**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.io.StringWriter;

**import** java.util.ArrayList;

**import** java.util.Date;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.ParserConfigurationException;

**import** javax.xml.transform.OutputKeys;

**import** javax.xml.transform.Transformer;

**import** javax.xml.transform.TransformerFactory;

**import** javax.xml.transform.dom.DOMSource;

**import** javax.xml.transform.stream.StreamResult;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**import** org.w3c.dom.Text;

**import** org.xml.sax.SAXException;

**public** **class** PedidosXML\_DOM {

**private** Document dom;

**public** PedidosXML\_DOM() {

dom = **null**;

}

**private** **void** addCliente(Usuario cliente, Element nodoCliente) {

Element nombre = dom.createElement("Nombre");

Text textoNombre =

dom.createTextNode(cliente.getNombre());

nombre.appendChild(textoNombre);

Element apellido = dom.createElement("Apellido");

Text textoApellido =

dom.createTextNode(cliente.getApellido());

apellido.appendChild(textoApellido);

Element user = dom.createElement("User");

Text textoUser = dom.createTextNode(cliente.getUser());

user.appendChild(textoUser);

nodoCliente.appendChild(nombre);

nodoCliente.appendChild(apellido);

nodoCliente.appendChild(user);

}

**private** **void** addProducto(Producto producto, Element nodoProducto) {

Element nombre = dom.createElement("Nombre");

Text textoNombre =

dom.createTextNode(producto.getNombre());

nombre.appendChild(textoNombre);

Element fechaAlta = dom.createElement("FechaAlta");

Text textoFechaAlta =

dom.createTextNode(String.

*valueOf*(producto.getFechaAlta().getTime()));

fechaAlta.appendChild(textoFechaAlta);

nodoProducto.appendChild(nombre);

nodoProducto.appendChild(fechaAlta);

}

**private** **void** addProductos(ArrayList<Producto> listaProductos,

Element nodoListaProductos) {

**for**(**int** i = 0; i < listaProductos.size(); i++) {

Element producto = dom.createElement("Producto");

addProducto(listaProductos.get(i), producto);

nodoListaProductos.appendChild(producto);

}

}

**private** **void** addPedido(Pedido pedido) {

// seleccionamos la raiz

Element root = dom.getDocumentElement();

// Se crea un nuevo elemento con el atributo del

// número de producto

Element unPedido = dom.createElement("Pedido");

unPedido.setAttribute("numeroPedido",

String.valueOf(pedido.getNumeroPedido()));

// Se crea un nuevo elemento para el cliente

Element cliente = dom.createElement("Cliente");

addCliente(pedido.getCliente(), cliente);

// Se crea un nuevo elemento para los productos

// de los que consta el pedido

Element productos = dom.createElement("Productos");

addProductos(pedido.getListaProductos(), productos);

// Se inserta el cliente y los productos

// en el elemento del pedido

unPedido.appendChild(cliente);

unPedido.appendChild(productos);

// Se inserta el pedido en la raíz

root.appendChild(unPedido);

/\*

\* Escribe en el fichero la representación del árbol XML

\*/

**private** **void** write(StringWriter sw, String path) {

**try** {

// Se crea un fichero para escribir en

//modo texto

PrintWriter writer = **new** PrintWriter(

**new** FileWriter(path));

// Se escribe todo el árbol en XML

writer.println(sw.toString());

// Se cierra el fichero

writer.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

/\*Transforma el árbol, agregando la cabecera

y añadiendo sangrados\*/

**private** **void** toFile(String ruta) {

**try** {

// Se vuelca el XML al fichero

TransformerFactory transFact =

TransformerFactory.*newInstance*();

// Se añade el sangrado

transFact.setAttribute("indent-number",

**new** Integer(3));

Transformer trans =

transFact.newTransformer();

/\* Se incluye la cabecera XML

y el sangrado\*/

trans.setOutputProperty(

OutputKeys.***OMIT\_XML\_DECLARATION***, "no");

trans.setOutputProperty(OutputKeys.***INDENT***,

"yes");

// Se hace la transformación

StringWriter sw = **new** StringWriter();

StreamResult sr = **new** StreamResult(sw);

DOMSource Transformar = **new** DOMSource(dom);

trans.transform(domSource, sr);

// Se escribe en el fichero

write(sw, ruta);

} **catch**(Exception ex) {

ex.printStackTrace();

}

}

/\*

\* Se crea un fichero XML a partir de una lista de pedidos

\*/

**public** **void** pedidosToXML(ArrayList<Pedido> pedidos,

String ruta) {

// Se crea un nuevo Document donde se va a guardar

// toda la estructura

**try** {

dom =

DocumentBuilderFactory.*newInstance*().

newDocumentBuilder().newDocument();

Element root =

dom.createElement("Pedidos");

dom.appendChild(root);

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

}

**for**(**int** i = 0; i < pedidos.size(); i++)

addPedido(pedidos.get(i));

// Se vuelca por pantalla

toFile(ruta);

}

// Lectura de XML con DOM

**private** Usuario readUsuario(Node nodoUsuario) {

Element elementoUsuario = (Element)nodoUsuario;

String nombre = elementoUsuario.

getElementsByTagName("Nombre").item(0).

getTextContent();

String apellido = elementoUsuario.

getElementsByTagName("Apellido").item(0).

getTextContent();

String user = elementoUsuario.

getElementsByTagName("User").item(0).

getTextContent();

**return** **new** Usuario(nombre, apellido, user);

}

**private** Producto readProducto(Node nodoProducto) {

Element elementoProducto = (Element)nodoProducto;

String nombre = elementoProducto.

getElementsByTagName("Nombre").item(0).

getTextContent();

**long** fechaAlta = Long.*valueOf*(elementoProducto.

getElementsByTagName("FechaAlta").item(0).

getTextContent());

**return** **new** Producto(nombre, **new** Date(fechaAlta));

}

**private** Pedido readPedido(Node nodoPedido) {

Element elementoPedido = (Element)nodoPedido;

**int** numeroPedido = Integer.*valueOf*(

elementoPedido.getAttribute("numeroPedido"));

Usuario user = readUsuario(elementoPedido.

getElementsByTagName("Cliente").item(0));

Pedido Integer. ValueOf = **new** Pedido(numeroPedido, user);

NodeList productos = elementoPedido.

getElementsByTagName("Productos");

**for**(**int** i = 0; i < productos.getLength(); i++) {

pedido.addProducto(readProducto

((Element)productos.item(0)));

}

**return** pedido;

}

/\*

\* Se crea una lista de pedidos procesando un fichero XML

\*/

**public** ArrayList<Pedido> XMLtoPedidos(String ruta) {

ArrayList<Pedido> pedidos = **new** ArrayList<Pedido>();

**try** {

DocumentBuilderFactory dbf =

DocumentBuilderFactory.*newInstance*();

DocumentBuilder db = dbf.newDocumentBuilder();

dom = db.parse(**new** File(ruta));

dom.getDocumentElement().normalize();

// Se seleccionan todos los pedidos y se van leyendo

NodeList listaPedidos = dom.getDocumentElement().

getElementsByTagName("Pedido");

**for**(**int** i = 0; i < listaPedidos.getLength(); i++) {

pedidos.add(readPedido(listaPedidos.item(i)));

}

} **catch** (SAXException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

}

**return** pedidos;

}

}