

# РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ АНАЛІЗУ СТРУКТУРИ XML - ФАЙЛІВ



Виконав: студент групи 1 КС – 15 сп  
Мединський В. В.  
Керівник: к.т.н., старший викладач каф. ОТ  
Трояновська Т. І.

# Предмет роботи програми

Предметом роботи даної програми є перегляд файлів, складених в форматі XML, який за останні роки став одним із базових носіїв даних практично у всіх галузях ІТ-індустрії, включаючи ігрову сферу та бізнес-логіку. Цей формат є текст-орієнтованим і прекрасно читається людиною із мінімальними технічними знаннями.

# Етапи створення програми

Створення програми мовою програмування Java поділяється на 4 основних етапи.

- Розробка головного класу програми.
  - Розробка користувацького інтерфейсу.
  - Реалізація основних функцій.
  - Головний клас програми відрізняється від інших класів наявністю так званої «головної функції», яка є точкою входу в програму і виконується першою при запуску.
- Використаємо для створення програми інтегроване середовище розробки JDeveloper.

# Застосування правил створення XML-документа до процесів аналізу його структури

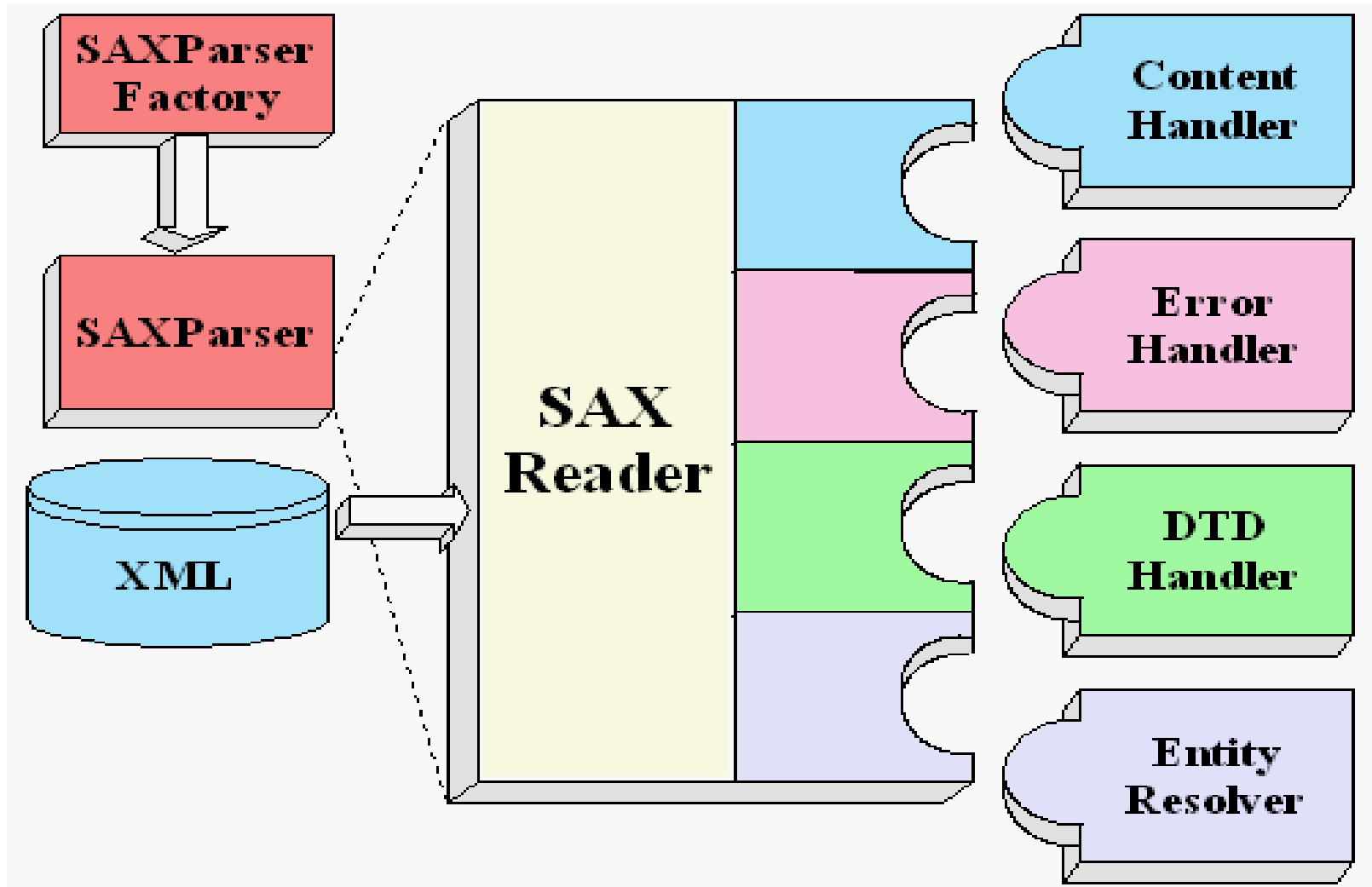
У загальному випадку XML- документи повинні задовольняти таким вимогам:

- У заголовку документа поміщається оголошення XML, у якому вказується мова розмітки документа, номер її версії і додаткова інформація.
- Кожний відкриваючий тег, що визначає деяку область даних у документі обов'язково повинний мати відповідний закриваючий тег.
- У XML враховується реєстр символів
- Всі значення атрибутів, використовуваних у визначенні тегів, повинні бути взяті в лапки.
- Вкладеність тегів у XML строго контролюється, тому необхідно стежити за порядком слідування відкриваючих і закриваючих тегів.
- Вся інформація, що розташовується між початковим і кінцевими тегами, розглядається в XML як дані і тому враховуються всі символи форматування.

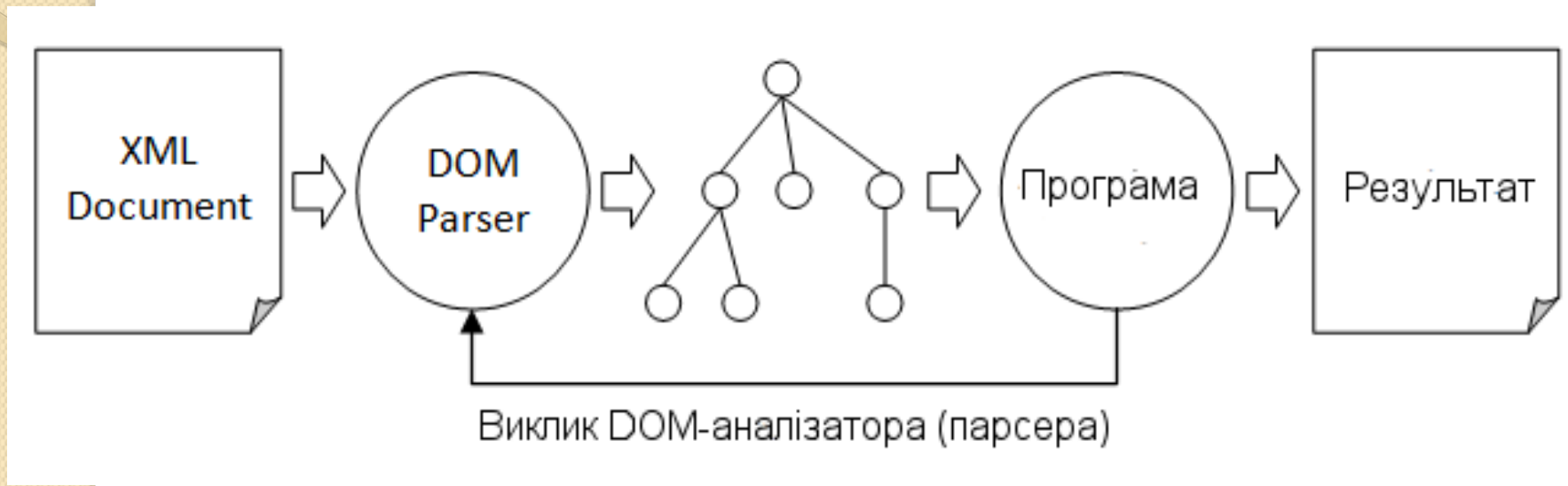
# Аналіз XML-документа та перевірка його на правильність

Для аналізу XML-документів використовується компонент, який носить назву JAXP – Java API for XML Processing. Він містить в собі два варіанти аналізаторів (парсерів) XML-файлів: так званий SAX (Simple API for XML) та DOM (Document Object Model)

# Принцип роботи SAX-аналізатора

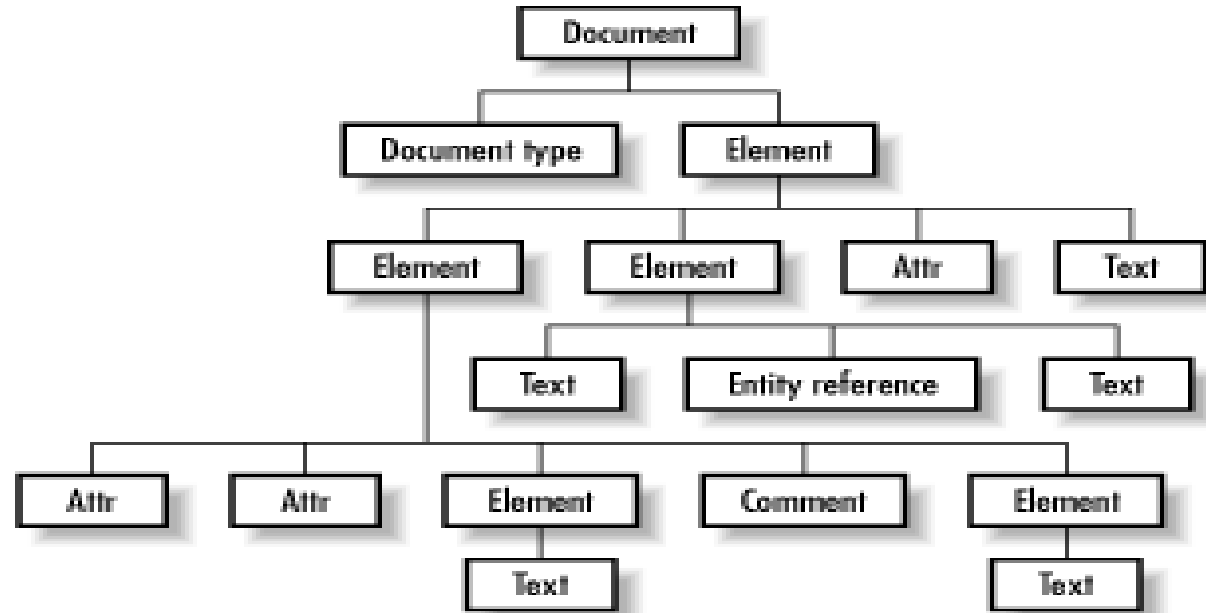


# Принцип роботи DOM-аналізатора



# Структура елементів типу даних DOM Document

Однак для візуалізації його змісту нам необхідно не тільки власне представлення документу, а й його деревовидне представлення. Дерево, як нам відомо із базового курсу алгоритмів, створюється за допомогою рекурсивного перегляду структури і перенесення відповідних даних в елемент керування, який це дерево здатен відобразити.

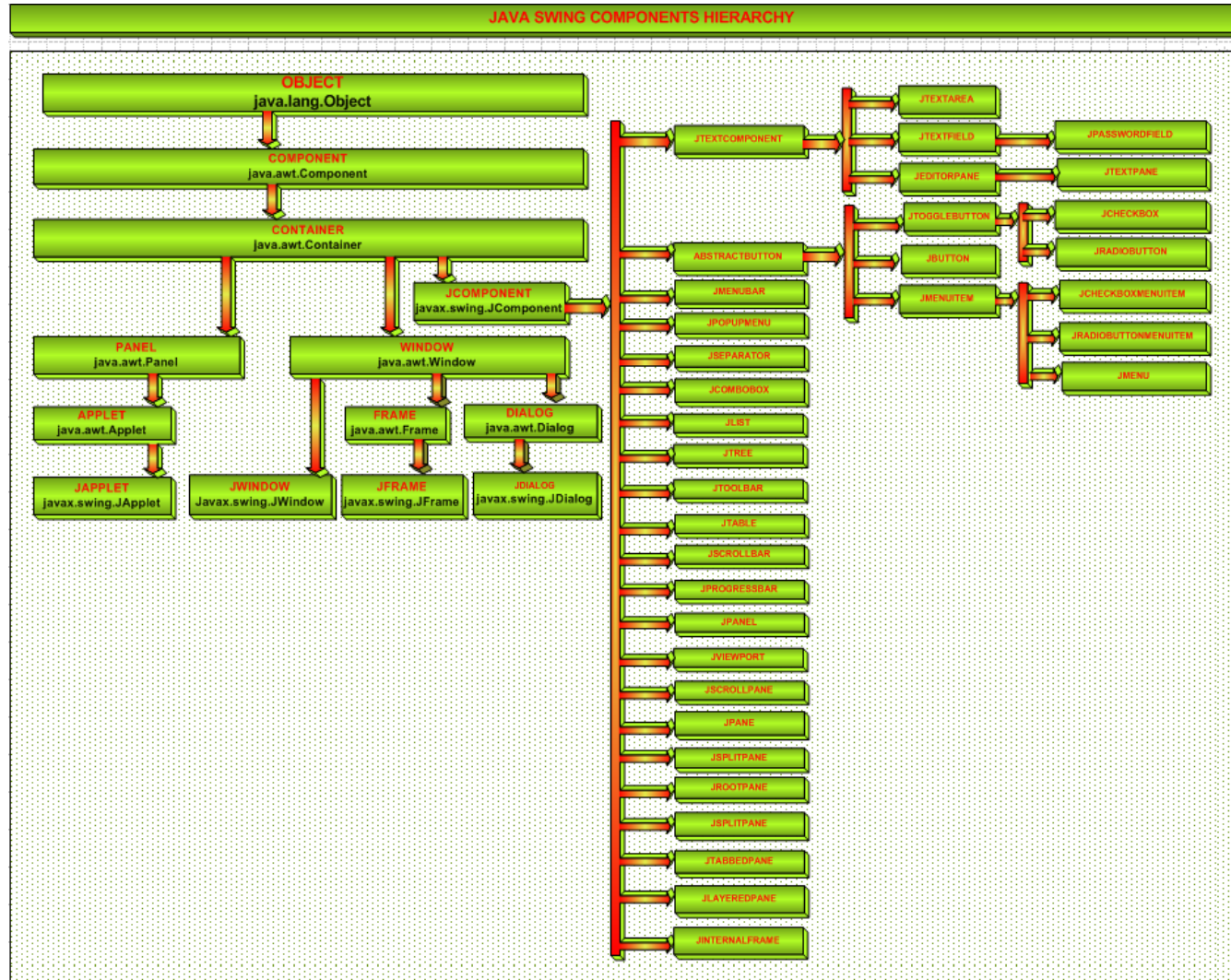




# Рекурсивне завантаження всіх елементів документа засобами бібліотеки Swing

Swing – один із найбільш комплексних компонентів мови програмування Java. Він містить повний набір елементів керування для конструювання графічної програми будь-якого призначення.

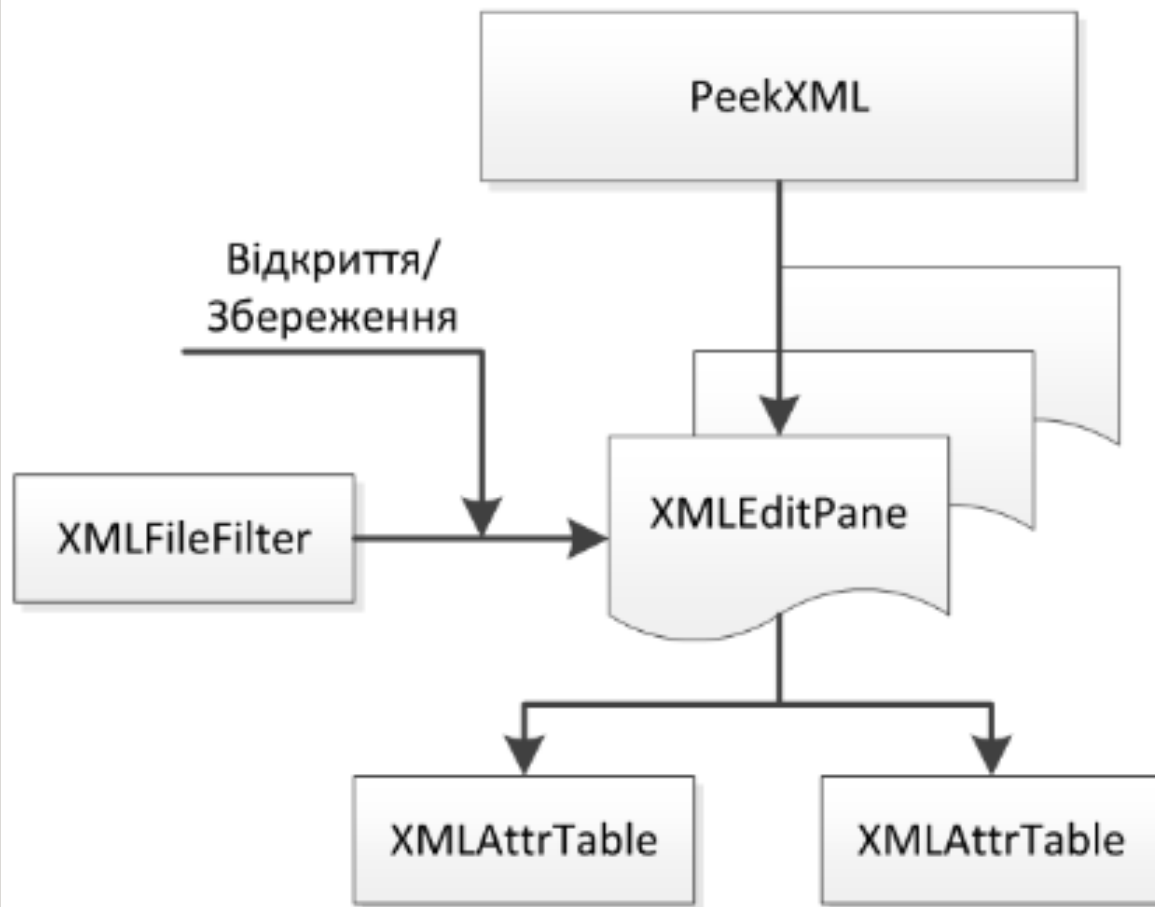
# Структура пакета Swing



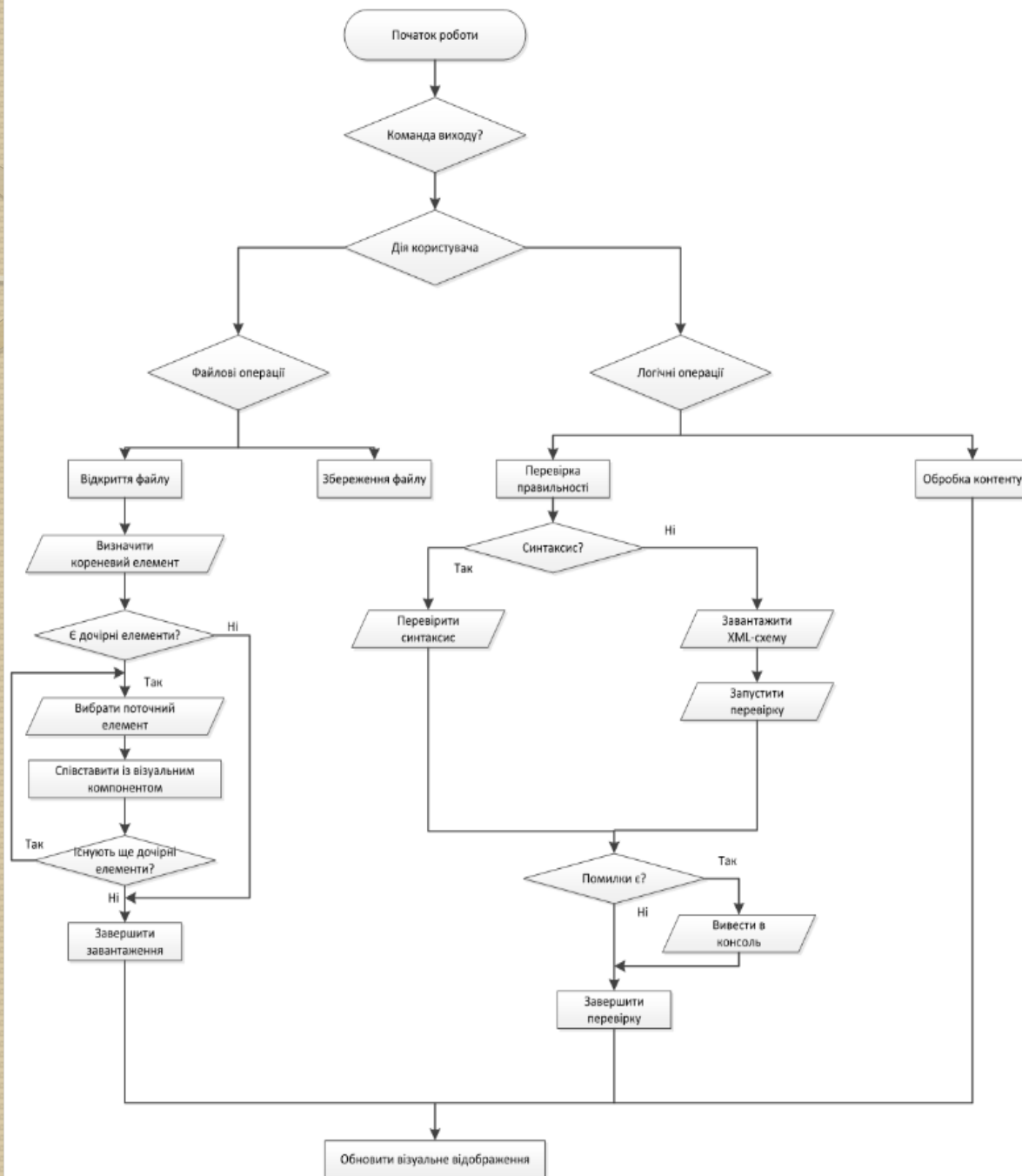
# Рекурсивне співставлення DOM- дерева дереву елемента керування

```
private void traverseXML(Element root,  
DefaultMutableTreeNode treeRoot)  
{  
if(root.hasChildNodes())  
{  
NodeList subs = root.getChildNodes();  
for(int i = 0; i < subs.getLength(); i++)  
if (subs.item(i) instanceof Element)  
{  
DefaultMutableTreeNode newNode = new  
DefaultMutableTreeNode(new  
XMLTreeNode((Element)subs.item(i)));  
traverseXML((Element)subs.item(i),  
newNode);  
treeRoot.add(newNode);  
}  
}  
}
```

# Діаграма класів програми аналізу структури XML-файлів

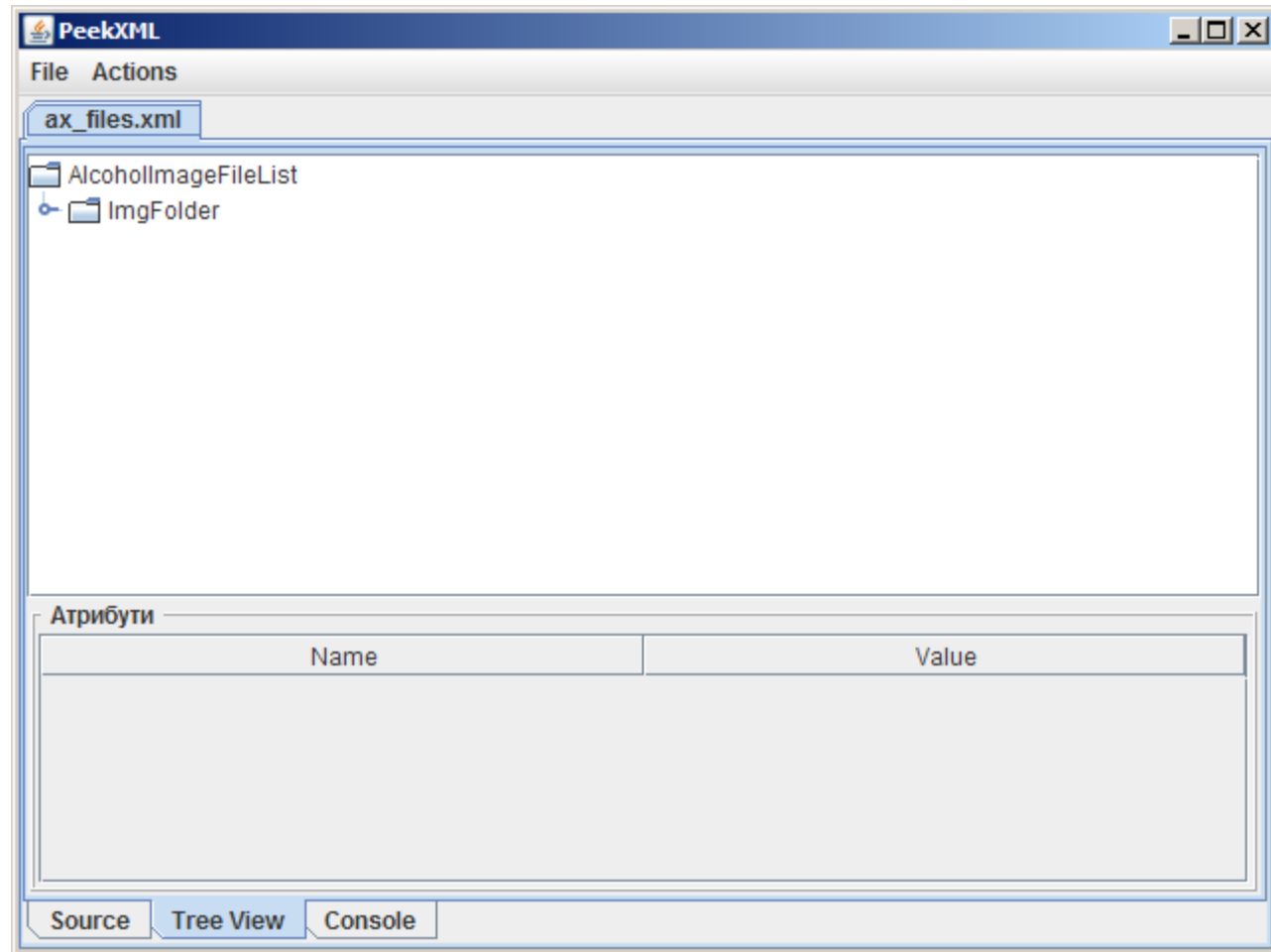


Таким чином, у результаті використання бібліотеки Swing отримаємо таку діаграму класів програмного засобу, що розробляється.



# Алгоритм роботи програми

# Вкладковий інтерфейс



# Створення головного класу

**Create Java Class**

Enter the details of your new class.

Name:

Package:

Extends:

Optional Attributes

Implements:

- ActionListener (java.awt.event)

Access Modifiers

- public
- package protected

Other Modifiers

- <None >
- abstract
- final

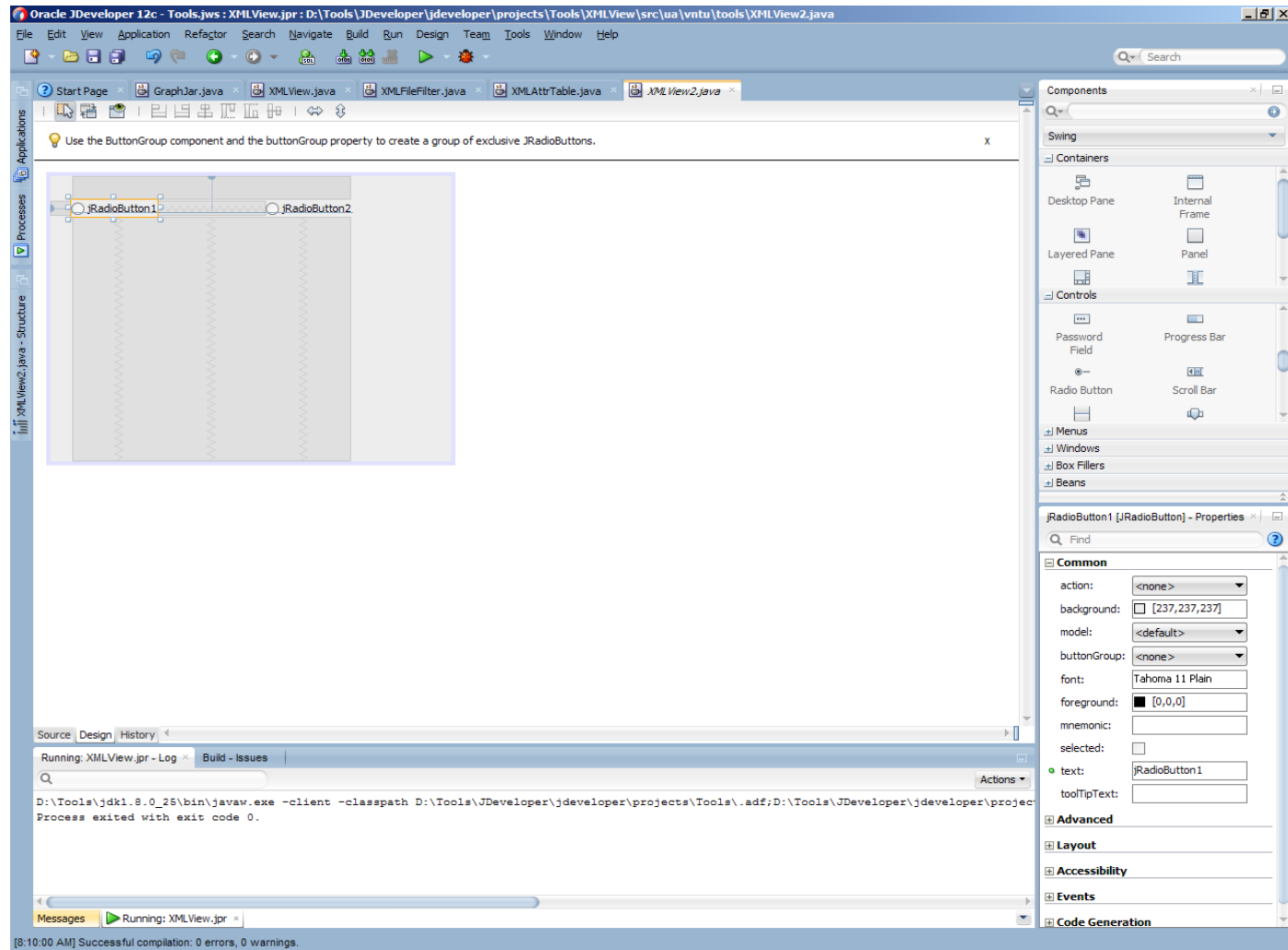
Constructors from Superclass

Implement Abstract Methods

Main Method

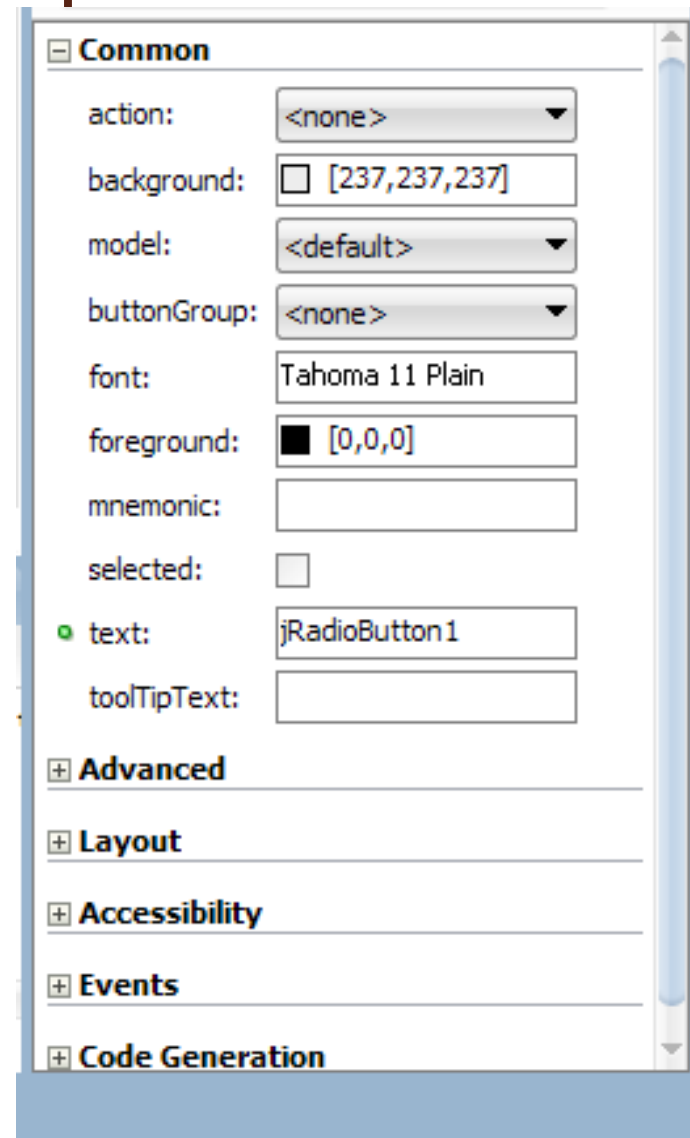
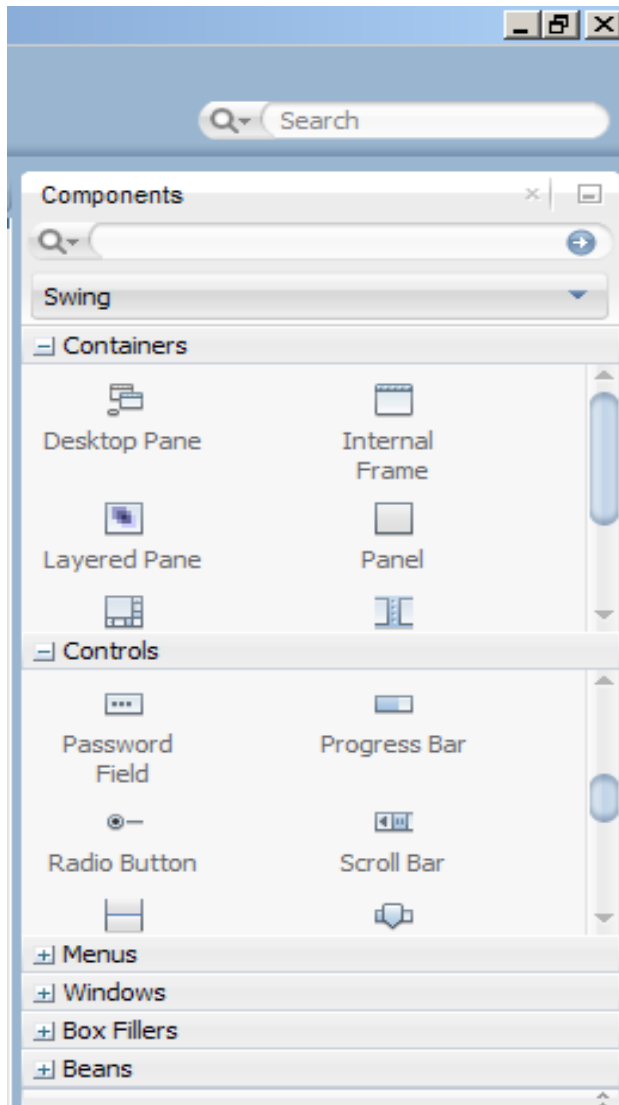
Help OK Cancel

# Візуальний редактор форм JDeveloper

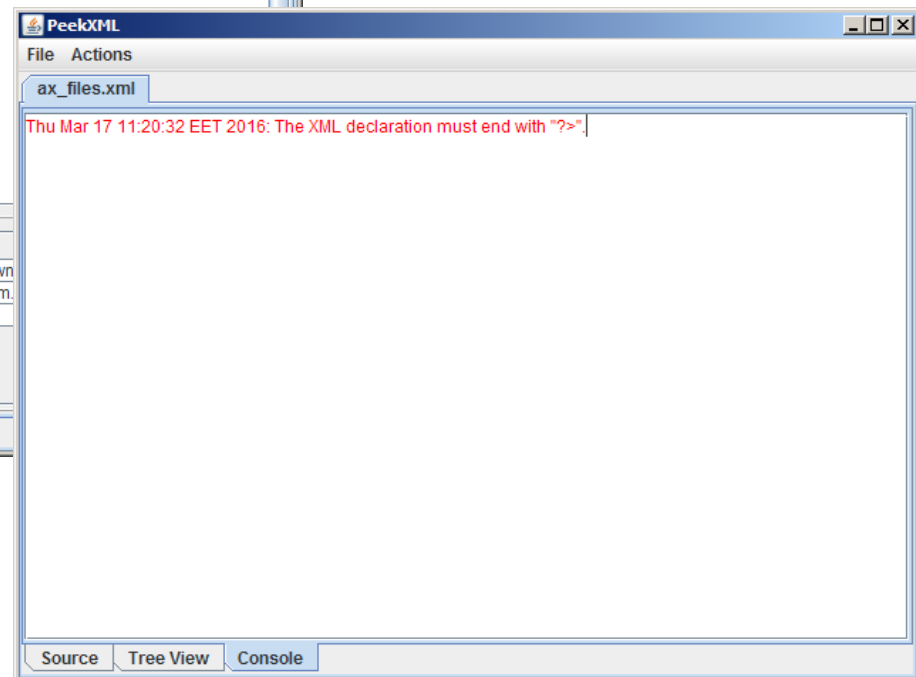
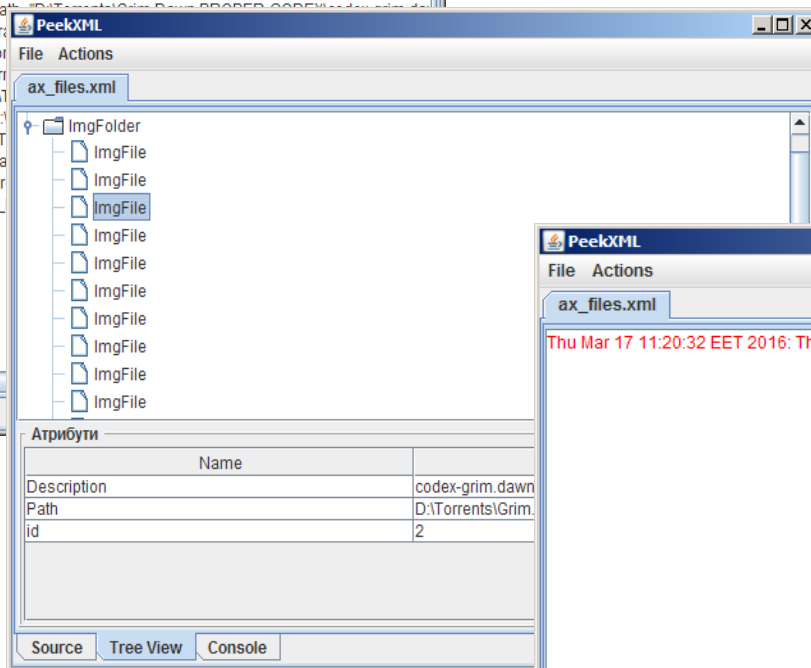
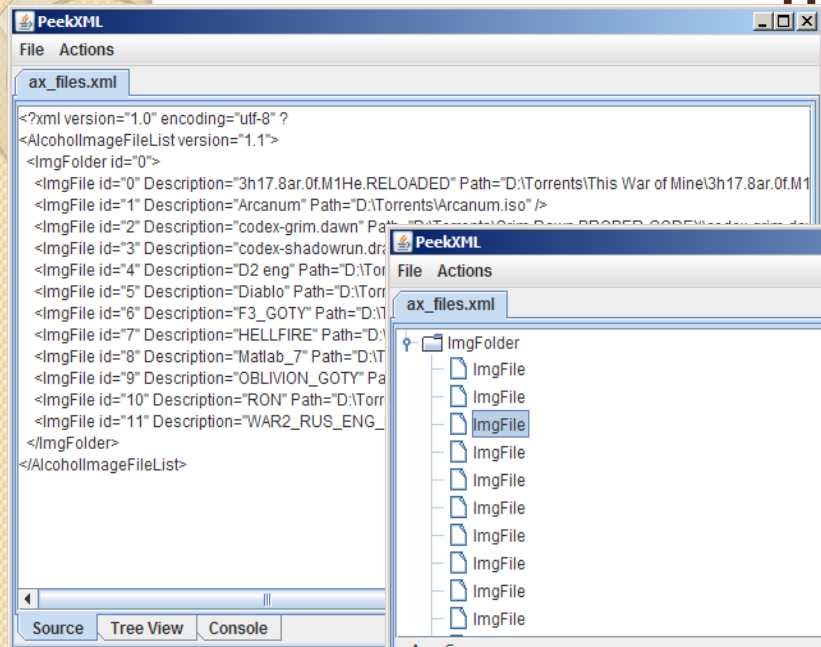




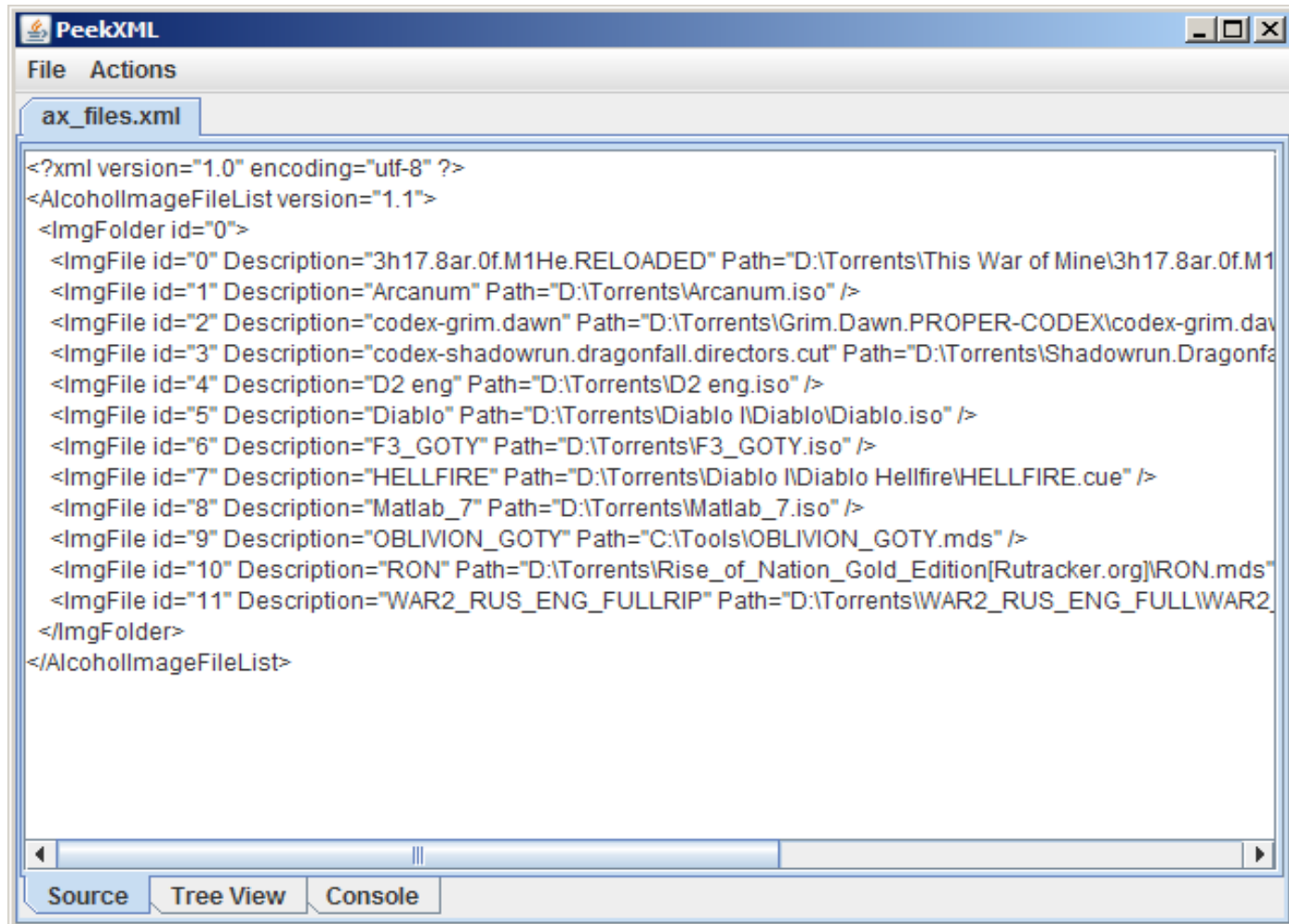
# Засоби створення та редагування форм



# Дизайн користувачького інтерфейсу



# Відкриття XML-документу в режимі редагування тексту

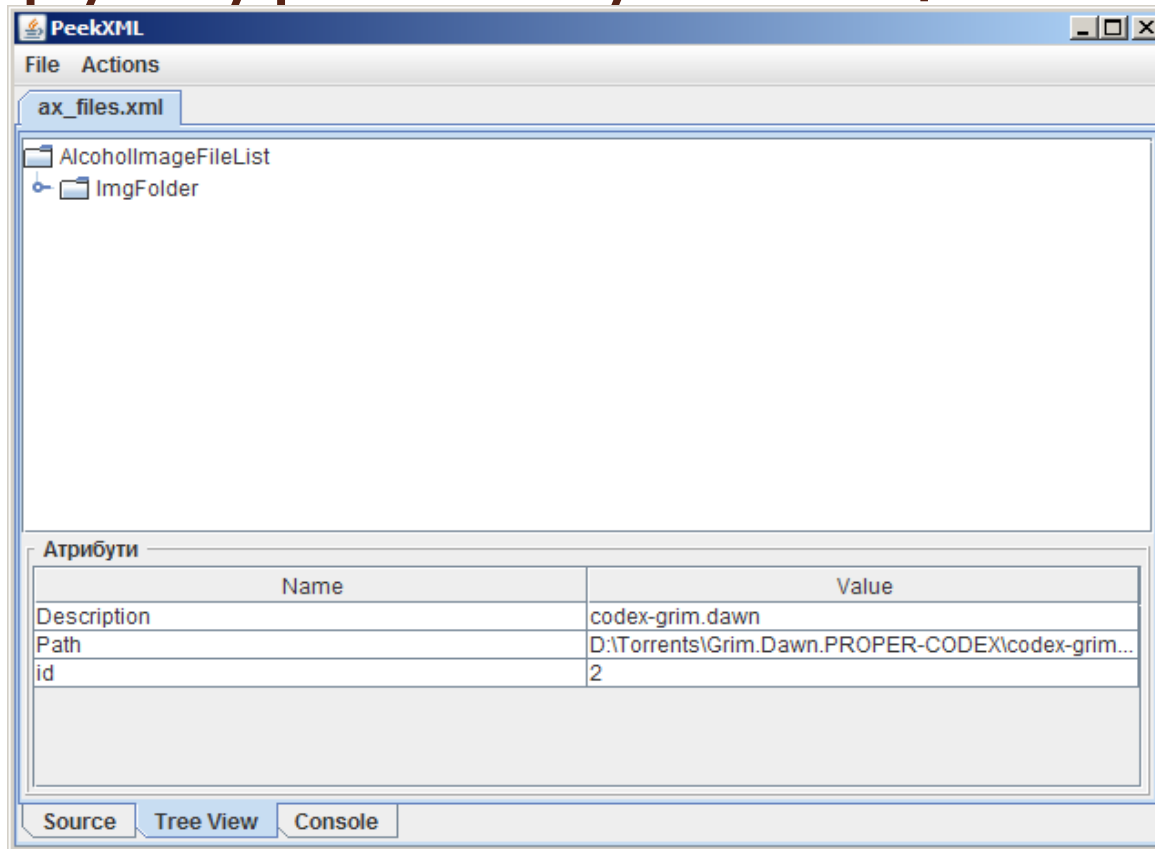


The screenshot shows the PeekXML application window. The title bar reads "PeekXML". Below the title bar is a menu bar with "File" and "Actions". A tab labeled "ax\_files.xml" is active. The main text area contains the following XML code:

```
<?xml version="1.0" encoding="utf-8" ?>
<AlcoholImageFileList version="1.1">
  <ImgFolder id="0">
    <ImgFile id="0" Description="3h17.8ar.0f.M1He.RELOADED" Path="D:\Torrents\This War of Mine\3h17.8ar.0f.M1
    <ImgFile id="1" Description="Arcanum" Path="D:\Torrents\Arcanum.iso" />
    <ImgFile id="2" Description="codex-grim.dawn" Path="D:\Torrents\Grim.Dawn.PROPER-CODEX\codex-grim.dav
    <ImgFile id="3" Description="codex-shadowrun.dragonfall.directors.cut" Path="D:\Torrents\Shadowrun.Dragonfa
    <ImgFile id="4" Description="D2 eng" Path="D:\Torrents\D2 eng.iso" />
    <ImgFile id="5" Description="Diablo" Path="D:\Torrents\Diablo \Diablo\Diablo.iso" />
    <ImgFile id="6" Description="F3_GOTY" Path="D:\Torrents\F3_GOTY.iso" />
    <ImgFile id="7" Description="HELLFIRE" Path="D:\Torrents\Diablo \Diablo Hellfire\HELLFIRE.cue" />
    <ImgFile id="8" Description="Matlab_7" Path="D:\Torrents\Matlab_7.iso" />
    <ImgFile id="9" Description="OBLIVION_GOTY" Path="C:\Tools\OBLIVION_GOTY.mds" />
    <ImgFile id="10" Description="RON" Path="D:\Torrents\Rise_of_Nation_Gold_Edition[Rutracker.org]\RON.mds"
    <ImgFile id="11" Description="WAR2_RUS_ENG_FULLLRIP" Path="D:\Torrents\WAR2_RUS_ENG_FULL\WAR2
  </ImgFolder>
</AlcoholImageFileList>
```

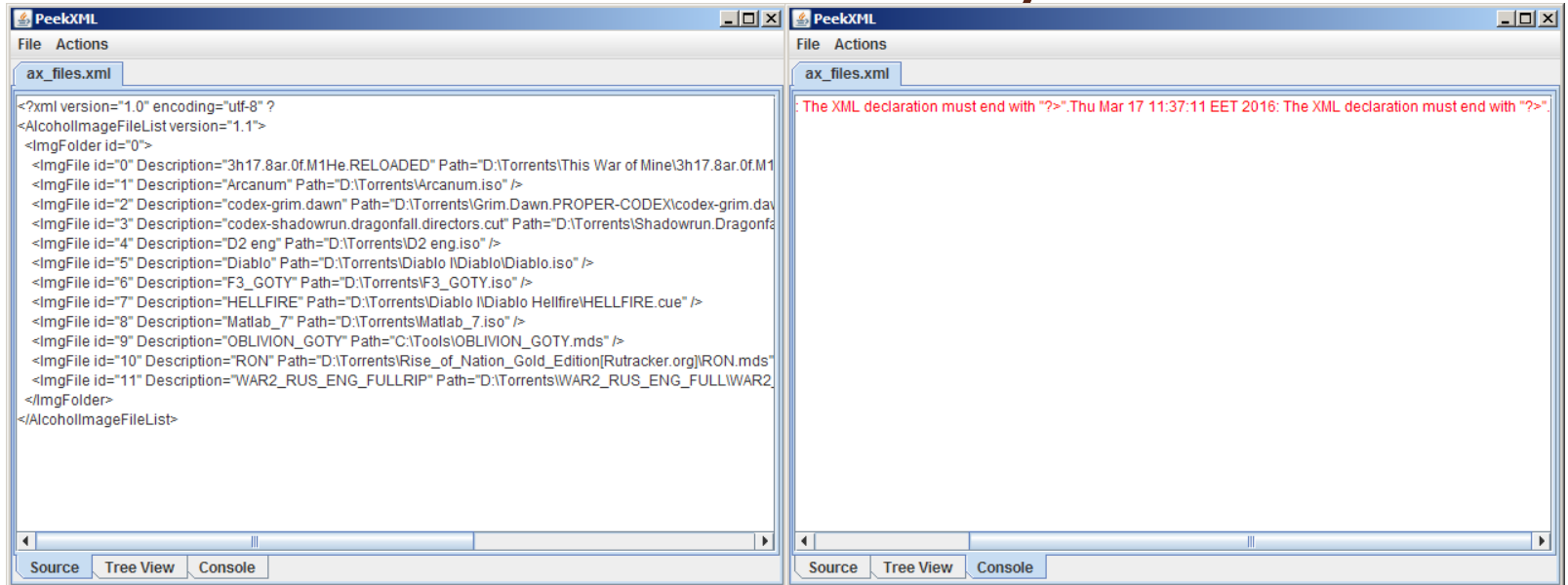
At the bottom of the window, there are three tabs: "Source" (selected), "Tree View", and "Console".

# Перегляд XML-файлу в режимі структурної візуалізації



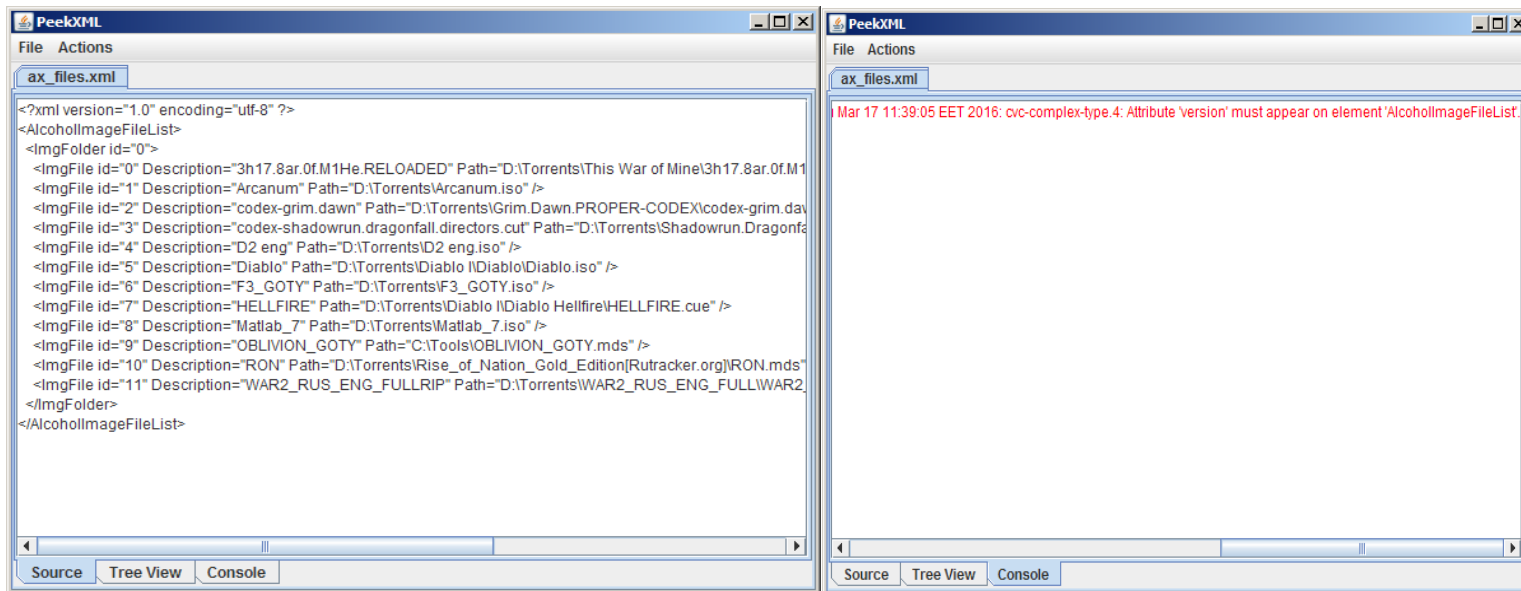
Тепер, переключивши вкладку на TreeView, розберемо файл за допомогою DOM-аналізатора, і перетворимо на візуальне дерево. Документ простенький, всього два рівні вкладеності, плюс прості текстові атрибути.

# Відображення помилки синтаксису



Внесемо помилку в документ і перевіримо, як вони знаходяться функціями перевірки.

# Результати помилки синтаксису



Завантажимо файл схеми документа і внесемо помилку (видалимо один атрибут). Це свідчить про те, що програма працює коректно у всіх режимах, і відображає структуру XML-документа відповідно запланованому функціоналу.

# Реалізація класів та функцій

До складу програми перегляду XML-файлів входить п'ять основних класів, що формують пакет `ua.vntu.tools`.

- `PeekXML`. Успадкований від класу головного вікна програми `JFrame`, і реалізує інтерфейси `ActionListener`. Представляє собою головний клас програми, який власне і запускається при запуску програми.
- `XMLEditPane`. Успадкований від класу `JPanel`, і містить все необхідне для побудови вкладкового інтерфейсу та реалізації основних функцій візуалізації та обробки контенту XML-документа.
- `XMLAttrTable`. Успадкований від класу таблиці `JPanel` і створений для простоти маніпуляцій із динамічною таблицею даних, адже відображення атрибутів необхідно робити «на льоту».
- `XMLFileFilter`. Успадкований від класу `FileFilter`. Службовий клас, який дозволяє при виборі файлів відфільтрувати тільки каталоги та XML-документи;
- `XMLTreeNode`. Службовий внутрішній клас, який використовується як обгортка XML-елемента для використання в таблиці.

**ДЯКУЮ ЗА УВАГУ.  
ДОПОВІДЬ  
ЗАВЕРШЕНО!**