

Especificación del lenguaje BLAH D:

Introducción

Este documento explica la especificación del lenguaje BLAH. Es un lenguaje simple, de alcance estático, imperativo, sin orientación a objetos con un sistema de tipos estático y similar a C/C++. El propósito del mismo está pensado para desarrollar operaciones de manipulación de strings en general y memeficar los archivos.

Léxico

El léxico del lenguaje está basado en el lenguaje natural. Las palabras reservadas del lenguaje, que no pueden ser usadas por el programador ni redefinidas son:

`integer, boolean, floating, character, string, for, from, to, while, do, if, then, else, return, stop, next, print, read, register, union, array, of, true, false, function, plus, minus, times, dividedby.`

Un identificador es una secuencia de letras, dígitos y underscores, comenzando por una letra. BLAH es sensible a la capitalización, por ejemplo, `foo` y `FOO` son identificadores diferentes. Tamaño máximo de los identificadores sin definir todavía.

Los espacios en blanco (i.e espacios, tabs, etc) sirven para separar tokens pero de resto son ignorados. Palabras reservadas e identificadores deben estar separados por un espacio en blanco, o un token que no sea palabra reservada ni otro identificador. En otras palabras, `arrayofinteger` es un identificador y `array)of)integer` se reconoce como 5 tokens.

Un número entero puede ser especificado en decimal (base 10), octal (base 8), o hexadecimal (base 16). Un entero decimal es una secuencia de dígitos decimales. Un entero en octal se especifica con un 0 (zero) al principio del número y es seguido por una secuencia de dígitos octales. Por ejemplo `01`, `012`, `05369`. Un entero hexadecimal debe empezar con `0x` o con `0X` y seguido de una secuencia de dígitos hexadecimales. Los dígitos hexadecimales incluyen los dígitos decimales y las letras del abecedario de la `a` a la `f`, en mayúsculas o minúsculas por igual. Para dejar constancia, los siguientes son enteros válidos:

`8, 16, 01234, 0x5ad89, 0X4Eb23, 09842`

Un número flotante es una secuencia de dígitos decimales, un punto, y continúa con otra secuencia de dígitos. Por tanto, ni `.12` ni `12.` son válidos, pero `12.0`

y 0.12 si lo son. Un número decimal puede tener un exponente opcional, que se denota con una **e** (minúscula o mayúscula) al final del número seguido del exponente, por ejemplo, 55.7E+2, 0.34e-8. Como se explicó más arriba 34.E+2 y .12e-15 son inválidos. Si no se especifica el signo del exponente el positivo (+) es asumido.

Una secuencia de caracteres encerrada entre comillas simples constituyen un string literal, es inválido escribir comilla simple en un string de este tipo. Una secuencia de caracteres entre comillas dobles requieren escapar signos especiales para tomarlos como parte de la cadena. Un string debe empezar y terminar en la misma línea no puede ser picado en varias líneas. Para ilustrar lo anterior:

```
'Aunque este string está escrito para que se vea deliberadamente
largo y ocupe más de una línea, debe empezar y terminar en la
misma línea para que pueda ser válido en el lenguaje BLAH'
```

```
'Este un string literal donde \n no es un salto de línea'
```

```
"En los strings con doble comilla \n es un salto de línea"
```

```
"Este es un string que requiere de escapar la doble comilla (\")
para que pueda ser considerada parte de la cadena"
```

```
'Este string es inválido pues tiene una comilla simple "'
extra en el medio'
```

Signos de puntuación usados por el lenguaje incluyen:

```
+ - * / % < <= > >= = == != && || ! , . [ ] ( ) { } " ' ,
```

Un comentario de una línea comienza con `//` y se extiende hasta el final de la línea. Como en C/C++, un comentario de bloque empieza por `/*` y termina con `*/`. Cualquier símbolo es permitido en el comentario excepto `*/` que termina el comentario actual. Así como en C/C++ los comentarios no se anidan.

Gramática

Estructura de los programas

Alcance

Tipos

Variables

Arreglos

Cadenas de Caracteres

Funciones

Invocacion de funciones

Registros y Uniones

Equivalencia de tipos y compatibilidad

Asignación

Estructuras de control

Expresiones

Funciones de libreria (I/O)

Programación Modular

Chequeos a tiempo de ejecución

Lo que no tiene el lenguaje