Práctica 1:
Acceso a datos de la Web

Introducción y Objetivos

El objetivo de la práctica es el acceso a datos procedentes de la Web, como ejemplo de Sistema de Información Distribuido, teniendo en cuenta sus particularidades (carácter abierto, dinamismo, etc.). En particular, consideraremos el acceso a datos en formato HTML ubicados en una o varias páginas web. Como ejemplo concreto, se propone como dominio de aplicación los datos de la bolsa, sin perjuicio de que los alumnos puedan proponer aplicaciones alternativas de su interés personal.

Los datos de la bolsa cambian con mucha frecuencia. No existe ninguna opción gratuita para obtener datos en tiempo real, pero hay diversas páginas que ofrecen información relativamente actualizada. Dado que las diferentes páginas tienen en la práctica diferentes tasas de actualización, si accedemos a varias de ellas tendremos una mejor tasa de refresco de la que cualquiera de ellas tiene de modo individual. Además, como la estructura de las páginas web puede cambiar próximamente (por ejemplo, para mejorar su diseño), se desarrollarán analizadores léxicos usando la herramienta JavaCC (http://javacc.java.net, plug-in para Eclipse accesible a través de http://homepages.gac.edu/~hvidsten/courses/MC270/Labs/project4-GacApplication/project-files/JavaCC/JavaCC-Eclipse.html), de modo que en caso de actualización solamente haya que modificar el fichero `.jj`.

Ejercicios

1. Escribir un programa Java que solicite al usuario el nombre de una empresa del IBEX 35 y le muestre su cotización actual y la diferencia con respecto a la anterior ejecución del programa. Los datos de las cotizaciones se obtendrán de http://www.infobolsa.es/acciones/ibex35, usando JavaCC.

Utilizar XML como formato del fichero auxiliar, utilizando SAX para realizar su lectura.

2. **Optativo.** Extender el ejercicio anterior para considerar también los datos de una segunda página web, http://www.iberbolsa.com, para la cual habrá que construir un nuevo analizador léxico. Si las dos páginas ofrecen cotizaciones diferentes y una de las cotizaciones es igual a la obtenida en la anterior ejecución del programa, supondremos que la otra es el valor más actualizado. Si la cotización de la anterior ejecución del programa es diferente, se considerará la fuente más fiable.

Entrega

La entrega se realizará a través de Moodle hasta las 12 horas del **8 de marzo** y consistirá en un fichero `.zip` que incluya al menos

- los analizadores léxicos `.jj`,
- las clases `.java` desarrolladas por el alumno y las generadas por JavaCC,
- un ejecutable `.jar` y
- una brevísima documentación en `pdf` con los nombres de los autores y los aspectos no triviales del trabajo realizado, incluyendo la instalación de las herramientas necesarias.

Apéndice. Plantilla para el fichero .jj

```
options
{
   static = false;
}

PARSER_BEGIN(Parser)

import java.io.*;
import java.net.*;
import java.util.*;
import javax.xml.parsers.*;

public class Parser implements ParserConstants
{

  // URL donde obtener empresas y cotizaciones "actuales"
  private final static String URL = "http://www.infobolsa.es/acciones/ibex35";

  // Fichero donde obtener empresas y cotizaciones "obsoletas"
  private final static String FICHERO = "cotizacion.xml";


  Hashtable <String, Double> leeTablaDeFichero(String fichero)
  {
    try
    {
      SAXParser saxParser = ...;
      CotizacionesHandler handler = new CotizacionesHandler();
      saxParser.parse(...);
      return handler.getTabla();
    }
    catch (Exception e)
    {
      return new Hashtable<String, Double> ();
    }
  }


  void escribeTablaEnFichero(Hashtable <String, Double> tabla, String nombre)
  {
    PrintStream out = null;
    try
    {
      ...
    }
    catch (Exception e)
    {
      e.printStackTrace();
    }
  }
```

```
  public static void main(String args [])
  {
    try
    {
      // Entrada de datos de la web
      InputStreamReader stream = ...
      Parser parser = ...
      Hashtable<String, Double> tablaCotizaciones = parser.leeTablaCotizaciones();

      // Entrada de datos de teclado
      System.out.println("Introduzca el nombre de una empresa del IBEX 35: ");
      ...
      // Entrada de datos de fichero
      Hashtable<String, Double> tablaAnterior = parser.leeTablaDeFichero(FICHERO);

      // Diferencia de precios
      ...

      // Salvamos los resultados en fichero
      parser.escribeTablaEnFichero(tablaCotizaciones, FICHERO);
    }
    catch (Exception e)
    {
      System.out.println("Exception " + e.getMessage());
    }
    catch (Error e)
    {
      System.out.println("Error " + e.getMessage());
    }
  }
}
PARSER_END(Parser)

SKIP :
{
  " "
| "\r"
| "\t"
| "\n"
}

TOKEN :
{
  < CABECERA : "<!DOCTYPE html>" >
| < HTML : "<html>" >
| < HTML_FIN : "</html>" >
| < HEAD : "<head>" >
| < HEAD_FIN : "</head>" >
| < BODY : "<body class=\"ifb-menu-push\">" >
| < BODY_FIN : "</body>" >
| < NOMBRE_EMPRESA : ... >
| < COTIZACION_EMPRESA : ... >
| < A_FIN : "</a>" >
| < ETIQUETA : "<" >
| < ETIQUETA_FIN : ">" >
| < BARRA : "/" >
| < NUMERO: (["0"-"9"])* ( "," )? (["0"-"9"])+ >
| < CARACTERES : ([ "A"-"Z", "a"-"z", "0"-"9", "Á", "É", "Í", "Ó", "Ú", "Ü", "Ñ",
"á", "é", "í", "ó", "ú", "ü", "ñ", "¡", "!", "€", ".", "¿", "?", ":", ";", ",",
"'", "=", "\"", "-", "_", "+", "°", "*", "(", ")", "\\", "@", "%", "#", "&", "[",
"]", "|", "{", "}", "$" ])+ >
}
```

```
Hashtable<String, Double> leeTablaCotizaciones() :
{
  Hashtable<String, Double> tabla = null;
}
{
  <CABECERA> <HTML> <HEAD> saltar() <HEAD_FIN> tabla = body() <HTML_FIN>
  {
    return tabla;
  }
}


void saltar() :
{}
{
  ( <CARACTERES> | <NUMERO> | <ETIQUETA> | <ETIQUETA_FIN> | <BARRA> | <A_FIN> )*
}


Hashtable<String, Double> body() :
{
  Hashtable<String, Double> tabla = null;
}
{
  <BODY> saltar()  tabla = cotizaciones()  <BODY_FIN>
  {
    return tabla;
  }
}



Hashtable<String, Double> cotizaciones() :
{
  Hashtable<String, Double> tabla = new Hashtable<String, Double> ();
}
{
  <NOMBRE_EMPRESA> ...
  {
    return tabla;
  }
}
```

Apéndice. Plantilla para el manejador de SAX

```
import java.util.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;


public class CotizacionesHandler extends DefaultHandler {

    private Hashtable<String,Double> tabla = new Hashtable<String,Double>();

    ...


    @Override
    public void startDocument() throws SAXException
    {

    }


    @Override
    public void endDocument() throws SAXException
    {

    }


    @Override
    public void startElement(String uri, String localName,String qName,
                             Attributes attributes) throws SAXException
    {
            ...
    }


    @Override
    public void endElement(String uri, String localName, String qName)
                             throws SAXException
    {
            ...
    }


    @Override
    public void characters(char ch[], int start, int length) throws SAXException
    {
            ...
    }


    public Hashtable<String,Double> getTabla() {
            return tabla;
    }

}
```