

## Proyecto #1 - Analizador léxico

### Revisor y corrector de archivos en PHP

Fecha de entrega: 6 de abril 2018, 06:55 am

#### Objetivo

En el primer proyecto de programación, usted realizará la aplicación del análisis léxico. La intención del proyecto consiste en utilizar los autómatas finitos determinísticos y las expresiones regulares como reconocedores de tokens para un lenguaje. Para ello, va a utilizar Flex (lex o cualquier otro generador de analizadores de léxico) para crear un escáner para un lenguaje de programación. El escáner se ejecutará a través del programa de origen, reconociendo los tokens en el orden en que se leen, hasta el final del archivo. Esta es una tarea bastante sencilla; la mayoría del trabajo se presenta en averiguar cómo funciona el flex.

#### Descripción

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje general ampliamente usado y Open Source de scripting que es especialmente adecuado para el desarrollo web y puede ser embebido en HTML. Su sintaxis es similar a C, Java y Perl y es fácil de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escriban páginas generadas dinámicamente con rapidez, pero puede hacer mucho más con PHP. El proyecto consiste en:

1. Crear un archivo de entrada para flex (lex o cualquier otro generador de analizadores de léxico) que contenga la definición lexicográfica del lenguaje PHP:
  - a. Palabras reservadas
  - b. Operadores
    - i. Aritméticos
    - ii. Lógicos
  - c. Tipos
    - i. lógicos
    - ii. enteros
    - iii. reales
    - iv. cadena
  - d. Identificadores
  - e. Variables
  - f. Constantes
  - g. Estructuras de control
    - i. if-then-else
    - ii. While
    - iii. Do-while
    - iv. For
    - v. Foreach
    - vi. Break

- vii. Switch
  - viii. Include
  - ix. continue
  - x. return
  - h. Variables predeterminadas
  - i. Funciones
  - j. Comentarios (una línea y multilínea)
  - k. Campos de acceso a base de datos (los campos de acceso a la base de datos de Oracle es: \$recordset['NOMBRE\_DEL\_CAMPO']. Como pueden ver el nombre del campo es en mayúsculas. Si el archivo de entrada viene en minúsculas, la salida debe mostrarlo en forma correcta (mayúsculas).
2. Flex leerá la estructura del archivo anterior y generará un programa en C, C++ o lo que genere según la herramienta que utilicen. Este programa **debe ser compilado en vivo**.
  3. El programa compilado lee un archivo de entrada en lenguaje PHP y determina si pertenece o no al lenguaje.
  4. La salida del programa compilado debe ser el mismo archivo de entrada si éste no contiene errores, pero en formato correcto de PHP (Colocando las estructuras de control en minúsculas, los campos de acceso a BD en mayúsculas, etc.)
  5. El escáner debe reportar los errores. Los errores deben ser presentados claramente en pantalla y llevarlos también en un archivo de salida.

### Sugerencias

- Cuidado con los espacios entre los tokens. El espacio en blanco (es decir, espacios, tabuladores y saltos de línea) sirve para separar tokens, pero por lo demás debe ser ignorado. Además, es muy fácil poner espacios por error y que dichos espacios interpreten de manera diferente un patrón.
- No coloque nuevas líneas en los patrones.
- Cuando tenga duda, coloque paréntesis entre el patrón para asegurar que está obteniendo la precedencia que necesita
- Coloque cada acción entre llaves (si bien es cierto no se requiere una línea simple, es mejor colocarla en una)
- Recomendando agregar los tipos de tokens uno a la vez y probar cada vez que se hace.
  - Cuidado con los caracteres que tienen un significado especial para lex, tales como \* y – (vea la documentación para saber cómo y cuándo quitar dichas condiciones especiales).
  - Los patrones para enteros, dobles (reales) y strings requerirán cuidado especial para asegurarse que todos los casos son cubiertos.

### Requerimientos generales

- El ejecutable debe ser llamado **miniPHP** y debe pedir el archivo de entrada y escribir todos los resultados en un archivo de salida del mismo nombre al de entrada con extensión **.out**
- El proyecto se entregará en un repositorio en GitHub; se calificará el proyecto hasta el último commit realizado antes de la hora de entrega. Se revisará el historial de versiones y commits en el repositorio. **No se aceptaran cambios en el momento de la calificación.**
- En el repositorio deberá incluir:
  - Un archivo MyRegEx.txt listando todas las expresiones regulares que implemente en su analizador.
  - Un archivo README.txt donde explique todo el funcionamiento y porque cree que su programa funciona correctamente y es robusto, así también detallando cómo se manejan los errores.

**¡Buena suerte!**

### Referencias:

- <http://php.net/manual/en/langref.php>
- <http://dinosaur.compilertools.net/#overview>