

LABORATORIO DE COMPILADORES E INTÉRPRETES

Batería de Tests para Análisis Semántico

Listado no exhaustivo de los tipos de errores del analizador semántico (estático) que se presentan en análisis del lenguaje de programación Pascal:

1. Que deben encontrarse:

(a) Colisión de nombres:

- 01A1 - Mismo identificador programa y variable global.
- 01A2 - Mismo identificador programa y procedimiento.
- 01A3 - Mismo identificador programa y función.
- 01B1 - Dos variables globales con el mismo nombre.
- 01B2 - Dos variables locales con el mismo nombre.
- 01C - Mismo identificador variable global y función.
- 01D - Mismo identificador variable global y procedimiento.
- 01E - Mismo identificador variable local y función local.
- 01F - Mismo identificador variable local y parámetro.
- 01G - Mismo identificador parámetro y función local.

(b) Identificadores no definidos:

- 02A1 - Identificador de procedimiento sin definir.
- 02A2 - Uso en entorno local de identificador de variable no definido.
- 02A3 - Uso de identificador de función no definido en expresión aritmética.
- 02A4 - Uso de identificador de función booleana no definido en condición de if.
- 02A5 - Uso de identificador de función booleana no definido en condición de while.
- 02B - Identificador de función sin definir.

(c) Aridad no coincidente (Arity mismatch):

- 03A - Pasaje de parámetro a una función sin parámetros.
- 03A1 - Pasaje de un parámetro booleano a función con dos parámetros booleanos.
- 03A2 - Pasaje de tres parámetros booleanos a función con dos parámetros booleanos.
- 03B1 - Pasaje de dos parámetros numéricos a procedimiento con tres parámetros numéricos.
- 03B2 - Pasaje de cuatro parámetros numéricos a procedimiento con tres parámetros numéricos.

(d) Tipo no coincidente (Type mismatches)

- 03C1 - Pasaje de parámetros booleano y numérico a función con dos parámetros booleanos.
- 03C2 - Pasaje de parámetros numérico y booleano a función con dos parámetros booleanos.
- 03C3 - Pasaje de parámetros numérico y booleano a procedimiento con dos parámetros numéricos.
- 03D1 - Pasaje de parámetros expresión booleana y expresión numérica a función con dos parámetros booleanos.
- 03D2 - Pasaje de parámetros expresión booleana y booleano a función con dos parámetros booleanos.
- 03E1 - Pasaje de parámetros expresión numérica y variable numérica a función con dos parámetros numéricos.
- 03E2 - Pasaje de parámetros expresión numérica y constante literal numérica a función con dos parámetros numéricos.
- 04A1 - Uso de expresión booleana asignada como retorno de función numérica.
- 04A2 - Uso de expresión numérica asignada como retorno de función booleana.
- 04A3 - Uso de operador booleano y variable booleana en expresión numérica asignada como retorno de función numérica.
- 04B - Uso de operador y variables booleanadas y numéricas. (OK)
- 04B1 - Uso de operador booleano con expresión de variables numéricas.
- 04B2 - Uso de operador booleano con expresión de variables numéricas.
- 04B3 - Uso de operador numérico con expresión de variables booleanas.
- 04B4 - Uso de operador numérico con expresión de variables booleanas.
- 04B5 - Uso de operadores numérico y booleano con expresión de variables booleanas.
- 04C1 - Uso de variable numérica como condición de if.
- 04C2 - Uso de función numérica como condición de if.
- 04C3 - Uso de expresión numérica como condición de if.
- 04C4 - Uso de expresión booleana mal formada como condición de if.
- 04C5 - Uso de expresión booleana bien formada como condición de if. (OK)
- 04D1 - Uso de expresión booleana mal formada como condición de while.
- 04D2 - Uso de expresión numérica como condición de while.
- 04D3 - Uso de variable numérica como condición de while.
- 04D4 - Uso de constante literal numérica como condición de while.

(e) Problemas de subrutinas

- 05A1 - Función booleana sin retorno

- 05A2 - Función numérica sin retorno
- 05B1 - Variable de retorno de función booleana usada en expresión
- 05B2 - Variable de retorno de función numérica usada en expresión
- 05C1 - Variable de retorno de función usada en procedimiento

2. Sugerencia de controles adecuados:

- Todos los Símbolos (identificadores) encontrados se incorporan a las Tablas de Símbolos
- Las palabras claves no pueden encontrarse en la Tabla de Símbolos.
- Las funciones y los procedimientos con argumentos se reflejan en la Tabla de Símbolos.
- Es posible deducir en tiempo de compilación los tipos de todas las expresiones.
- Todas las funciones devueven valores del tipo correcto.
- La condición de una sentencia if siempre es una expresión booleana.
- La condición de una sentencia while siempre es una expresión booleana.
- Los operandos de una expresión binaria siempre tienen el mismo tipo (no coerción).
- Los operandos de una expresión binaria siempre tienen el tipo correcto (considerar comparaciones).
- El operando de una expresión unaria siempre tiene el tipo correcto.
- Todos los Símbolos de la Tabla de Símbolos tienen un único tipo asignado.
- Todos los literales numéricos se operan con operadores del tipo numérico o de comparación.
- El lado derecho y el lado izquierdo de una asignación siempre tienen el mismo tipo.
- Las funciones y procedimientos siempre tienen el tipo correcto en los argumentos que son pasados como parámetros.
- Las funciones y procedimientos siempre tienen el número correcto de argumentos que son pasados como parámetros.
- No pueden realizarse llamadas a procedimientos en una expresión.
- No pueden realizarse llamadas a funciones fuera de una expresión.
- Las funciones siempre deben devolver valores, en caso contrario debe señalarse como error.