **or** an **array** element.
11. [Invalid reference!] <Line 23> "pro" **is** a **procedure**, it can't be referenced.
12. [Actual parameter **of** read **procedure** **type** error!] <Line 23> "read" 1th expression parameter "pro" **is** **not** a variant **or** an **array** element.
13. [Actual parameter **of** read **procedure** **type** error!] <Line 24> "read" 1th expression parameter "fun" **is** **not** a variant **or** an **array** element.
14. [Invalid reference!] <Line 25> "f" **is** a **array**, it can't be referenced.
15. [Actual parameter **of** read **procedure** **type** error!] <Line 25> "read" 1th expression parameter "f" **is** **not** a variant **or** an **array** element.

* 分析

read库函数的参数只能是var或者数组的元素，不能是主程序名、过程名、参数等。测试结果符合预期，语义分析程序继续运行

### 测试4

* 测试用例：read和write在调用时至少得有一个实参

1. **program** test(input,output);
2. **begin**
3. read;
4. **write**;
5. **end**.

* 预期结果

报库函数read、write至少包含一个参数的错误

* 测试结果

1. [Read actual parameter missing!] <Line 3> **procedure** "read" must have at least one actual parameter.
2. [Write actual parameter missing!] <Line 4> **procedure** "**write**" must have at least one actual parameter.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试5

* 测试用例：writeln可以不带参数

1. **program** test(input,output);
2. **begin**
3. **writeln**;
4. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序没有报错

* 分析

库函数writeln可以没有参数。测试结果符合预期，语义分析程序继续运行。

### 测试6

* 测试用例：过程调用时实参与形参的对应检查

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. **procedure** pro(**var** c:**integer**;d:**char**);
4. **begin**
5. **writeln**(c);
6. **writeln**(d);
7. **end**;
8. **begin**
9. pro; //缺少实参
10. pro(a); //缺少实参
11. pro(a,'y'); //正确
12. pro(a,b); //第二个实参类型不匹配
13. **end**.

* 预期结果

报过程pro没有实参、参数个数不够以及参数类型不匹配的错误

* 测试结果

1. [Procedure parameter number mismatch!] <Line 9> Procedure "pro" should have 2 but **not** 0 parameters.
2. [Procedure parameter number mismatch!] <Line 10> Procedure "pro" should have 2 but **not** 1 parameters.
3. [Expression **type** error!] <Line 12> Expression "b" used **for** 2th actual parameter **of** **procedure** call **of** "pro" should be **char** but **not** **integer**.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试7

* 测试用例：过程调用，表达式无法作为实参的情况

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d:**array**[1..5] **of** **integer**;
5. **function** fun:**integer**;
6. **begin**
8. **end**;
10. **procedure** pro(a,b,c:**integer**);
11. **begin**
12. **if** a<=b **then**
13. **if** b<=c **then**
14. **writeln**(1)
15. **else**
16. **writeln**(0)
17. **else**
18. **writeln**(0);
19. **end**;
21. **begin**
22. pro(a,b,c); //正确
23. pro(test,b,c); //test是主程序名
24. pro(input,b,c); //input是主程序参数
25. pro(fun,b,c); //正确
26. pro(v,b,c); //v未定义
27. pro(1,b,c); //正确
28. pro(d[1],b,c); //正确
29. **end**.

* 预期结果

报过程名、主程序参数无法作为过程的实参以及过程实参类型不正确、参数未定义的错误；报function没有返回值的警告

* 测试结果

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Here are the semantic warnings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. [Return value statement missing!] <Line 5> Incomplete return value statement **of** **function** "fun".
3. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please pay attention **to** these semantic warnings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
4. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Here are the semantic errors\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
5. [Invalid reference] <Line 23> Procedure name "test" can't be referenced
6. [Expression **type** error!] <Line 23> Expression "test" used **for** 1th actual parameter **of** **procedure** call **of** "pro" should be **integer** but **not** error.
7. [Invalid reference!] <Line 24> "input" **is** a parameter **of** **program**, it can't be referenced.
8. [Expression **type** error!] <Line 24> Expression "input" used **for** 1th actual parameter **of** **procedure** call **of** "pro" should be **integer** but **not** error.
9. [Undefined identifier!] <Line 26> v has **not** been defined.
10. [Expression **type** error!] <Line 26> Expression "v" used **for** 1th actual parameter **of** **procedure** call **of** "pro" should be **integer** but **not** error.

* 分析

过程名、未定义的变量不能作为过程的实参。测试结果符合预期，语义分析程序继续运行

### 测试8

* 测试用例

1. **program** test(input,output);
2. **const** h=5;
3. **var** d:**array**[1..5] **of** **integer**;
4. e,f,g:**integer**;
6. **procedure** pro(**var** a,b,c:**integer**);//检查a,b,c是否已经从小到大排序
7. **begin**
8. **if** a<=b **then**
9. **if** b<=c **then**
10. **writeln**(1)
11. **else**
12. **writeln**(0)
13. **else**
14. **writeln**(0);
15. **end**;
17. **begin**
18. pro(d[1],d[2],d[3]); //正确
19. pro(h,e,f); //第一个参数是常量标识符，不能作为引用参数对应的形参
20. pro(d,e,f); //第一个参数是数组名，不能作为引用参数对应的形参
21. pro(e+f,e,f); //第一个参数是复杂表达式，不能作为引用参数对应的形参
22. pro(e,e>f,f); //第二个参数是复杂表达式，不能作为引用参数对应的形参
23. pro(e,f,1); //第三个参数是常量，不能作为引用参数对应的形参
24. **end**.

* 预期结果

报常量、常量标识符、复杂表达式、数组名不能作为引用参数的错误

* 测试结果

1. [Referenced actual parameter error!] <Line 19> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
2. [Invalid reference!] <Line 20> "d" **is** a **array**, it can't be referenced.
3. [Referenced actual parameter error!] <Line 20> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
4. [Referenced actual parameter error!] <Line 21> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
5. [Referenced actual parameter error!] <Line 22> The 2th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
6. [Referenced actual parameter error!] <Line 23> The 3th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.

* 分析

过程调用，实参无法被引用调用的情况(不能是常量、不能是复杂表达式，只能是简单变量或数组元素)，测试结果符合预期，语义分析程序继续运行

### 测试9

* 测试用例

1. **program** test(input,output);
2. **var** d,e,f:**integer**;
3. **procedure** pro(**var** a:**real**;b,c:**real**);//a为引用参数, b和c为传值参数
4. **begin**
5. **if** a<=b **then**
6. **if** b<=c **then**
7. **writeln**(1)
8. **else**
9. **writeln**(0)
10. **else**
11. **writeln**(0);
12. **end**;
14. **begin**
15. pro(d,e,f); //d为integer类型，引用参数必须保证类型强一致，所以第一个实参表达式报错
16. **end**.

* 预期结果

报引用参数的实参应为real而不是integer的错误

* 测试结果

1. [Expression **type** error!] <Line 15> Expression "d" used **for** 1th actual parameter **of** **procedure** call **of** "pro" should be **real** but **not** **integer**.

* 分析

过程调用，传值参数支持从int到real的隐式类型转化，而引用参数则需类型强一致，即不支持任何类型转换。测试结果符合预期，语义分析程序继续运行

### 测试10

* 测试用例

1. **program** test(input,output);
2. **var**
3. a: **integer**;
4. b: **array** [1..5] **of** **integer**;
5. c: **integer**;
6. **begin**
7. read(a);
8. read(b);
9. read(b[1]);
10. read(a+c);
11. **end**.

* 预期结果

报read参数不能是数组名或者复杂表达式的错误

* 测试结果

1. [Invalid reference!] <Line 8> "b" **is** a **array**, it can't be referenced.
2. [Actual parameter **of** read **procedure** **type** error!] <Line 8> "read" 1th expression parameter "b" **is** **not** a variant **or** an **array** element.
3. [Actual parameter **of** read **procedure** **type** error!] <Line 10> "read" 1th expression parameter "a + c" **is** **not** a variant **or** an **array** element.

* 分析

read或readln的参数不为简单变量或数组元素。测试结果符合预期，语义分析程序继续运行

## 测试子程序定义的宏观检查

* 测试用例：子程序名已定义

1. **program** test(input,output);
2. **var** a:**integer**;
3. b:**array**[1..5] **of** **integer**;
4. **function** fun:**integer**;
5. **begin**
7. **end**;
9. **procedure** pro;
10. **begin**
12. **end**;
14. **function** fun:**char**; //已被定义为函数
15. **begin**
17. **end**;
19. **procedure** pro; //已被定义为过程
20. **begin**
22. **end**;
24. **procedure** test; //已被定义为主程序名
25. **begin**
27. **end**;
29. **procedure** input; //已被定义为主程序参数
30. **begin**
32. **end**;
34. **procedure** a; //已被定义为变量
35. **begin**
37. **end**;
39. **procedure** b; //已被定义为数组
40. **begin**
42. **end**;
44. **begin**
46. **end**.

* 预期结果

报过程名已被定义的错误

* 测试结果

1. [Duplicate defined error!] <Line 14> "fun" has already been defined **as** a **function** at line 4.
2. [Duplicate defined error!] <Line 19> "pro" has already been defined **as** a **procedure** at line 9.
3. [Duplicate defined error!] <Line 24> "test" has already been defined **as** a (sub)**program** name at line 1.
4. [Duplicate defined error!] <Line 29> "input" has already been defined **as** a parameter **of** **program** at line 1.
5. [Duplicate defined error!] <Line 34> "a" has already been defined **as** a normal variant at line 2.
6. [Duplicate defined error!] <Line 39> "b" has already been defined **as** a **array** at line 3.

* 分析

过程名已被定义。测试结果符合预期，语义分析程序继续运行

## 测试函数返回值语句存在性检查

### 测试1

* 测试用例：缺少返回值语句

1. **program** test(input,output);
2. **function** fun:**integer**;
3. **var** a,b:**integer**;
4. **begin**
5. **if** a>0 **then**
6. fun:=1
7. **else**
8. **writeln**(a); //该分支缺少返回值语句
9. **end**;
10. **begin**
11. **writeln**(fun);
12. **end**.

* 预期结果

报子函数缺少返回值语句的警告

* 测试结果

1. [Return value statement missing!] <Line 2> Incomplete return value statement **of** **function** "fun".

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试2

* 测试用例

1. **program** test(input,output);
2. **function** fun:**integer**;
3. **var** a,b:**integer**;
4. **begin**
5. **if** a>0 **then**
6. fun:=1
7. **else**
8. **begin**
9. **writeln**(a);
10. fun:=3;
11. **end**
12. **end**;
13. **begin**
14. **writeln**(fun);
15. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序没有报错

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试3

* 测试用例：缺少返回值语句

1. **program** test(input,output);
2. **function** fun(a:**integer**):**integer**;
3. **begin**
4. **if** a=1 **then**
5. **begin**
6. **writeln**('a','=','1');
7. fun:=1
8. **end**
9. **else**
10. **begin**
11. **if** a=2 **then**
12. **begin**
13. **writeln**('a','=','2');//该分支缺少返回值语句
14. **end**
15. **else**
16. **begin**
17. **writeln**('a','=','?');
18. fun:=3
19. **end**;
20. **end**;
21. **end**;
22. **begin**
23. **writeln**(fun(5));
24. **end**.

* 预期结果

报子函数没有返回值的警告

* 测试结果

1. [Return value statement missing!] <Line 2> Incomplete return value statement **of** **function** "fun".

* 分析

缺少返回值语句，测试结果符合预期，语义分析程序继续运行

### 测试4

* 测试用例

1. **program** test(input,output);
2. **function** fun(a:**integer**):**integer**;
3. **begin**
4. **if** a=1 **then**
5. **begin**
6. **writeln**('a','=','1');
7. fun:=1
8. **end**
9. **else**
10. **begin**
11. **if** a=2 **then**
12. **begin**
13. **writeln**('a','=','2');
14. fun:=2
15. **end**
16. **else**
17. **begin**
18. **writeln**('a','=','?');
19. **end**;
20. fun:=100;
21. **end**;
22. **end**;
23. **begin**
24. **writeln**(fun(5));
25. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序没有报错

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试5

* 测试用例

1. **program** test(input,output);
2. **function** fun(a:**integer**):**integer**;
3. **begin**
4. fun:=100; //这里直接返回
5. **if** a=1 **then**
6. **begin**
7. **writeln**('a','=','1');
8. fun:=1
9. **end**
10. **else**
11. **begin**
12. **if** a=2 **then**
13. **begin**
14. **writeln**('a','=','2');
15. fun:=2
16. **end**
17. **else**
18. **begin**
19. **writeln**('a','=','?');
20. **end**;
21. **end**;
22. **end**;
23. **begin**
24. **writeln**(fun(5));
25. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序没有报错

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试6

* 测试用例

1. **program** test(input,output);
2. **function** fun(a:**integer**):**integer**;
3. **begin**
4. **if** a=1 **then** //该if-else语句所有分支均包含返回值语句
5. **begin**
6. **writeln**('a','=','1');
7. fun:=1
8. **end**
9. **else**
10. **begin**
11. **if** a=2 **then**
12. **begin**
13. **writeln**('a','=','2');
14. fun:=2
15. **end**
16. **else**
17. **begin**
18. **writeln**('a','=','?');
19. fun:=100;
20. **end**;
21. **end**;
22. **if** a=3 **then**
23. **writeln**('a','=','3')
24. **else**
25. **writeln**('a','=','?','?','?');
26. **end**;
27. **begin**
28. **writeln**(fun(5));
29. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序没有报错

* 分析

测试结果符合预期，语义分析程序继续运行

## 测试变量相关检查

### 测试1

* 测试用例：变量重定义检查（作用域）

1. **program** test(input,output);
2. **var** test,input:**integer**; //test已定义为主程序名，input已定义为主程序参数
3. a:**real**; //正确
4. b:**boolean**; //正确
5. c:**array**[1..5] **of** **integer**; //正确
6. a:**char**; //a已定义为变量
7. c:**real**; //c已定义为数组
9. **function** fun(**var** a:**integer**;b:**char**):**integer**;
10. **var** fun:**integer**; //fun已定义为当前所在的函数名
11. a:**real**; //a已定义为当前所在函数的引用参数
12. b:**boolean**; //b已定义为当前所在函数的传值参数
13. **begin**
15. **end**;
17. **procedure** pro(**var** a:**integer**;b:**char**);
18. **var** fun:**integer**; //正确
19. pro:**char**; //pro已定义为当前所在的过程名
20. a:**real**; //a已定义为当前所在过程的引用参数
21. b:**boolean**; //b已定义为当前所在过程的传值参数
22. **begin**
24. **end**;
26. **begin**
28. **end**.

* 预期结果

报变量已被定义为主程序名、函数参数、变量的错误

* 测试结果

1. [Duplicate defined error!] <Line 2> "test" has been defined **as** the name **of** **program** at Line 1.
2. [Duplicate defined error!] <Line 2> "input" has been defined **as** a **program** parameter at Line 1.
3. [Duplicate defined error!] <Line 6> "a" has already been defined **as** a normal variant at line 3.
4. [Duplicate defined error!] <Line 7> "c" has already been defined **as** a **array** at line 5.
5. [Duplicate defined error!] <Line 10> "fun" has already been defined **as** a (sub)**program** name at line 9.
6. [Duplicate defined error!] <Line 11> "a" has already been defined **as** a **var** parameter at line 9.
7. [Duplicate defined error!] <Line 12> "b" has already been defined **as** a value parameter at line 9.
8. [Duplicate defined error!] <Line 19> "pro" has already been defined **as** a (sub)**program** name at line 17.
9. [Duplicate defined error!] <Line 20> "a" has already been defined **as** a **var** parameter at line 17.
10. [Duplicate defined error!] <Line 21> "b" has already been defined **as** a value parameter at line 17.

* 分析

变量重定义检查（作用域），在同一个子函数内不能定义重复名字的变量、和子函数名相同的变量和与主函数中变量名相同的变量；但是可以定义与另一个子函数的函数名或变量名相同的变量

### 测试2

* 测试用例：数组定义下界比上界大

1. **program** test(input,output);
2. **var** a:**array**[10..5] **of** **integer**; //下界比上界大
3. b:**array**[5..10] **of** **integer**; //正确
4. c:**array**[5..5] **of** **integer**; //正确
5. **begin**
7. **end**.

* 预期结果

报数组下界比上界大的错误

* 测试结果

1. [Array range upsidedown error!] <Line 2> 1th range **of** **array** "a" have larger low bound **and** smaller high bound, which **is** 10 **and** 5.

* 分析

数组定义下界比上界大，测试结果符合预期，语义分析程序继续运行

## 测试常量相关检查

### 测试1

* 测试用例：常量重定义

1. **program** test(input,output);
2. **const** test=1; //已定义为主程序名
3. input=2; //已定义为主程序参数
4. d=d; //右值未定义
5. a=test; //右值非常量
6. b=a; //右值未定义
7. c=3; //正确
8. a=b; //右值未定义
9. c=c; //已定义为常量
11. **function** fun(**var** a:**integer**;b:**char**):**integer**;
12. **const** fun=-5; //已定义为当前所在函数名
13. a=-10; //已定义为当前所在函数的引用参数
14. b=a; //已定义为当前所在函数的传值参数
15. **begin**
17. **end**;
19. **procedure** pro(**var** a:**integer**;b:**char**);
20. **const** fun=3; //正确
21. pro=-6; //已定义为当前所在过程名
22. a=fun; //左值已定义为当前所在过程的引用参数
23. b=pro; //左值已定义为当前所在过程的传值参数
24. c=fun; //正确
25. **begin**
27. **end**;
29. **begin**
31. **end**.

* 预期结果

报常量重复定义、常量右值未定义、常量右值非常量的错误

* 测试结果

1. [Duplicate defined error!] <Line 2> "test" has been defined **as** the name **of** **program** at Line 1.
2. [Duplicate defined error!] <Line 3> "input" has been defined **as** a **program** parameter at Line 1.
3. [Undefined identifier!] <Line 4> d has **not** been defined.
4. [Symbol kinds mismatch!] <Line 5> "test" defined at line 1 **is** a (sub)**program** name but **not** a constant.
5. [Undefined identifier!] <Line 6> a has **not** been defined.
6. [Undefined identifier!] <Line 8> b has **not** been defined.
7. [Duplicate defined error!] <Line 9> "c" has already been defined **as** a constant at line 7.
8. [Duplicate defined error!] <Line 12> "fun" has already been defined **as** a (sub)**program** name at line 11.
9. [Duplicate defined error!] <Line 13> "a" has already been defined **as** a **var** parameter at line 11.
10. [Duplicate defined error!] <Line 14> "b" has already been defined **as** a value parameter at line 11.
11. [Duplicate defined error!] <Line 21> "pro" has already been defined **as** a (sub)**program** name at line 19.
12. [Duplicate defined error!] <Line 22> "a" has already been defined **as** a **var** parameter at line 19.
13. [Duplicate defined error!] <Line 23> "b" has already been defined **as** a value parameter at line 19.

* 分析

const c=1；c=c；这种是不对的，相当于常量c重复定义。测试结果符合预期，语义分析程序继续运行

### 测试2

* 测试用例

1. **program** test(input,output);
2. **const** a=1; //正确
3. b=a; //正确
4. c=d; //右值未定义
5. f=test; //右值是主程序名而不是常量
6. f=input; //右值是主程序参数而不是常量
7. **var** e:**integer**;
8. g:**array**[1..5] **of** **integer**;
10. **procedure** pro;
11. **begin**
13. **end**;
15. **function** fun(**var** n:**integer**;m:**char**):**integer**;
16. **const** a=e; //右值是变量而不是常量
17. a=g; //右值是数组而不是常量
18. a=fun; //右值是当前所在函数名而不是常量
19. a=n; //右值是当前所在函数的引用参数而不是常量
20. a=m; //右值是当前所在函数的传值参数而不是常量
21. a=pro; //右值是子过程名而不是常量
22. **begin**
24. **end**;
26. **begin**
28. **end**.

* 预期结果

报常量赋值语句右值未定义、右值不是常量的错误

* 测试结果

1. [Undefined identifier!] <Line 4> d has **not** been defined.
2. [Symbol kinds mismatch!] <Line 5> "test" defined at line 1 **is** a (sub)**program** name but **not** a constant.
3. [Symbol kinds mismatch!] <Line 6> "input" defined at line 1 **is** a parameter **of** **program** but **not** a constant.
4. [Symbol kinds mismatch!] <Line 16> "e" defined at line 7 **is** a normal variant but **not** a constant.
5. [Symbol kinds mismatch!] <Line 17> "g" defined at line 8 **is** a **array** but **not** a constant.
6. [Symbol kinds mismatch!] <Line 18> "fun" defined at line 15 **is** a (sub)**program** name but **not** a constant.
7. [Symbol kinds mismatch!] <Line 19> "n" defined at line 15 **is** a **var** parameter but **not** a constant.
8. [Symbol kinds mismatch!] <Line 20> "m" defined at line 15 **is** a value parameter but **not** a constant.
9. [Symbol kinds mismatch!] <Line 21> "pro" defined at line 10 **is** a **procedure** but **not** a constant.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试3

* 测试用例：常量传播

1. **program** test(input,output);
2. **const** a = 1.5;
3. b = 3;
4. c = -a;
5. d = -b;
6. e = 2;
7. f = 5;
8. g = -c;
9. **var** h:**integer**;
10. i:**real**;
11. j:**array**[1..10] **of** **integer**;
12. **begin**
13. **writeln**(a);
14. **writeln**(b);
15. **writeln**(c);
16. **writeln**(d);
17. **writeln**(e);
18. **writeln**(f);
19. **writeln**(g);
20. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

1. 1.500000
2. 3
3. -1.500000
4. -3
5. 2
6. 5
7. 1.500000

语义分析程序没有报错，a.b.c.d.e.f.g的值如上

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试4

* 测试用例：常量传播与常量表达式的计算

1. **program** test(input,output);
2. **const** a = 1.5;
3. b = 3;
4. c = -a;
5. d = -b;
6. e = 2;
7. f = 5;
8. g = -c;
9. **var** h:**integer**;
10. i:**real**;
11. j:**array**[1..10] **of** **integer**;
12. **begin**
13. **writeln**(a);
14. **writeln**(b);
15. **writeln**(c);
16. **writeln**(d);
17. **writeln**(e);
18. **writeln**(f);
19. **writeln**(g);
20. h := h / (2+3-f); //除0错误
21. h := h **mod** (10 **mod** e); //除0错误
22. h := h **div** (1 **div** d); //除0错误
23. j[b+f]:=3; //正确
24. **writeln**(j[b+f]); //正确
25. j[-f]:=5; //越界
26. **end**.

* 预期结果

报除0和数组越界的错误

* 测试结果

1. [Divide zero error!] <Line 20> The value **of** expression "(2 + 3 - f)" **is** 0, which **is** the second operand **of** operation "/".
2. [Divide zero error!] <Line 21> The value **of** expression "(10 **mod** e)" **is** 0, which **is** the second operand **of** operation "**mod**".
3. [Divide zero error!] <Line 22> The value **of** expression "(1 **div** d)" **is** 0, which **is** the second operand **of** operation "**div**".
4. [Array range out **of** bound!] <Line 25> The value **of** expression " - f" **is** -5, but the range **of** **array** "j" 0th index **is** 1 **to** 10.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试5

* 测试用例：常量左值引用

1. **program** test(input,output);
2. **const** a = 1.5;
3. b = 3;
4. c = -a;
5. d = -b;
6. e = 2;
7. f = 5;
8. g = -c;
9. **begin**
10. a:=3; //常量不能作为左值
11. e:=6-3; //常量不能作为左值
12. g:=b+c; //常量不能作为左值
13. **end**.

* 预期结果

报常量不能作为左值的错误

* 测试结果

1. [Constant **as** l-value error!] <Line10> Costant "a" can't be referenced **as** l-value.
2. [Constant **as** l-value error!] <Line11> Costant "e" can't be referenced **as** l-value.
3. [Constant **as** l-value error!] <Line12> Costant "g" can't be referenced **as** l-value.

* 分析

常量左值引用错误，测试结果符合预期，语义分析程序继续运行

## 测试函数调用相关检查

### 测试1

* 测试用例：函数调用未定义，标识符名不是函数

1. **program** test(input,output);
2. **const** f=5;
3. **var** a,b:**integer**;
4. c:**array**[1..5] **of** **integer**;
5. **procedure** pro(**var** d:**integer**; e:**char**);
6. **begin**
7. a:=d(1); //d是当前所在过程的引用参数而不是函数
8. a:=e(1); //e是当前所在过程的传值参数而不是函数
9. a:=f(5); //f是常量而不是函数
10. **end**;
12. **begin**
13. a:=fun(1); //fun未定义
14. a:=b(1); //b是普通变量而不是函数
15. a:=c(1); //c是数组而不是函数
16. a:=test(1); //test是主程序名而不是函数
17. a:=input(1); //input是主程序参数而不是函数
18. a:=pro(1); //pro是子过程名而不是函数
19. **end**.

* 预期结果

报引用参数不是函数、传值参数不是函数、常量不是函数等错误

* 测试结果

1. [Symbol kinds mismatch!] <Line 7> "d" defined at line 5 **is** a **var** parameter but **not** a **function**.
2. [Assign statement **type** mismatch!] <Left at line 7, right at line 7> Left "a" **type** **is** **integer** **while** right "d(1)" **type** **is** error.
3. [Symbol kinds mismatch!] <Line 8> "e" defined at line 5 **is** a value parameter but **not** a **function**.
4. [Assign statement **type** mismatch!] <Left at line 8, right at line 8> Left "a" **type** **is** **integer** **while** right "e(1)" **type** **is** error.
5. [Symbol kinds mismatch!] <Line 9> "f" defined at line 2 **is** a constant but **not** a **function**.
6. [Assign statement **type** mismatch!] <Left at line 9, right at line 9> Left "a" **type** **is** **integer** **while** right "f(5)" **type** **is** error.
7. [Undefined identifier!] <Line 13> fun has **not** been defined.
8. [Assign statement **type** mismatch!] <Left at line 13, right at line 13> Left "a" **type** **is** **integer** **while** right "fun(1)" **type** **is** error.
9. [Symbol kinds mismatch!] <Line 14> "b" defined at line 3 **is** a normal variant but **not** a **function**.
10. [Assign statement **type** mismatch!] <Left at line 14, right at line 14> Left "a" **type** **is** **integer** **while** right "b(1)" **type** **is** error.
11. [Symbol kinds mismatch!] <Line 15> "c" defined at line 4 **is** a **array** but **not** a **function**.
12. [Assign statement **type** mismatch!] <Left at line 15, right at line 15> Left "a" **type** **is** **integer** **while** right "c(1)" **type** **is** error.
13. [Symbol kinds mismatch!] <Line 16> "test" defined at line 1 **is** a (sub)**program** name but **not** a **function**.
14. [Assign statement **type** mismatch!] <Left at line 16, right at line 16> Left "a" **type** **is** **integer** **while** right "test(1)" **type** **is** error.
15. [Symbol kinds mismatch!] <Line 17> "input" defined at line 1 **is** a parameter **of** **program** but **not** a **function**.
16. [Assign statement **type** mismatch!] <Left at line 17, right at line 17> Left "a" **type** **is** **integer** **while** right "input(1)" **type** **is** error.
17. [Symbol kinds mismatch!] <Line 18> "pro" defined at line 5 **is** a **procedure** but **not** a **function**.
18. [Assign statement **type** mismatch!] <Left at line 18, right at line 18> Left "a" **type** **is** **integer** **while** right "pro(1)" **type** **is** error.

* 分析

函数调用未定义，标识符名不是函数。测试结果符合预期，语义分析程序继续运行。

### 测试2

* 测试用例：函数调用参数个数不对，类型不匹配

1. **program** test(input,output);
2. **function** fun(a,b,c:**real**):**integer**;//检查a，b，c是否已经为从小到大排序
3. **begin**
4. **if** a<=b **then**
5. **if** b<=c **then**
6. fun:=1;
7. fun:=0
8. **end**;
9. **begin**
10. **writeln**(fun); //参数缺失
11. **writeln**(fun(1)); //参数缺失
12. **writeln**(fun(1,'a')); //参数缺失
13. **writeln**(fun(1,2,3)); //正确
14. **writeln**(fun('a',1.1,3)); //参数类型不一致
15. **end**.

* 预期结果

报子函数参数缺失、参数类型不一致的问题

* 测试结果

1. [Function parameter number mismatch!] <Line 10> Function "fun" should have 3 but **not** 0 parameters.
2. [Function parameter number mismatch!] <Line 11> Function "fun" should have 3 but **not** 1 parameters.
3. [Function parameter number mismatch!] <Line 12> Function "fun" should have 3 but **not** 2 parameters.
4. [Expression **type** error!] <Line 14> Expression "'a'" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **real** but **not** **char**.

* 分析

函数调用参数个数不对，类型不匹配

### 测试3

* 测试用例

1. **program** test(input,output);
2. **function** fun(a,b,c:**real**):**integer**;//检查a，b，c是否已经为从小到大排序
3. **begin**
4. **if** a<=b **then**
5. **if** b<=c **then**
6. fun:=1;
7. fun:=0
8. **end**;
9. **begin**
10. **writeln**(fun(1,2,3));
11. **writeln**(fun(3,1,2));
12. **end**.

* 预期结果

语义分析程序不报错

* 测试结果

语义分析程序不报错

* 分析

测试结果符合预期，语义分析程序继续运行。

### 测试4

* 测试用例：函数调用，表达式无法作为实参的情况

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d:**array**[1..5] **of** **integer**;
5. **procedure** pro;
6. **begin**
8. **end**;
10. **function** fun(a,b,c:**integer**):**integer**;//检查a，b，c是否已经为从小到大排序
11. **begin**
12. **if** a<=b **then**
13. **if** b<=c **then**
14. fun:=1;
15. fun:=0
16. **end**;
17. **begin**
18. **writeln**(fun(a,b,c)); //正确
19. **writeln**(fun(test,b,c)); //test是主程序名
20. **writeln**(fun(input,b,c)); //input是主程序参数
21. **writeln**(fun(pro,b,c)); //pro是子过程明
22. **writeln**(fun(v,b,c)); //v未定义
23. **writeln**(fun(1,b,c)); //正确
24. **writeln**(fun(d[1],b,c)); //正确
25. **end**.

* 预期结果

报主程序名、主程序参数、子过程名不能作为函数实参以及函数实参未定义的错

* 测试结果

1. [Invalid reference] <Line 19> Procedure name "test" can't be referenced
2. [Expression **type** error!] <Line 19> Expression "test" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **integer** but **not** error.
3. [Invalid reference!] <Line 20> "input" **is** a parameter **of** **program**, it can't be referenced.
4. [Expression **type** error!] <Line 20> Expression "input" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **integer** but **not** error.
5. [Invalid reference!] <Line 21> "pro" **is** a **procedure**, it can't be referenced.
6. [Expression **type** error!] <Line 21> Expression "pro" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **integer** but **not** error.
7. [Undefined identifier!] <Line 22> v has **not** been defined.
8. [Expression **type** error!] <Line 22> Expression "v" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **integer** but **not** error.

* 分析

函数调用，表达式无法作为实参的情况。测试结果符合预期，语义分析程序继续运行。

### 测试5

* 测试用例：函数调用，实参无法被引用调用的情况(不能是常量、不能是复杂表达式，只能是简单变量或数组元素)

1. **program** test(input,output);
2. **const** h=5;
3. **var** d:**array**[1..5] **of** **integer**;
4. e,f,g:**integer**;
5. **function** fun(**var** a,b,c:**integer**):**integer**;//检查a,b,c是否已经从小到大排序
6. **begin**
7. **if** a<=b **then**
8. **if** b<=c **then**
9. fun:=1;
10. fun:=0
11. **end**;
13. **procedure** pro(**var** a:**integer**;b:**integer**);
14. **var** c:**integer**;
15. **begin**
16. **writeln**(fun(a,b,c)); //正确
17. **end**;
18. **begin**
19. **writeln**(fun(d[1],d[2],d[3]));//正确
20. **writeln**(fun(h,e,f)); //第一个参数是常量标识符，不能作为引用参数对应的形参
21. **writeln**(fun(d,e,f)); //第一个参数是数组名，不能作为引用参数对应的形参
22. **writeln**(fun(e+f,e,f)); //第一个参数是复杂表达式，不能作为引用参数对应的形参
23. **writeln**(fun(e,e>f,f)); //第二个参数是复杂表达式，不能作为引用参数对应的形参
24. **writeln**(fun(e,f,1)); //第三个参数是常量，不能作为引用参数对应的形参
25. **end**.

* 预期结果

报函数的实参不能是常量、数组名和复杂表达式的错

* 测试结果

1. [Referenced actual parameter error!] <Line 20> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
2. [Invalid reference!] <Line 21> "d" **is** a **array**, it can't be referenced.
3. [Referenced actual parameter error!] <Line 21> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
4. [Referenced actual parameter error!] <Line 22> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
5. [Referenced actual parameter error!] <Line 23> The 2th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.
6. [Referenced actual parameter error!] <Line 24> The 3th actual parameter expression should be a normal variable、value parameter、referenced parameter **or** **array** element.

* 分析

函数调用，实参无法被引用调用的情况(不能是常量、不能是复杂表达式，只能是简单变量或数组元素) 。测试结果符合预期，语义分析程序继续运行。

### 测试6

* 测试用例：函数调用，传值参数支持从int到real的隐式类型转化，而引用参数则需类型强一致，即不支持任何类型转换

1. **program** test(input,output);
2. **var** d,e,f:**integer**;
3. **function** fun(**var** a:**real**;b,c:**real**):**integer**;//a为引用参数, b和c为传值参数
4. **begin**
5. **if** a<=b **then**
6. **if** b<=c **then**
7. fun:=1;
8. fun:=0
9. **end**;
11. **begin**
12. **writeln**(fun(d,e,f)); //d为integer类型，引用参数必须保证类型强一致，所以第一个实参表达式报错
13. **end**.

* 预期结果

报引用参数类型不匹配的错误

* 测试结果

1. [Expression **type** error!] <Line 12> Expression "d" used **for** 1th actual parameter **of** **function** call **of** "fun" should be **real** but **not** **integer**.

* 分析

函数调用，传值参数支持从int到real的隐式类型转化，而引用参数则需类型强一致，即不支持任何类型转换

### 测试7

* 测试用例：将函数单独作为一条语句

1. **program** test(input,output);
2. **var** d,e,f:**integer**;
3. **function** fun(**var** a:**real**;b,c:**real**):**integer**;//a为引用参数, b和c为传值参数
4. **begin**
5. **if** a<=b **then**
6. **if** b<=c **then**
7. fun:=1;
8. fun:=0
9. **end**;
11. **begin**
12. fun(d,e,f);//该PASCAL-S语言不支持函数的未引用调用
13. **end**.

* 预期结果

报function是一个函数而不是一个过程的错误

* 测试结果

1. [Symbol kinds mismatch!] <Line 12> "fun" defined at line 3 **is** a **function** but **not** a **procedure**.

* 分析

PASCAL\_S不支持函数单独作为一条语句，因此报错。测试结果符合预期，语义分析程序继续运行。

## 测试表达式相关检查

### 测试1

* 测试用例：关系运算符类型检查

1. **program** test(input,output);
2. **const** m = 100;
3. **var** a,b,c:**integer**;
4. d,e,f:**real**;
5. g,h,i:**char**;
6. j,k,l:**boolean**;
8. **function** fun:**real**;
9. **begin**
10. fun:=1;
11. **end**;
13. **begin**
14. j:=a>=b; //正确
15. j:=d>e; //正确
16. j:=g<=h; //正确
17. j:=j<k; //正确
18. j:=a<>d; //正确，隐式类型转换
19. j:=a=m; //正确
20. j:=a=fun; //正确，隐式类型转换
21. j:=a=g; //错误，a和g类型不匹配
22. k:=a>=j; //错误，a和j类型不匹配
23. k:=d>g; //错误，d和g类型不匹配
24. k:=d<=j; //错误，d和j类型不匹配
25. k:=g<j; //错误，g和j类型不匹配
26. **end**.

* 预期结果

报real、integer、char和boolean类型不匹配，integer、real与char类型不匹配

* 测试结果

1. [Operands expression **type** mismatch!] <Left at line 21, right at line 21> Left "a" **type** **is** **integer** **while** right "g" **type** **is** **char**.
2. [Assign statement **type** mismatch!] <Left at line 21, right at line 21> Left "j" **type** **is** **boolean** **while** right "a = g" **type** **is** error.
3. [Operands expression **type** mismatch!] <Left at line 22, right at line 22> Left "a" **type** **is** **integer** **while** right "j" **type** **is** **boolean**.
4. [Assign statement **type** mismatch!] <Left at line 22, right at line 22> Left "k" **type** **is** **boolean** **while** right "a >= j" **type** **is** error.
5. [Operands expression **type** mismatch!] <Left at line 23, right at line 23> Left "d" **type** **is** **real** **while** right "g" **type** **is** **char**.
6. [Assign statement **type** mismatch!] <Left at line 23, right at line 23> Left "k" **type** **is** **boolean** **while** right "d > g" **type** **is** error.
7. [Operands expression **type** mismatch!] <Left at line 24, right at line 24> Left "d" **type** **is** **real** **while** right "j" **type** **is** **boolean**.
8. [Assign statement **type** mismatch!] <Left at line 24, right at line 24> Left "k" **type** **is** **boolean** **while** right "d <= j" **type** **is** error.
9. [Operands expression **type** mismatch!] <Left at line 25, right at line 25> Left "g" **type** **is** **char** **while** right "j" **type** **is** **boolean**.
10. [Assign statement **type** mismatch!] <Left at line 25, right at line 25> Left "k" **type** **is** **boolean** **while** right "g < j" **type** **is** error.

* 分析

integer与real类型可以隐式转换，其他不行

### 测试2

* 测试用例：not、and、or运算符

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d,e,f:**real**;
4. g,h,i:**char**;
5. j,k,l:**boolean**;
7. **function** fun1:**integer**;
8. **begin**
9. fun1:=1;
10. **end**;
12. **function** fun2:**boolean**;
13. **begin**
14. fun2:=2>1;
15. **end**;
17. **begin**
18. j:= **not** a; //a是integer而不是boolean
19. j:= **not** d; //d是real而不是boolean
20. j:= **not** g; //g是char而不是boolean
21. j:= **not** j; //正确
22. j:= **not** fun1; //fun1是integer而不是boolean
23. j:= **not** fun2; //正确
24. j:= j **and** k; //正确
25. j:= a **and** l; //a是integer而不是boolean
26. j:= g **and** h; //g是char而不是boolean，h是char而不是boolean
27. j:= j **or** k; //正确
28. j:= a **or** l; //a是integer而不是boolean
29. j:= g **or** h; //g是char而不是boolean，h是char而不是boolean
30. j:= ((a+b)>c) **or** k; //正确
31. **end**.

* 预期结果

报integer、char、real类型与bool不匹配的错误

* 测试结果

1. [Operand expression **type** error!] <Line 18> Expression "a" **type** should be **boolean** but **not** **integer**.
2. [Assign statement **type** mismatch!] <Left at line 18, right at line 18> Left "j" **type** **is** **boolean** **while** right " **not** a" **type** **is** error.
3. [Operand expression **type** error!] <Line 19> Expression "d" **type** should be **boolean** but **not** **real**.
4. [Assign statement **type** mismatch!] <Left at line 19, right at line 19> Left "j" **type** **is** **boolean** **while** right " **not** d" **type** **is** error.
5. [Operand expression **type** error!] <Line 20> Expression "g" **type** should be **boolean** but **not** **char**.
6. [Assign statement **type** mismatch!] <Left at line 20, right at line 20> Left "j" **type** **is** **boolean** **while** right " **not** g" **type** **is** error.
7. [Operand expression **type** error!] <Line 22> Expression "fun1" **type** should be **boolean** but **not** **integer**.
8. [Assign statement **type** mismatch!] <Left at line 22, right at line 22> Left "j" **type** **is** **boolean** **while** right " **not** fun1" **type** **is** error.
9. [Operand expression **type** error!] <Line 25> Expression "a" **type** should be **boolean** but **not** **integer**.
10. [Assign statement **type** mismatch!] <Left at line 25, right at line 25> Left "j" **type** **is** **boolean** **while** right "a **and** l" **type** **is** error.
11. [Operand expression **type** error!] <Line 26> Expression "g" **type** should be **boolean** but **not** **char**.
12. [Operand expression **type** error!] <Line 26> Expression "h" **type** should be **boolean** but **not** **char**.
13. [Assign statement **type** mismatch!] <Left at line 26, right at line 26> Left "j" **type** **is** **boolean** **while** right "g **and** h" **type** **is** error.
14. [Operand expression **type** error!] <Line 28> Expression "a" **type** should be **boolean** but **not** **integer**.
15. [Assign statement **type** mismatch!] <Left at line 28, right at line 28> Left "j" **type** **is** **boolean** **while** right "a **or** l" **type** **is** error.
16. [Operand expression **type** error!] <Line 29> Expression "g" **type** should be **boolean** but **not** **char**.
17. [Operand expression **type** error!] <Line 29> Expression "h" **type** should be **boolean** but **not** **char**.
18. [Assign statement **type** mismatch!] <Left at line 29, right at line 29> Left "j" **type** **is** **boolean** **while** right "g **or** h" **type** **is** error.

* 分析

and, not, or 要求运算数类型为boolean, 且不支持integer, real到boolean的隐式转换

### 测试3

* 测试用例：+、-、\*、/、minus

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d,e,f:**real**;
4. g,h,i:**char**;
5. j,k,l:**boolean**;
6. **begin**
7. a:=b+c; //正确
8. d:=d+e; //正确
9. a:=b+f; //错误，左值为integer，右值为real
10. d:=e+c; //正确
12. a:=b/c; //正确
13. d:=d/e; //正确
14. a:=b/f; //错误，左值为integer，右值为real
15. d:=e/c; //正确
17. a:=g-h; //错误，g和h均为char
18. d:=j-k; //错误，j和k均为boolean
19. a:=b-g; //错误，g为char
20. d:=c-j; //错误，j为boolean
21. a:=d-g; //错误，a为int，d为real，g为char
22. d:=d-l; //错误，l为boolean
23. a:=g-k; //错误，a为integer，g为char，k为boolean
25. a:=g\*h; //错误，g和h均为char
26. d:=j\*k; //错误，j和k均为boolean
27. a:=b\*g; //错误，g为char
28. d:=c\*j; //错误，j为boolean
29. a:=d\*g; //错误，a为int，d为real，g为char
30. d:=d\*l; //错误，l为boolean
31. a:=g\*k; //错误，a为integer，g为char，k为boolean
33. a:=-b; //正确
34. d:=-b; //正确
35. a:=-d; //错误，a为integer，d为real
36. d:=-d; //正确
37. a:=-g; //错误，a为integer，g为char
38. d:=-g; //错误，d为real，g为char
39. a:=-j; //错误，a为integer，j为boolean
40. d:=-j; //错误，d为real，j为boolean
41. **end**.

* 预期结果

报左边integer与右边real类型不匹配、integer与boolean类型不匹配等左右表达式类型不匹配的错误

* 测试结果

1. [Assign statement **type** mismatch!] <Left at line 9, right at line 9> Left "a" **type** **is** **integer** **while** right "b + f" **type** **is** **real**.
2. [Assign statement **type** mismatch!] <Left at line 14, right at line 14> Left "a" **type** **is** **integer** **while** right "b / f" **type** **is** **real**.
3. [Operand expression **type** error!] <Line 17> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
4. [Operand expression **type** error!] <Line 17> Expression "h" **type** should be **integer** **or** **real** but **not** **char**.
5. [Assign statement **type** mismatch!] <Left at line 17, right at line 17> Left "a" **type** **is** **integer** **while** right "g - h" **type** **is** error.
6. [Operand expression **type** error!] <Line 18> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
7. [Operand expression **type** error!] <Line 18> Expression "k" **type** should be **integer** **or** **real** but **not** **boolean**.
8. [Assign statement **type** mismatch!] <Left at line 18, right at line 18> Left "d" **type** **is** **real** **while** right "j - k" **type** **is** error.
9. [Operand expression **type** error!] <Line 19> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
10. [Assign statement **type** mismatch!] <Left at line 19, right at line 19> Left "a" **type** **is** **integer** **while** right "b - g" **type** **is** error.
11. [Operand expression **type** error!] <Line 20> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
12. [Assign statement **type** mismatch!] <Left at line 20, right at line 20> Left "d" **type** **is** **real** **while** right "c - j" **type** **is** error.
13. [Operand expression **type** error!] <Line 21> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
14. [Assign statement **type** mismatch!] <Left at line 21, right at line 21> Left "a" **type** **is** **integer** **while** right "d - g" **type** **is** error.
15. [Operand expression **type** error!] <Line 22> Expression "l" **type** should be **integer** **or** **real** but **not** **boolean**.
16. [Assign statement **type** mismatch!] <Left at line 22, right at line 22> Left "d" **type** **is** **real** **while** right "d - l" **type** **is** error.
17. [Operand expression **type** error!] <Line 23> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
18. [Operand expression **type** error!] <Line 23> Expression "k" **type** should be **integer** **or** **real** but **not** **boolean**.
19. [Assign statement **type** mismatch!] <Left at line 23, right at line 23> Left "a" **type** **is** **integer** **while** right "g - k" **type** **is** error.
20. [Operand expression **type** error!] <Line 25> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
21. [Operand expression **type** error!] <Line 25> Expression "h" **type** should be **integer** **or** **real** but **not** **char**.
22. [Assign statement **type** mismatch!] <Left at line 25, right at line 25> Left "a" **type** **is** **integer** **while** right "g \* h" **type** **is** error.
23. [Operand expression **type** error!] <Line 26> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
24. [Operand expression **type** error!] <Line 26> Expression "k" **type** should be **integer** **or** **real** but **not** **boolean**.
25. [Assign statement **type** mismatch!] <Left at line 26, right at line 26> Left "d" **type** **is** **real** **while** right "j \* k" **type** **is** error.
26. [Operand expression **type** error!] <Line 27> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
27. [Assign statement **type** mismatch!] <Left at line 27, right at line 27> Left "a" **type** **is** **integer** **while** right "b \* g" **type** **is** error.
28. [Operand expression **type** error!] <Line 28> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
29. [Assign statement **type** mismatch!] <Left at line 28, right at line 28> Left "d" **type** **is** **real** **while** right "c \* j" **type** **is** error.
30. [Operand expression **type** error!] <Line 29> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
31. [Assign statement **type** mismatch!] <Left at line 29, right at line 29> Left "a" **type** **is** **integer** **while** right "d \* g" **type** **is** error.
32. [Operand expression **type** error!] <Line 30> Expression "l" **type** should be **integer** **or** **real** but **not** **boolean**.
33. [Assign statement **type** mismatch!] <Left at line 30, right at line 30> Left "d" **type** **is** **real** **while** right "d \* l" **type** **is** error.
34. [Operand expression **type** error!] <Line 31> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
35. [Operand expression **type** error!] <Line 31> Expression "k" **type** should be **integer** **or** **real** but **not** **boolean**.
36. [Assign statement **type** mismatch!] <Left at line 31, right at line 31> Left "a" **type** **is** **integer** **while** right "g \* k" **type** **is** error.
37. [Assign statement **type** mismatch!] <Left at line 35, right at line 35> Left "a" **type** **is** **integer** **while** right " - d" **type** **is** **real**.
38. [Operand expression **type** error!] <Line 37> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
39. [Assign statement **type** mismatch!] <Left at line 37, right at line 37> Left "a" **type** **is** **integer** **while** right " - g" **type** **is** error.
40. [Operand expression **type** error!] <Line 38> Expression "g" **type** should be **integer** **or** **real** but **not** **char**.
41. [Assign statement **type** mismatch!] <Left at line 38, right at line 38> Left "d" **type** **is** **real** **while** right " - g" **type** **is** error.
42. [Operand expression **type** error!] <Line 39> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
43. [Assign statement **type** mismatch!] <Left at line 39, right at line 39> Left "a" **type** **is** **integer** **while** right " - j" **type** **is** error.
44. [Operand expression **type** error!] <Line 40> Expression "j" **type** should be **integer** **or** **real** but **not** **boolean**.
45. [Assign statement **type** mismatch!] <Left at line 40, right at line 40> Left "d" **type** **is** **real** **while** right " - j" **type** **is** error.

* 分析

+-\*/支持integer到real的隐式转换，不支持real到integer的隐式转换等其他类型之间的隐式转换

### 测试4

* 测试用例：div、mod

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d,e,f:**real**;
4. g,h,i:**char**;
5. j,k,l:**boolean**;
6. **begin**
7. a:= a **div** b; //正确
8. a:= a **div** d; //d为real
9. a:= a **div** g; //g为char
10. a:= a **div** j; //j为boolean
11. a:= d **div** g; //d为real，g为char
12. a:= d **div** j; //d为real，j为boolean
13. a:= g **div** j; //g为char，j为boolean
15. a:= a **mod** b; //正确
16. a:= a **mod** d; //d为real
17. a:= a **mod** g; //g为char
18. a:= a **mod** j; //j为boolean
19. a:= d **mod** g; //d为real，g为char
20. a:= d **mod** j; //d为real，j为boolean
21. a:= g **mod** j; //g为char，j为boolean
22. **end**.

* 预期结果

报告运算数类型不为integer而是real、char、boolean等类型的错误

* 测试结果

1. [Operand expression **type** error!] <Line 8> Expression "d" **type** should be **integer** but **not** **real**.
2. [Assign statement **type** mismatch!] <Left at line 8, right at line 8> Left "a" **type** **is** **integer** **while** right "a **div** d" **type** **is** error.
3. [Operand expression **type** error!] <Line 9> Expression "g" **type** should be **integer** but **not** **char**.
4. [Assign statement **type** mismatch!] <Left at line 9, right at line 9> Left "a" **type** **is** **integer** **while** right "a **div** g" **type** **is** error.
5. [Operand expression **type** error!] <Line 10> Expression "j" **type** should be **integer** but **not** **boolean**.
6. [Assign statement **type** mismatch!] <Left at line 10, right at line 10> Left "a" **type** **is** **integer** **while** right "a **div** j" **type** **is** error.
7. [Operand expression **type** error!] <Line 11> Expression "d" **type** should be **integer** but **not** **real**.
8. [Operand expression **type** error!] <Line 11> Expression "g" **type** should be **integer** but **not** **char**.
9. [Assign statement **type** mismatch!] <Left at line 11, right at line 11> Left "a" **type** **is** **integer** **while** right "d **div** g" **type** **is** error.
10. [Operand expression **type** error!] <Line 12> Expression "d" **type** should be **integer** but **not** **real**.
11. [Operand expression **type** error!] <Line 12> Expression "j" **type** should be **integer** but **not** **boolean**.
12. [Assign statement **type** mismatch!] <Left at line 12, right at line 12> Left "a" **type** **is** **integer** **while** right "d **div** j" **type** **is** error.
13. [Operand expression **type** error!] <Line 13> Expression "g" **type** should be **integer** but **not** **char**.
14. [Operand expression **type** error!] <Line 13> Expression "j" **type** should be **integer** but **not** **boolean**.
15. [Assign statement **type** mismatch!] <Left at line 13, right at line 13> Left "a" **type** **is** **integer** **while** right "g **div** j" **type** **is** error.
16. [Operand expression **type** error!] <Line 16> Expression "d" **type** should be **integer** but **not** **real**.
17. [Assign statement **type** mismatch!] <Left at line 16, right at line 16> Left "a" **type** **is** **integer** **while** right "a **mod** d" **type** **is** error.
18. [Operand expression **type** error!] <Line 17> Expression "g" **type** should be **integer** but **not** **char**.
19. [Assign statement **type** mismatch!] <Left at line 17, right at line 17> Left "a" **type** **is** **integer** **while** right "a **mod** g" **type** **is** error.
20. [Operand expression **type** error!] <Line 18> Expression "j" **type** should be **integer** but **not** **boolean**.
21. [Assign statement **type** mismatch!] <Left at line 18, right at line 18> Left "a" **type** **is** **integer** **while** right "a **mod** j" **type** **is** error.
22. [Operand expression **type** error!] <Line 19> Expression "d" **type** should be **integer** but **not** **real**.
23. [Operand expression **type** error!] <Line 19> Expression "g" **type** should be **integer** but **not** **char**.
24. [Assign statement **type** mismatch!] <Left at line 19, right at line 19> Left "a" **type** **is** **integer** **while** right "d **mod** g" **type** **is** error.
25. [Operand expression **type** error!] <Line 20> Expression "d" **type** should be **integer** but **not** **real**.
26. [Operand expression **type** error!] <Line 20> Expression "j" **type** should be **integer** but **not** **boolean**.
27. [Assign statement **type** mismatch!] <Left at line 20, right at line 20> Left "a" **type** **is** **integer** **while** right "d **mod** j" **type** **is** error.
28. [Operand expression **type** error!] <Line 21> Expression "g" **type** should be **integer** but **not** **char**.
29. [Operand expression **type** error!] <Line 21> Expression "j" **type** should be **integer** but **not** **boolean**.
30. [Assign statement **type** mismatch!] <Left at line 21, right at line 21> Left "a" **type** **is** **integer** **while** right "g **mod** j" **type** **is** error.

* 分析

div、mod要求两个表达式类型均为integer，其他类型不支持

### 测试5

* 测试用例：/、div、mod的除0错误，常数表达式的计算

1. **program** test(input,output);
2. **var** a,b,c:**integer**;
3. d,e,f:**real**;
4. **begin**
5. a:=a / (2+3-5); //除0错误
6. a:=a **div** ( 6 **mod** 3); //除0错误
7. a:=a **mod** ( 1 **div** 3); //除0错误
8. **end**.

* 预期结果

报除0错误

* 测试结果

1. [Divide zero error!] <Line 5> The value **of** expression "(2 + 3 - 5)" **is** 0, which **is** the second operand **of** operation "/".
2. [Divide zero error!] <Line 6> The value **of** expression "(6 **mod** 3)" **is** 0, which **is** the second operand **of** operation "**div**".
3. [Divide zero error!] <Line 7> The value **of** expression "(1 **div** 3)" **is** 0, which **is** the second operand **of** operation "**mod**".

* 分析

/、div、mod不允许除数为0

### 测试6

* 测试用例

1. **program** test(input,ouput);
2. **var** a,b,c:**integer**;
3. d,e,f:**real**;
4. g,h,i:**char**;
5. j,k,l:**boolean**;
6. m:**array**[1..5] **of** **integer**;
8. **procedure** pro;
9. **begin**
11. **end**;
13. **function** fun:**integer**;
14. **begin**
16. **end**;
18. **begin**
19. a:=a+test; //test是主程序名
20. a:=a-input; //input是主程序参数
21. a:=a+pro; //pro是子过程名
22. a:=a-fun; //正确
23. a:=a+m; //m是数组名
24. **end**.

* 预期结果

报主程序名、主程序参数、过程名、数组不能被引用，以及语句左右类型不匹配的错误

* 测试结果

1. [Invalid reference] <Line 19> Procedure name "test" can't be referenced
2. [Assign statement **type** mismatch!] <Left at line 19, right at line 19> Left "a" **type** **is** **integer** **while** right "a + test" **type** **is** error.
3. [Invalid reference!] <Line 20> "input" **is** a parameter **of** **program**, it can't be referenced.
4. [Assign statement **type** mismatch!] <Left at line 20, right at line 20> Left "a" **type** **is** **integer** **while** right "a - input" **type** **is** error.
5. [Invalid reference!] <Line 21> "pro" **is** a **procedure**, it can't be referenced.
6. [Assign statement **type** mismatch!] <Left at line 21, right at line 21> Left "a" **type** **is** **integer** **while** right "a + pro" **type** **is** error.
7. [Invalid reference!] <Line 23> "m" **is** a **array**, it can't be referenced.
8. [Assign statement **type** mismatch!] <Left at line 23, right at line 23> Left "a" **type** **is** **integer** **while** right "a + m" **type** **is** error.

* 分析

测试结果符合预期，语义分析程序继续运行。

## 测试库程序、主程序名、主程序参数相关检查

**库程序有四个,为read,write,writeln,exit，均为过程**

**主程序名不能和库程序同名**

**主程序参数不能和库程序、主程序名同名**

**其它所有标识符不能和这三者同名**

### 测试1

* 测试用例

1. **program** read(input,output); //主程序名与库程序同名
2. **begin**
4. **end**.

* 预期结果

报read已经被定义为库函数的错误

* 测试结果

1. [Duplicate defined error!] <Line 1> Name **of** **program** "read" has been defined **as** a lib **program**.

* 分析

测试结果符合预期，语义分析程序继续运行。

### 测试2

* 测试用例

1. **program** test(**write**,output); //主程序参数与库程序同名
2. **begin**
4. **end**.

* 预期结果

报主程序参数write已被定义为库函数的错误

* 测试结果

1. [Dulicate defined error!] <Line 1> parameter **of** **program** "**write**" has been defined **as** a lib **program**.

* 分析

测试结果符合预期，语义分析程序继续运行。

### 测试3

* 测试用例

1. **program** test(test,output); //主程序参数与主程序名同名
2. **begin**
4. **end**.

* 预期结果

报主程序参数test与主程序名相同的错误

* 测试结果

1. [Duplicate defined error!] <Line 1> parameter **of** **program** "test" **is** the same **as** name **of** **program**.

* 分析

测试结果符合预期，语义分析程序继续运行

### 测试4

* 测试用例

1. **program** test(input,output);
2. **const** output=5; //常量与主程序参数同名
3. **var** read:**integer**; //变量与库程序同名
4. test:**char**; //变量与主程序名同名
5. input:**real**; //变量与主程序参数同名
6. **write**:**array**[1..5] **of** **integer**; //数组名与库程序名同名
8. **function** **writeln**:**integer**; //函数名与库程序同名
9. **begin**
11. **end**;
13. **procedure** exit; //过程名与库程序同名
14. **begin**
16. **end**;
18. **function** test:**integer**; //函数名与主程序名同名
19. **begin**
21. **end**;
23. **procedure** test; //过程名与主程序名同名
24. **begin**
26. **end**;
28. **function** input:**integer**; //函数名与主程序参数同名
29. **begin**
31. **end**;
33. **procedure** output; //过程名与主程序参数同名
34. **begin**
36. **end**;
37. **procedure** pro(**var** test:**integer**;**write**:**real**;input:**char**);//参数与主程序名、库程序名、主程序参数名同名
38. **const** **writeln**=5; //常量与库程序同名
39. **var** output:**integer**; //变量与主程序参数同名
40. **begin**
42. **end**;
44. **begin**
46. **end**.

* 预期结果

报常量、变量、参数等与库函数、主程序名、主程序参数相等的错误

* 测试结果

1. [Duplicate defined error!] <Line 2> "output" has been defined **as** a **program** parameter at Line 1.
2. [Duplicate defined error!] <Line 3> "read" has been defined **as** a lib **program**.
3. [Duplicate defined error!] <Line 4> "test" has been defined **as** the name **of** **program** at Line 1.
4. [Duplicate defined error!] <Line 5> "input" has been defined **as** a **program** parameter at Line 1.
5. [Duplicate defined error!] <Line 6> "**write**" has been defined **as** a lib **program**.
6. [Duplicate defined error!] <Line 8> "**writeln**" has already been defined **as** a lib **program**.
7. [Duplicate defined error!] <Line 13> "exit" has already been defined **as** a lib **program**.
8. [Duplicate defined error!] <Line 18> "test" has already been defined **as** a (sub)**program** name at line 1.
9. [Duplicate defined error!] <Line 23> "test" has already been defined **as** a (sub)**program** name at line 1.
10. [Duplicate defined error!] <Line 28> "input" has already been defined **as** a parameter **of** **program** at line 1.
11. [Duplicate defined error!] <Line 33> "output" has already been defined **as** a parameter **of** **program** at line 1.
12. [Duplicate defined error!] <Line 38> "test" has been defined **as** the name **of** **program** at Line 1.
13. [Duplicate defined error!] <Line 38> "**write**" has been defined **as** a lib **program**.
14. [Duplicate defined error!] <Line 38> "input" has been defined **as** a **program** parameter at Line 1.
15. [Duplicate defined error!] <Line 39> "**writeln**" has been defined **as** a lib **program**.
16. [Duplicate defined error!] <Line 40> "output" has been defined **as** a **program** parameter at Line 1.

* 分析

变量、常量、参数、数组等不能与库函数、主程序和主程序参数重名。测试结果符合预期，语义分析程序继续运行。

## 综合测试

* 测试用例

1. program read(write,output);//主程序名、主程序参数和库程序同名
2. **const** m=10;
3. exit=m;//exit是库函数
4. var a,x,y:integer;
5. b:real;
6. c:**char**;
7. d:boolean;
8. e:array[1..5] of integer;
9. function fun1:integer;//报函数没有返回语句的警告
10. begin
11. v:=a;//v未定义
12. end;
14. function fun2:integer;
15. begin
16. fun2:=1;
17. fun2:=e[6];//数组下标越界
18. fun2;//函数不能作为一条单独的语句
19. a:=fun2[1];//错把函数名当做数组
20. end;
22. procedure pro1(a:integer;b:real);
23. var c:real;
24. d:integer;
25. begin
26. c:=a+b;//integer可以隐式转换为real
27. d:=a+b;//real不能隐式转换为integer
28. end;
30. procedure pro2(var a:real;b:integer);
31. begin
32. exit(a+b);//过程没有返回值
33. end;
35. begin
36. a:=1;
37. m:=a;//常量赋值语句右值不为常量
38. b:=2;
39. c:=3;//赋值语句左右类型不匹配
40. d:=a>b;//d为false
41. **if** a then b:=b+1;//if条件表达式不为boolean
42. repeat a:=a+c until not d;//a:=a+c语句左右类型不匹配
43. **for** b:=10 to 1 **do** e(a,a);//循环变量不可以是real；错把数组名当做函数
44. **while** a<10 **do** a:=a+1;
45. x:=pro1(x,y);//pro1为过程，没有返回值
46. x:=1;
47. y:=2;
48. pro1(x,y);//传值参数支持integer到real的隐式转换
49. pro2(x,y);//pro2的第一个参数为传引用，integer无法隐式转换为real
50. pro1(x+y);//pro1有两个参数
51. pro1(x+y,x+y);//传值参数支持复合表达式
52. pro2(x+y);//pro2有两个参数
53. pro2(x+y,x+y);//pro2第一个参数为引用参数，只能是变量或者数组元素，不能是复杂表达式
54. end.

* 预期结果

1. 第1行报主程序名、主程序参数与库程序名重名的错误；
2. 第3行报exit是库程序，不能被定义为常量的错误；
3. 第9行报fun1子函数没有返回值的警告；
4. 第11行报变量v未定义错误；
5. 第17行报数组下标越界错误；
6. 第18行报fun2为函数而不是过程，不能作为一条单独语句的错误；
7. 第19行报fun2为函数而非数组的错误；
8. 第27行报d为integer而a+b为real，类型不匹配的错误；
9. 第32行报pro2为过程，没有返回值的错误；

10. 第37行报常量m不能作为左值被引用的错误；

11. 第39行报赋值语句左右类型不匹配的错误；

12. 第41行报if条件语句类型不为boolean的错误；

13. 第42行报c的类型不为integer且a:=a+c语句左右类型不匹配的错误；

14. 第43行报for语句循环变量类型错误；

15. 第45行报pro1为过程而非函数，没有返回值的错误；

16. 第49行报x为integer而第一个形参为real，类型不匹配的错误（传引用传输不支持integer到real的隐式转换）；

17. 第50行报pro1应有两个参数而实参只有一个的错误；

18. 第52行报pro2应有两个参数而实参只有一个的错误；

19. 第53行报pro2的第一个参数只能是变量、数组元素而不能是复杂表达式的错误（传引用参数不支持复杂表达式）

* 测试结果

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Here are the semantic warnings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. [Return value statement missing!] <Line 9> Incomplete **return** value statement of function "fun1".
3. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please pay attention to these semantic warnings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
5. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Here are the semantic errors\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
6. [Duplicate defined error!] <Line 1> Name of program "read" has been defined as a lib program.
7. [Dulicate defined error!] <Line 1> parameter of program "write" has been defined as a lib program.
8. [Duplicate defined error!] <Line 3> "exit" has been defined as a lib program.
9. [Undefined identifier!] <Line 11> v has not been defined.
10. [Assign statement type mismatch!] <Left at line 11, right at line 11> Left "v" type is error **while** right "a" type is integer.
11. [Array range out of bound!] <Line 17> The value of expression "6" is 6, but the range of array "e" 0th index is 1 to 5.
12. [Symbol kinds mismatch!] <Line 18> "fun2" defined at line 14 is a function but not a procedure.
13. [Symbol kinds mismatch!] <Line 19> "fun2" defined at line 14 is a (sub)program name but not a array.
14. [Assign statement type mismatch!] <Left at line 19, right at line 19> Left "a" type is integer **while** right "fun2" type is error.
15. [Assign statement type mismatch!] <Left at line 27, right at line 27> Left "d" type is integer **while** right "a + b" type is real.
16. [Return value redundancy!] <Line 32> Number of **return** value of procedure must be 0, that is, exit must have no actual parameters.
17. [Constant as l-value error!] <Line37> Costant "m" can't be referenced as l-value.
18. [Assign statement type mismatch!] <Left at line 39, right at line 39> Left "c" type is **char** **while** right "3" type is integer.
19. [Expression type error!] <Line 41> Expression "a" used **for** condition of **if** statement should be boolean but not integer.
20. [Operand expression type error!] <Line 42> Expression "c" type should be integer or real but not **char**.
21. [Assign statement type mismatch!] <Left at line 42, right at line 42> Left "a" type is integer **while** right "a + c" type is error.
22. [Usage type error!] <Line 43> "b" used **for** cyclic variable of **for** statement should be integer but not real.
23. [Symbol kinds mismatch!] <Line 45> "pro1" defined at line 22 is a procedure but not a function.
24. [Assign statement type mismatch!] <Left at line 45, right at line 45> Left "x" type is integer **while** right "pro1(x, y)" type is error.
25. [Expression type error!] <Line 49> Expression "x" used **for** 1th actual parameter of procedure call of "pro2" should be real but not integer.
26. [Procedure parameter number mismatch!] <Line 50> Procedure "pro1" should have 2 but not 1 parameters.
27. [Procedure parameter number mismatch!] <Line 52> Procedure "pro2" should have 2 but not 1 parameters.
28. [Referenced actual parameter error!] <Line 53> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

* 分析

测试结果符合预期，语义分析程序继续运行。