

编译原理-计算表达式测试样例

陆庭伟-2021300004028

一、词法分析测试用例

正确的输入

这些输入都是正确的标识符、常数、表达式和程序结构的示例。

1. `program test; begin x = 123, y = x * 2 end.`
2. `program simple; begin a = 5, b = a + 4 end.`
3. `program calc; begin result = (1 + 2) * (3 / 4) end.`
4. `program vars; begin var1 = 100, var2 = var1 - 20 end.`
5. `program floatOp; begin floatRes = 99.9 / 3.33 end.`
6. `program powerTest; begin powResult = 2 ^ 10 end.`
7. `program nested; begin complex = (a + b) * (c - d) end.`
8. `program mixedTypes; begin mix = 5 + 6.7 - 2 end.`
9. `program longExpr; begin a = 1, b = 2, c = 3, d = 4, e = 5, f = 6, g = 7, h = 8, i = 9, j = 10 end.`
10. `program quickMaths; begin add = 1 + 1, sub = 2 - 1, mul = 3 * 3, div = 4 / 2, pow = 5 ^ 2 end.`

边界条件

这些测试包括单个符号或非常长的输入。

1. `x = 1;`
2. `program short; begin x = y end.`
3. `program nothing; begin end.`
4. `program longInput; begin a = 1, b = 2, c = 3, d = 4, e = 5, f = 6, g = 7, h = 8, i = 9, j = 10, k = 11, l = 12, m = 13, n = 14, o = 15, p = 16, q = 17, r = 18, s = 19, t = 20 end.`
5. `a = 0;`

6. `program single; begin x = 1 end.`
7. `b = 3;`
8. `c = 3.14159;`
9. `d = 1000;`

错误处理

包括非法字符和空输入。

1. `program error; begin x = @ end.`
2. `???`
3. `x = $100;`

4.空输入

二、语法分析测试用例：

1、基本表达式测试

用例 1：单一变量赋值

1. 输入：`program test1; begin x = 10 end.`
预期输出：无语法错误，正确解析赋值语句。
2. 输入：`program test2; begin y = 100 end.`
预期输出：无语法错误，正确处理整数赋值。
3. 输入：`program test3; begin z = 0 end.`
预期输出：无语法错误，正确处理零值赋值。

用例 2：算术表达式

1. 输入：`program calc1; begin result = 3 + 4 * (2 - 1) end.`
预期输出：无语法错误，正确处理优先级和括号。
2. 输入：`program calc2; begin result = (5 + 2) * 3 end.`
预期输出：无语法错误，正确解析表达式中的括号和乘法。

3. 输入：`program calc3; begin result = 10 / (5 - 2) ^ 2 end.`
预期输出：无语法错误，正确处理除法和指数运算的组合。

2、复合语句测试

用例 3：多重赋值和逗号运算

1. 输入：`program multi1; begin x = 10, y = x + 1, z = y - 2 end.`
预期输出：无语法错误，正确分析多重赋值和表达式计算。
2. 输入：`program multi2; begin a = 5, b = a * 2, c = b / 3 end.`
预期输出：无语法错误，正确处理变量间的依赖和运算符优先级。
3. 输入：`program multi3; begin p = 15, q = p ^ 2, r = q - p end.`
预期输出：无语法错误，正确解析幂运算和算术运算的组合。

3、错误处理测试

用例 4：嵌套块错误

1. 输入：`program nested1; begin x = 1, begin y = 2 end end.`
预期输出：语法错误，项目不处理嵌套的程序块。
2. 输入：`program nested2; begin a = 3, begin b = 4 end, c = 5 end.`
预期输出：语法错误，多个嵌套块未被正确处理。
3. 输入：`program nested3; begin x = 6, begin y = 7 end, z = 8 end.`
预期输出：语法错误，嵌套块中的逗号使用错误。

用例 5：未关闭的括号

1. 输入：`program error1; begin a = (1 + 2 * 3 end.`
预期输出：语法错误提示，报告错误位置。
2. 输入：`program error2; begin b = (4 + 5 * (6 - 2 end.`
预期输出：语法错误提示，报告错误位置。
3. 输入：`program error3; begin c = ((7 + 8) * 9 end.`
预期输出：语法错误提示，报告错误位置。

用例 6：多余的句号

1. 输入：`program dot1; begin x = 10. end.`
预期输出：语法错误提示，报告错误位置。

2. 输入：`program dot2; begin y = 20. end.`
预期输出：语法错误提示，报告错误位置。
3. 输入：`program dot3; begin z = 30 end..`
预期输出：语法错误提示，报告错误位置。

4、关键字和特殊字符的处理

用例 7：关键字作为变量名

1. 输入：`program keyword1; begin end = 10, begin = 5 end.`
预期输出：语法错误提示，
`end` 和 `begin` 不能用作变量名。
2. 输入：`program keyword2; begin program = 15 end.`
预期输出：语法错误提示，
`program` 不能用作变量名。
3. 输入：`program keyword3; begin program = 30, begin = 40 end.`
预期输出：语法错误提示，
`program` 和 `begin` 不能用作变量名。

用例 8：非法字符(非法字符在词法分析阶段即被识别)

1. 输入：`program char1; begin x = 10 & y = 20 end.`
预期输出：词法错误提示，非法字符
`&`。
2. 输入：`program char2; begin x = 30 # y = 40 end.`
预期输出：词法错误提示，非法字符
`#`。
3. 输入：`program char3; begin x = 50 ? y = 60 end.`
预期输出：词法错误提示，非法字符
`?`。

三、语义分析测试用例

测试用例1：正确的变量赋值和运算

- 输入：`program multi; begin x = 10, y = x + 1, z = y - 2 end.`
- 预期输出：无错误，生成的中间代码为：

```

①1 (=, _x, _, _10)
①2 (+, _x, _1, _T1)
①3 (=, _y, _, _T1)
①4 (-, _y, _2, _T2)
①5 (=, _z, _, _T2)

```

计算结果：{'x': 10, 'y': 11, 'z': 9}

测试用例2：使用未声明的变量

- 输入：`program error; begin x = a + 1 end.`
- 预期输出：错误提示“Undefined variable 'a' used in assignment”

测试用例3：复合语句正确的多重赋值和计算

- 输入：`program validOps; begin a = 5, b = a * 2, c = b / 3 end.`
- 预期输出：无错误，生成的中间代码为：

```

①1 (=, _a, _, _5)
①2 (*, _a, _2, _T1)
①3 (=, _b, _, _T1)
①4 (/, _b, _3, _T2)
①5 (=, _c, _, _T2)

```

测试用例4：在表达式中使用多个未声明变量

- 输入：`program multiError; begin x = y + z end.`
- 预期输出：错误提示“Undefined variable 'y' used in assignment”（和类似错误对于 'z'）

测试用例5：变量在声明之前被使用

- 输入：`program preUse; begin x = a, a = 10 end.`
- 预期输出：错误提示“Undefined variable 'a' used in assignment before declaration”

测试用例6：正确处理幂运算表达式

- 输入：`program powerCalc; begin x = 10, y = x ^ 2 end.`

- **预期输出：**无错误，生成的中间代码为：

```
①1 (=, _x, _, _10)
①2 (^, _x, _2, _T1)
①3 (=, _y, _, _T1)
```

计算结果：{'x': 10, 'y': 100}