

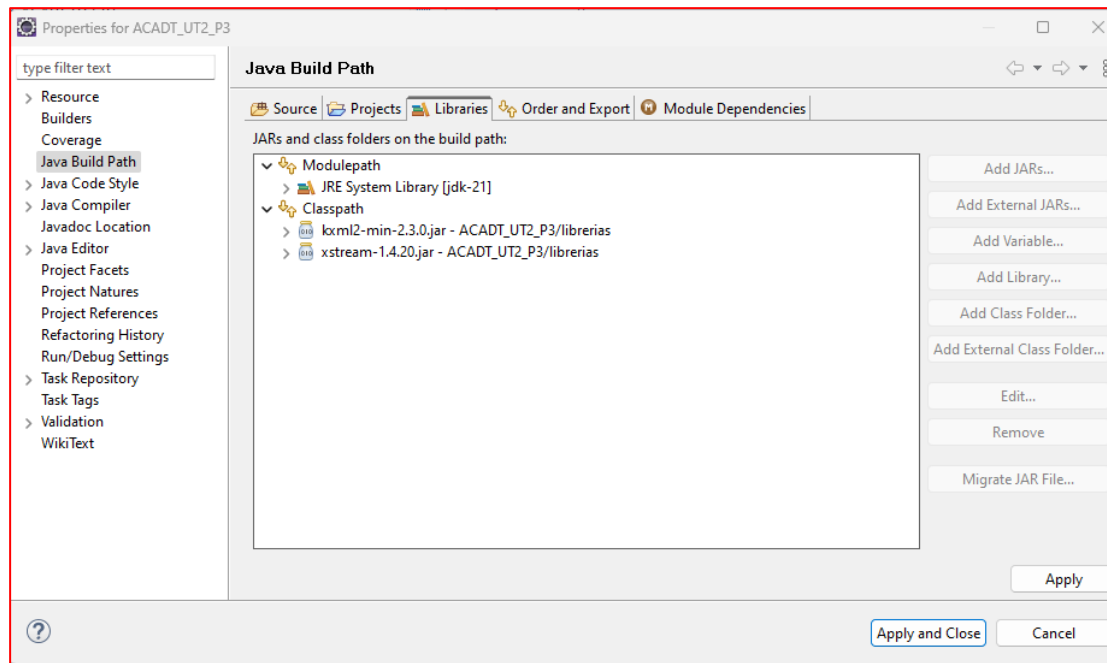
Acceso a Datos

UT2. MANEJO DE FICHEROS XML Y JSON
FICHEROS XML - OBJETOS

XStream - Configuración

- Vamos a ver cómo se pueden serializar de forma sencilla objetos Java a XML y viceversa.
- Utilizaremos para ello la librería XStream. Para poder utilizarla hemos de descargarnos los JAR desde el sitio web: <http://x-stream.github.io/download.html> para el ejemplo se ha descargado el fichero Binary distribution (xstream-distribution-1.4.20-bin.zip) que hemos de descomprimir y buscar el JAR xstream-1.4.20, que está en la carpeta lib, que es el que usaremos para el ejemplo.
- También necesitamos el fichero JAR kxml2-min-2.3.0 (es un analizador sintáctico de XML ligero basado en Java, es de tipo pull) que se encuentra también en la carpeta lib→xstream.
- Una vez que tenemos los dos ficheros, los definimos en el CLASSPATH de nuestro proyecto: Project→Properties→Java Build Path. En la pestaña Libraries, en la opción de Classpath pulsamos Add JARS.

XStream - Configuración



XStream – Objeto a XML

- Partimos del fichero contactos.dat que utilizamos en ejercicios anteriores y que contiene objetos Contacto.
- Crearemos una lista de objetos Contacto y la convertiremos en un fichero de datos XML.
- Necesitaremos la clase Contacto (ya definida) y la clase ListaContactos en la que se define una lista de objetos Contacto y los métodos de addContacto para añadir y un contacto y el método getListaContactos (que devuelve una lista de contactos) y que pasaremos al fichero XML:

XStream – Objeto a XML

```
import java.util.ArrayList;
import java.util.List;

public class ListaContactos {
    private List<Contacto> lista = new ArrayList<Contacto>();

    public ListaContactos(){ }

    public void add(Contacto con) {
        lista.add(con);
    }

    public List<Contacto> getListaContactos() {
        return lista;
    }
}
```

XStream – Objeto a XML

- El proceso consistirá en recorrer el fichero contactos.dat para crear una lista de contactos para después crearnos el fichero Contactos.xml.
- El proceso de ir leyendo objetos Contacto e insertarlo en una Lista no hace falta que lo expliquemos en detalle porque ya sabemos hacerlo. Empezaremos viendo lo relacionado con ficheros XML.

XStream – Objeto a XML

- Las librerías para importar serían:

```
import com.thoughtworks.xstream.XStream;  
import com.thoughtworks.xstream.io.xml.DomDriver;
```

- En primer lugar para utilizar XStream, simplemente creamos una instancia de la clase XStream. A la hora de instanciarlo es necesario indicarle el tipo de parser que va utilizar DOM, Stax,... si no ponemos nada utilizará el parser KML:

```
XStream xstream = new XStream();
```

- Si queremos utilizar el parser de DOM e indicarle el tipo de codificación instanciaremos Xstream de la siguiente forma:

```
XStream xstream = new XStream(new DomDriver("UTF-8"));
```

XStream – Objeto a XML

- En general las etiquetas XML se corresponden con el nombre de los atributos de la clase, pero se pueden cambiar usando el método `alias()`.
- En nuestro ejemplo hemos dado un alias a la clase `Contacto`, en el XML aparecerá con el nombre `DatosContacto`:

```
xstream.alias("DatosContacto", Contacto.class);
```

- Para que **no aparezca el atributo *lista*** de la clase `ListaContactos` en el XML generado se ha utilizado el método `addImplicitCollection()`:

```
xstream.addImplicitCollection(ListaContactos.class, "lista");
```


XStream – Objeto a XML

- Si queremos que no aparezca algún campo de la clase utilizaremos el siguiente el método omitField():

```
xstream.omitField(Contacto.class, "telefono");
```

- Por último, para generar el fichero Contactos.xml a partir de la lista de objetos se utiliza el método toXML(objeto, OutputStream):

```
xstream.toXML(listaContactos, new FileOutputStream("Contactos.xml"));
```

XStream – Objeto a XML

- Ejemplo completo (I)

```
public static void CrearFicheroXML() {  
  
    File fichero = new File("contactos.dat");  
    ListaContactos listaContactos = null;  
  
    try {  
        FileInputStream filein = new FileInputStream(fichero); // flujo de entrada  
        // Sonecta el flujo de bytes al flujo de datos  
        ObjectInputStream dataIS = new ObjectInputStream(filein);  
        System.out.println("Comienza el proceso de creación del fichero a XML ...");  
  
        // Creamos un objeto Lista de Contactos  
        listaContactos = new ListaContactos();  
        try {  
            while (true) { // lectura del fichero  
                Contacto contacto = (Contacto) dataIS.readObject(); // leer un contacto  
                listaContactos.add(contacto); // añadir contacto a la lista  
            }  
        } catch (EOFException eoe) {  
            System.out.println("Fin del fichero");  
        } catch (ClassNotFoundException cnfe) {  
            cnfe.printStackTrace();  
        }  
        finally {  
            dataIS.close(); // cerrar stream de entrada  
        }  
    }  
}
```

XStream – Objeto a XML

- Ejemplo completo (II)

```
//XStream xstream = new XStream();
XStream xstream = new XStream(new DomDriver("UTF-8"));

// Cambiar de nombre a las etiquetas XML
xstream.alias("DatosContacto", Contacto.class);
xstream.alias("Contactos", ListaContactos.class);

// Quitar etiqueta lista (atributo de la clase ListaContactos)
xstream.addImplicitCollection(ListaContactos.class, "lista");

xstream.omitField(Contacto.class, "telefono");

xstream.toXML(listaContactos, new FileOutputStream("Contactos.xml"));

System.out.println("Creado fichero XML....");

} catch (FileNotFoundException fnfe) {
    fnfe.printStackTrace();
} catch (IOException ioe) {
    ioe.printStackTrace();
}
```

XStream – XML a Objeto (deserialización)

- Se deben utilizar los métodos `alias()` y `addImplicitCollection()` para leer el XML ya que se usaron para hacer la escritura del mismo. Para obtener el objeto con la lista de personas o lo que es lo mismo, para deserializar el objeto, utilizamos el método `fromXML(InputStream)`:

```
contactos = (ListaContactos) xstream.fromXML(new FileInputStream("Contactos.xml"));
```

- A continuación se muestra el ejemplo completo

XStream – XML a Objeto (deserialización)

```
public static void LeerXML() {
    XStream xstream = new XStream(new DomDriver("UTF-8"));

    // Vamos disminuir la seguridad de la deserialización de Xstream.
    xstream.addPermission(AnyTypePermission.ANY);
    xstream.alias("DatosContacto", Contacto.class);
    xstream.alias("Contactos", ListaContactos.class);
    xstream.addImplicitCollection(ListaContactos.class, "lista");

    try {
        ListaContactos contactos;
        // Recuperamos el objeto ListaContactos
        contactos = (ListaContactos) xstream.fromXML(new FileInputStream("Contactos.xml"));
        System.out.println("Numero de Contactos: " + contactos.getListContactos().size());
        List<Contacto> listaContactos = contactos.getListContactos();

        // iterador de la lista para recorrer cada contacto
        ListIterator<Contacto> iterador = listaContactos.listIterator();
        while (iterador.hasNext()) {
            Contacto c = (Contacto) iterador.next(); // Obtengo contacto
            System.out.println(
                "Nombre: " + c.getNombre()
                + ", apellidos: " + c.getApellidos()
                + ", email: " + c.getEmail());
        }
        System.out.println("Fin de listado ");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

Dudas y preguntas

