Algoritmos y Estructuras de Datos
Enunciado Primer Parcial Año 2020

Abril 2020

Realizar un programa que evalúe un archivo de formato HTML. El Hyper Text Markup Language (HTML) es un tipo de archivo de texto que contiene tags que indican las propiedades del texto que encierra . Los tags son de la forma <xx> </xx>, siempre encerrados entre signos <> . El tipo de tag esta dado por el valor que encierren los signos y la mayoria de los tipos de tags estan definidos para trabajar de a pares con el formato <tagx> </tagx> que determina una área de validez del tag <inicio> </final>

Debe realizar el trabajo en tres partes: la primera (*tokenize*) que tome como entrada el archivo con extension html, lea su contenido y obtenga en una lista de tokens. Un token es un tag del tipo de los descriptos antes o el texto que encierra entre tags. Utilice un archivo que no tenga errores en cuanto a la estructura de tags que contiene. Ayudese con una estructura Pila y vaya leyendo caracter a caracter el contenido del archivo. Ejemplo, el siguiente texto:

```
<h1>Este es el titulo</h1>
```

se descompone en tres tokens:

```
<h1>
Este es el titulo
</h1>
```

La segunda parte del trabajo (*parsing*), consiste en tomar como entrada la lista de tokens de la parte anterior y obtener una lista de objetos que representen cada tipo de tag y el texto que encierran. Para ello deberá utilizar una estructura Pila para poder determinar el tipo de tag y su alcance. Por alcance nos referimos a sus limites y valores contenidos. Tenga presente que un tag puede contener dentro de su área no solo texto plano, sino que tambien otros tags, como en el siguiente caso que representa una tabla con lineas y celdas:

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
```

```
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

En el ejemplo anterior, que no está separado en tokens y puede verlo en:
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_headings surge que el tag **table** contiene 4 tokens del tipo **tr**, y a su vez, cada uno de estos contiene 3 tokens del tipo **td** con distinto texto cada uno.

Para esta segunda clase debera utilizar un diseño de clases en C++ de la siguiente forma:

- Una clase abstracta tokenhtml que contenga entre otras cosas una lista donde se van a incluir los objetos tags anidados y un método **show**.
- Tantas subclases que hereden de la anterior como tipos de tags vaya a parsear. Para el caso del ejemplo de la **table**, habrá una clase **table**, otra **tr** y otra **td**, todas heredadas de la clase abstracta

La tercera parte consiste en implementar el método **show** en cada subclase y dada la lista de objetos del tipo tokenhtml obtenida en la segunda parte, presentar en pantalla uno a uno cada tokenhtml de la lista, mostrando el tipo de token que es y su contenido, preferentemente respetando el anidamiento que tenga cada objeto token. Dado el ejemplo de la **table**, la salida de pantalla sería:

```
> table
    tr
        td Firstname
        td Lastname
        td Age
    tr
        td Jill
        td Smith
        td 50
        ...
```

Los clases que deberá utilizar para resolver el problema son:

- Pila: que contiene tokens y es utilizada como auxiliar para el procesamiento
- Tokenhtml: que identifica el tipo de token y su valor, incluyendo los anidados
- Lista de tokenhtml: que contiene los valores a mostrar

Deberá codificar la solución usando Clases de C++, con métodos que respeten el comportamiento expuesto.

Puede desarrollar cada parte del trabajo independiente de las otras desacoplando los resultados de cada una y tomando como entrada valores simulados. En este caso deberá integrar las partes una vez concluidas, manteniendo un formato de resultado / entrada de cada parte consistente.