

Лабораторная работа № 1.

«Правильно построенные XML-документы»

Продолжительность выполнения лабораторной работы: 2 ак. часа.

Целью лабораторной работы является знакомство с языком XML, который используется на платформе Android для описания компоновок и других ресурсов, а также с правилами построения xml-файлов.

Теоретическая часть

XML (англ. eXtensible Markup Language, расширяемый язык разметки) — это текстовый формат, который предназначен для хранения структурированных данных, обмена информацией между программами и создания на его основе специализированных языков разметки. Является подмножеством языка SGML.

XML-документ имеет иерархическую структуру в виде дерева с элементами, вложенными в другие элементы, и с единственным элементом верхнего уровня – корневым элементом. В документе могут содержаться элементы с одинаковыми именами. В элементе могут содержаться атрибуты, имена атрибутов внутри одного элемента не должны повторяться. Также внутри элемента может располагаться текстовое содержимое.

Листинг 1. Файл strings.xml.

```
<resources>  
    <string name="app_name">NewApp</string>  
</resources>
```

В листинге 1 приведён код файла strings.xml, автоматически сгенерированного средой разработки Android Studio. В данном файле указываются строковые ресурсы. Корневым элементов является элемент <resources>. В него может быть вложено любое количество элементов <string>, каждый из которых задаёт один строковый ресурс. Для каждого строкового ресурса указывается значение в содержимом элемента и имя ресурса в атрибуте name.

Как следует из названия, XML является расширяемым языком, это означает, что можно создавать элементы и атрибуты с любыми названиями. Но, естественно, для

каждой задачи, например, для разработки под Android, оговариваются допустимые имена элементов и их атрибутов.

Существует два уровня правильности XML-документов: правильно построенные (англ. well-formed, корректно сформированные) и действительные или валидные (англ. valid).

Правильно построенные документы соответствуют общим правилам синтаксиса XML. Если XML-документ построен неправильно, то XML-парсер обязан классифицировать ситуацию как фатальную ошибку.

Валидный XML-документ должен соответствовать некоторым семантическим правилам. Это более строгая дополнительная проверка корректности документа на соответствие заранее определённым, но уже внешним правилам, в целях минимизации количества ошибок. Обычно семантические правила хранятся в специальных файлах — схемах (форматы DTD или XML Schema), где подробно описывается структура документа, все допустимые названия элементов, атрибутов, допустимое количество элементов, и многое другое.

На платформе Android, например, при создании любого ресурса в XML-документе в корневом элементе необходимо указывать ссылку на схему в атрибуте `xmlns:android="http://schemas.android.com/apk/res/android"`. Если ошибиться в названии какого-либо элемента или атрибута, приложение запускаться не будет.

Правильное построение документов XML

Визуально структура xml-документа может быть представлена как дерево элементов. Элементы XML описываются тегами. Рассмотрим правила построения XML-документов.

1. В XML версии 1.1 документ начинается с объявления XML

Объявление XML — это строка, указывающая версию XML, которая располагается перед корневым элементом. В версии 1.0, которая используется на Android, объявление XML может быть опущено, в версии 1.1 оно обязательно. Также в объявлении XML может быть указана кодировка символов и наличие внешних зависимостей.

```
<?xml version="1.0" encoding="UTF-8"?>
```

2. XML-документ должен содержать строго один корневой элемент. То есть текст или другие данные всего документа должны быть расположены между единственным начальным корневым тегом и соответствующим ему конечным тегом.

3. Все элементы должны начинаться с открывающего (начального) тега, например, <color>, и заканчиваться закрывающим (конечным) тегом, например, </color>. Всё, что расположено между ними: другие элементы или текст, называется содержимым элемент. Если у элемента нет содержимого, то такой элемент называют пустым. У пустого элемента могут быть указаны атрибуты. Для пустых элементов можно применять особую форму записи, в которой перед закрывающей угловой скобкой ставится косая черта, тогда использовать закрывающий тег не нужно.

Листинг 2. Пустой элемент TextView.

```
<TextView  
    android:id="@+id/text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

4. Имена элементов и имена атрибутов не могут содержать пробелы, но могут быть на любом языке, поддерживаемом кодировкой XML-документа. Имя может начинаться с буквы, подчёркивания, двоеточия. Остальными символами имени могут быть те же символы, а также цифры, дефис, точка.

5. Теги XML являются регистрозависимыми.

6. Значения атрибутов всегда заключаются в кавычки (одинарные или двойные), одно и то же имя атрибута не может встречаться дважды в одном элементе. Не следует использовать разные типы кавычек для значений атрибутов одного тега. На платформе Android по умолчанию используются двойные кавычки (код символа в Юникод U+0022).

7. XML не допускает перекрывающихся элементов. Например, данный код некорректен: <A>.

8. Все используемые теги обязательно должны быть закрыты.

9. Спецсимволы надо экранировать. В XML существуют два метода записи специальных символов: ссылка на сущность и ссылка по номеру символа.

Сущностью (англ. entity) называются именованные данные, обычно текстовые, в частности, спецсимволы. Ссылка на сущность указывается в том месте, где должна быть сущность и состоит из амперсанда (&), имени сущности и точки с запятой (;).

В XML предопределены 5 сущностей (таблица 1).

Таблица 1. Предопределённые сущности в XML.

Название	Символ	Ссылка на сущность
Амперсанд	&	&
Знак меньше	<	<
Знак больше	>	>
Апостроф (одинарная кавычка)	'	'
Двойная кавычка (универсальная)	"	"

Листинг 3. Пример экранирования символов.

```
<string name="big_text">Это большой текст, в котором встречаются  
различные спецсимволы, например, амперсанд - &amp;, знаки больше-  
меньше &lt; &gt;, одинарная (&apos;) и двойная (&quot;)  
кавычка!</string>
```

Это большой текст, в котором
встречаются различные спецсимволы,
например, амперсанд - &, знаки
больше-меньше < >, одинарная (') и
двойная (") кавычка!

Рисунок 1. Текстовое поле, в котором выводится текстовый ресурс big_text.

Иногда бывает необходимо определить неразрывный пробел, который часто используется в HTML (). В XML такой предопределённой сущности нет, поэтому надо указывать ссылку по номеру символа вместо .

Ссылка по номеру символа выглядит как ссылка на сущность, но вместо имени сущности указывается символ # и число (в десятичной или шестнадцатеричной записи), являющееся номером символа в кодовой таблице Юникод. Это обычно символы, которые невозможно закодировать напрямую, например, буква арабского алфавита в ASCII-кодированном документе. Амперсанд может быть представлен следующим образом: &.

10. Комментарии в XML-документ начинаются с символов <!-- и заканчиваются символами -->.

Задание

Рассмотрите документ Lab1.xml. В данном документе содержатся различные ошибки с точки зрения правильности построения XML-документов. Исправьте их. Проверить корректность документа, открыв его через любой web-браузер.

Придумайте свою структуру данных, которую можно описать с помощью XML, и создайте документ с различными элементами и их атрибутами (не менее 10 различных элементов), соблюдая правила построения XML-документов.

Контрольные вопросы:

1. Что такое XML?
2. Сколько корневых элементов может быть в XML-документе?
3. Что такое объявление XML в XML-документе?
4. Что такое пустой элемент в XML-документе и как его можно записывать?
5. Какие символы могут содержать имена элементов и атрибутов в XML-документе?
6. С каких символов могут начинаться имена элементов и атрибутов в XML-документе?
7. Теги XML являются регистрозависимыми или регистронезависимыми?
8. Допускается ли перекрытие элементов в XML-документе?
9. Как можно экранировать спецсимволы в XML?
10. Какие спецсимволы можно экранировать в XML с помощью ссылки на сущность?