Алгоритм

Для подання інформації про автоматизовані об’єкти в програмі були використані класи: клас Country, клас City. Для зберігання множини об’єктів були використані динамічні масиви. Управління динамічними масивами було здійснено за допомогою класу java.util.ArrayList. Для управління об'єктами був передбачений окремий клас – клас Worldmap. Даний клас містить масиви об'єктів і надає методи для редагування об'єктів, а також для завантаження і збереження даних в XML-файл. Для зберігання інформації про автоматизовані об'єкти використовується один XML-файл. Файл має ієрархічну структуру, яка відображає залежності між об'єктами.

Обробка документа XML всередині програми здійснювалася за допомогою DOM парсера. DOM (Document Object Model – об’єктна модель документа) – використовується для деревовидного представлення інформації, що зберігається в документі XML. DOM включає в себе набір інтерфейсів, що містяться в пакеті org.w3c.dom, зокрема:

* Document – представлення документа XML
* Element – елемент XML
* Text – текстовий рядок

Перед роботою з документом XML, був створений парсер для документа (javax.xml.parsers.DocumentBuilder). Зроблено це було за допомогою спеціальної фабрики парсерів (javax.xml.parsers.DocumentBuilderFactory). Після того як парсер створений, був створений документ на основі файлу xml. Щоб отримати кореневий елемент документа був використаний метод getDocumentElement. Для отримання назви елемента був використаний метод getTagName. Для отримання дочірніх елементів, що містяться всередині даного елемента, був використаний метод getElementsByTagName. Даний метод повертає колекцію елементів із заданим ім'ям. Для отримання значення атрибута був використаний метод елемента getAttribute. Для отримання тексту, що міститься всередині елемента (і всіх його дочірніх елементів) був використаний метод getTextContent.

Валідація документа XML – перевірка коректності структури документа XML. Валідація була здійснена на основі опису DTD. Для валідації документа XML за описом DTD було налаштовано відповідним чином фабрику парсерів і парсер. Для фабрики парсерів було за допомогою методу setValidating включена перевірку структури документа. Для парсера було розроблено спеціальний обробник помилок, що підтримує інтерфейс org.xml.sax.ErrorHandler. Даний обробник визначає дії, які будуть проводитися парсером при виникненні помилок в документі. Перед викликом методу parse, був створений екземпляр обробника і зареєстрований в парсері. Слід зазначити, що посилання на необхідний опис DTD був розміщений всередині документа XML.

Потім був сформований документ XML на основі наявних даних з використанням дерева DOM. Щоб зберегти отримане дерево в файл, був використаний Transformation API для XML (пакет javax.xml.transform).