Pràctica 1: SeriesTimeUB: Iteració 1 - Parser XML

DS - curs 2018-2019

XML parser

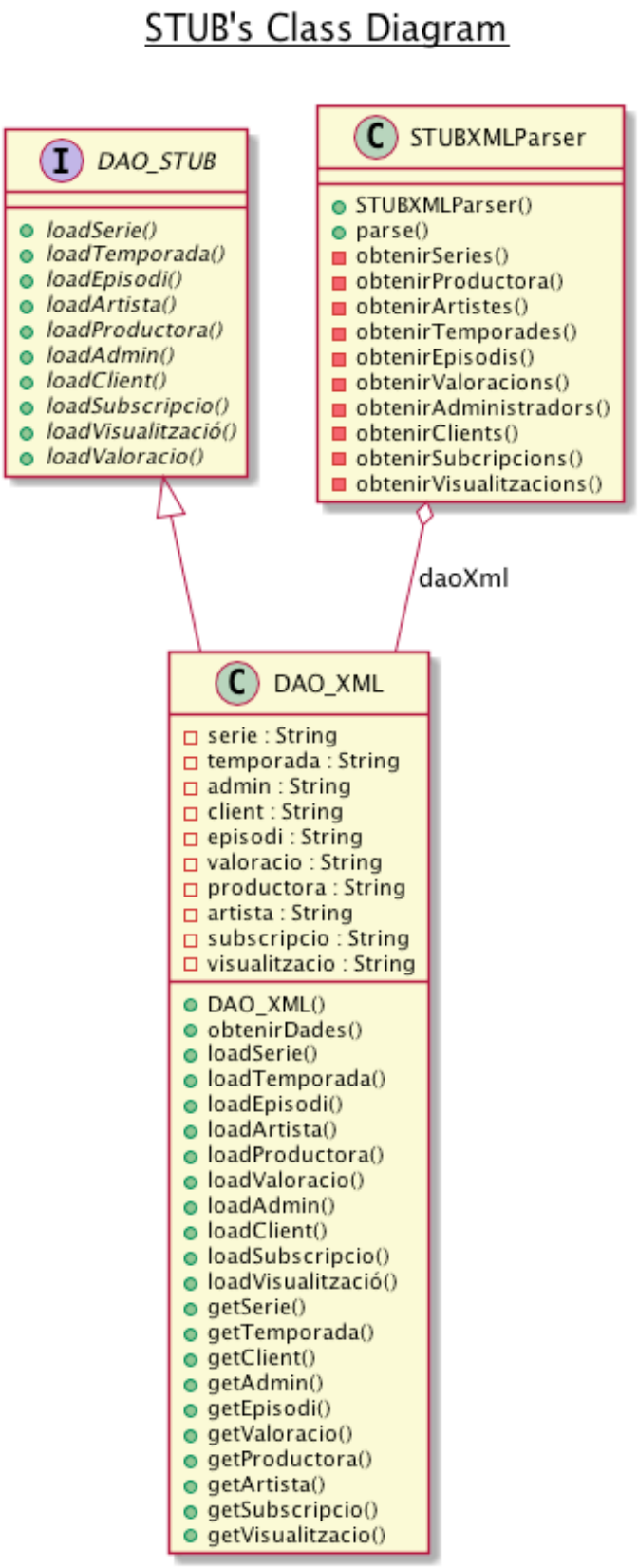
Per a realitzar l'entrega L2 de la pràctica, us facilitem els següents elements:

- Un fitxer XML amb les dades essencials que necessiteu per a treballar amb la vostra aplicació (stUB.xml de la carpeta data).
- Un DAO (Data Access Object) per a obtenir les dades del fitxer XML facilitat (interfície DAO_STUB)
- Un projecte de IntelliJ com a exemple de com executar els tests a Concordion de les dades llegides amb un exemple de DAO que implementa l'accés al fitxer XML (classe DAO_XML).

Mitjançant aquests elements haureu de carregar un set de dades inicials al vostre sistema per a fer proves de la vostra aplicació. A continuació us expliquem l'estructura del projecte proporcionat i el procés que heu de seguir.

Estructura del projecte

- doc
    - `El javadoc de l'aplicació (es pot generar el javadoc amb Tools -> Generate JavaDoc)`
- Source Packages
    - `DAO_STUB.java`
    - `DAU_XML.java`
    - `STUBXMLParser.java`
- Test Packages
    - `STUBDataManager.html`
    - `STUBDataManagerTest.java`
    - `STUBDataManagerSuite.html`
    - `STUBDataManagerSuite.java`
- data (fitxer XML amb les dades a llegir)
    - `StUB.xml`


STUB's Class Diagram
DAO_STUB
loadSerie()
loadTemporada()
loadEpisodi()
loadArtista()
loadProductora()
loadAdmin()
loadClient()
loadSubscripcio()
loadVisualització()
loadValoracio()
STUBXMLParser
STUBXMLParser()
parse()
obtenirSeries()
obtenirProductora()
obtenirArtistes()
obtenirTemporades()
obtenirEpisodis()
obtenirValoracions()
obtenirAdministradors()
obtenirClients()
obtenirSubcripcions()
obtenirVisualitzacions()
daoXml
DAO_XML
serie : String
temporada : String
admin : String
client : String
episodi : String
valoracio : String
productora : String
artista : String
subscripcio : String
visualitzacio : String
DAO_XML()
obtenirDades()
loadSerie()
loadTemporada()
loadEpisodi()
loadArtista()
loadProductora()
loadValoracio()
loadAdmin()
loadClient()
loadSubscripcio()
loadVisualització()
getSerie()
getTemporada()
getClient()
getAdmin()
getEpisodi()
getValoracio()
getProductora()
getArtista()
getSubscripcio()
getVisualitzacio()

Procés a seguir

1. Compileu els fitxers que hi ha a dins del directori `src`. Si feu servir IntelliJ, podeu obrir el projecte directament.

2. Executeu el test que us proporcionem amb Concordion (botó dret a sobre del projecte STUB->test ó Crtl+Mj+F6)

3. Es presenta el patró DAO implementat per accedir a un framework (i no a una llibreria). Fixeu-vos com:

    a. El DAO_XML, a dins del seu mètode constructor crea un parser per a obtenir les dades del fitxer XML, i tot seguit parseja les seves dades.

```
public DAO_XML(String nomFitxer) {
      STUBXMLParser parser = new STUBXMLParser(this);
      parser.parse(nomFitxer);
}
```

    b. El parser escaneja el fitxer XML i va fent crides als mètodes `loadClient()`, `loadSerie()` etc. de la classe `DAO_XML`. Els mètodes d'aquesta última classe es limiten a guardar les dades en variables privades. Fixeu-vos que `DAO_XML` és una implementació de la interfície `DAO_STUB`, on aquests mètodes estan definits.

4. Heu de crear la classe `DAO_XML_STUB`, que és on vosaltres manegareu les dades i que implementarà l'interfície `DAO_STUB` amb les vostres dades. Seguint l'exemple que us hem proporcionat, creeu a la vostra aplicació un objecte `DAO_XML`.

5. A continuació crideu al mètode constructor del `STUBXMLParser`, passant-li per paràmetre la ruta del fitxer XML.

6. El parser anirà cridant en ordre als corresponents mètodes de creació: `loadSerie()`, `loadClient()`, etc..

7. A dins de cadascun d'aquests mètodes és on vosaltres haureu de guardar les dades passades per paràmetre a les estructures de dades o classes que creieu convenients.

A tenir en compte

Podeu incrementar el número de paràmetres, podeu canviar el tipus de dades que retorna el mètode, etc. sempre i quan genereu el javadoc corresponent i ho expliqueu a la memòria.

Podeu també afegir més casos en el fitxer XML, com diferents valoracions, visualitzacions, subscripcions, etc. per a poder provar els vostres tests d'acceptació.

Exemple

Un breu exemple per a que acabeu de veure com podeu obtenir les dades. En verd i negreta podeu veure el codi afegit a la classe:

```
public class STUB_XML_Exemple {

	List<Serie> series = new ArrayList<Serie>();

	. . .

	public void loadSerie(String id, String title, String desc){

		Serie serie = new Serie(id, title, desc);

		series.add(serie);

	}
```

En aquest exemple, però la llista de Series és un atribut intern del STUB_XML_Exemple, fet que vulnera el principi de Dependència de codi ja que STUB_XML_Exemple manega internament la llista de sèries i en cas de volguer-se canviar la seva implementació de llista per un altre tipus, caldria modificar el codi intern de STUB_XML_Parser. Com modificareu el codi per a no vulnerar-lo?