

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования “Омский государственный технический университет”  
Кафедра “Автоматизированные системы обработки информации и  
управления”

**ОТЧЁТ ПО ДОМАШНЕМУ ЗАДАНИЮ**  
по дисциплине «Основы командной разработки»  
студентки Бабиковой Софии Денисовны, группы ПИН-252Т

Доцент, к.т.н.

О.Б. Малков

Студент

С. Д. Бабикова

Омск 2025

## **ВВЕДЕНИЕ**

Сегодня при командной разработке сложных программ следует использовать специальные инструменты, которые помогают отслеживать изменения в коде и файлах проекта. Они позволяют программистам работать вместе над одним проектом, возвращаться к прошлым версиям, добавлять комментарии и удобно управлять ветками.

Одним из самых популярных таких инструментов является Git - популярная система контроля версий. Она удобная, быстрая и помогает организовать работу над проектом для нескольких людей. На основе Git создана платформа GitHub, где можно хранить проекты, работать над ними совместно и делиться своими интересными результатами.

Умение пользоваться Git и GitHub - важный навык для любого разработчика. В этой работе нужно научиться создавать и копировать (клонировать) репозитории, работать с ветками - создавать, объединять, переключать и удалять их, а также синхронизировать все изменения между локальным и удалённым репозиториями.

## ЗАДАНИЕ

Создать XML-документ, представляющий информацию о знаменитых зданиях мировой архитектуры. Созданный документ должен содержать не менее 3 уровней вложенности включая корневой узел. Число элементов документа, не имеющих вложенных элементов, должно быть не менее пяти. Элементы документа должны содержать комментарии о своем содержании. По крайней мере, у двух разных уровней узлов должны быть атрибуты. Для описания структуры XML- документа необходимо сформировать DTD и XSD и связать их с XML-документом. Также необходимо создать XSL-документ, с помощью которого трансформировать XML-документ в HTML-документ.

## Организация работы в среде Git и на веб-платформе GitHub

В начале надо было создать удаленный репозиторий на платформе Git Hub и пригласить в свой репозиторий коллаборатора – доцента, к.т.н. Малкова О. Б., это было создано для того чтобы была возможность вести диалог и получать комментарии и работе. Далее нужно было установить git bash, и с помощью него связать удаленный репозиторий с локальным при помощи команды git clone. Для отслеживания порядка выполнения заданий был создан главный issue, названный «Разработка проекта на основе выданного задания». В нем были последовательно описаны пункты, которые надо было осуществить. План включал в себя такие пункты, как: прочитать источники информации о создании XML-документов и работе с ними, создать XML-документ, содержащий информацию о предметной области (мировые архитектурные сооружения), для описания структуры XML-документа сформировать DTD, связав его с XML-документом, сформировать XSD, связав его с XML-документом, создать XSL-документ, с помощью которого трансформировать XML-документ в HTML- документ. Также для каждого задания заводился отдельный issue , где точно описывались этапы выполнения. Были заведены такие issue, как создание XML-документа по предметной области, создание DTD для XML-документа, создание XSD схемы для XML-документа, создание XSL-документа для трансформации XML в HTML.

После этого была создана ветка feature2, в которую постепенно были добавлены нужные файлы. Подробнее об этом, сначала требовалось открыть локальный репозиторий в git bash и проверить, что мы находимся на ветке main с помощью команды git checkout main. Далее создавалась ветка feature2. После этого добавлялся нужный файл, добавлялся с помощью команды git add . и добавлялся комментарий изменения через команду git commit -m "", через команды git status и git log , можно было осуществлять проверку. Дальше с помощью команды git push -u origin feature2 изменения в локальном репозитории переносились в удаленный. Это было проделано с пятью файлами.

Сами файлы были написаны и связаны между собой в среде разработки VS Code. Проверить корректную работу сайта можно было с помощью расширения Live Server.

Далее на платформе GitHub можно было увидеть появившуюся кнопку Pull Request, при нажатии на которую можно было слить ветку feature2 в main. После сливания нужно прописать в Git Bash команду `git pull origin main`, предварительно убедившись, что мы находимся на ветке main. Это делалось для того чтобы изменения в удаленном репозитории, отразились и в локальном репозитории. После ветку feature2 надо было удалить с помощью команды `git branch -d feature`, что привело к удалению ветки на локальном репозитории и `git push origin --delete feature` – удаление ветки на удаленном репозитории.

Отчет был загружен по тому же принципу.

## XML-документ

На рисунке 1 представлен отрывок кода XML-документа. Полный код файла представлен в приложении А.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="iphones.xsl"?>
<iphone_sales xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="iphones.xsd">

  <analytics source="Statista" year="2024">
    <period>2014-2024</period>
    <total_models>5</total_models>
    <methodology>Анализ данных международных аналитических агентств</methodology>
  </analytics>

  <iphone_models category="top_sellers">

    <model name="iPhone 6">
      <specifications>
        <image>image/6 iphone.jpg</image>
        <release_year>2014</release_year>
        <display>4.7"</display>
        <sales>220</sales>
      </specifications>
    </model>

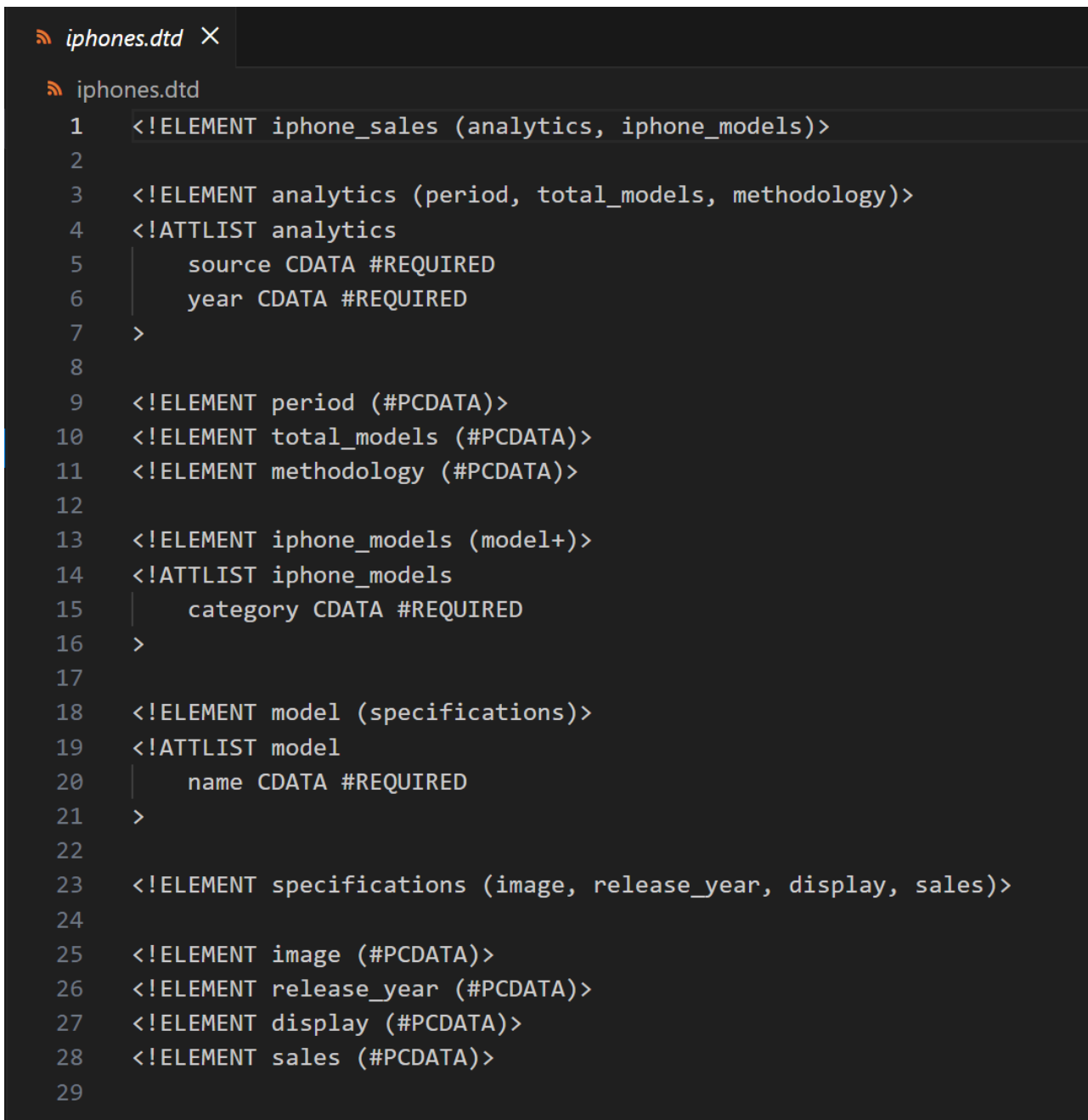
    <model name="iPhone 11">
      <specifications>
        <image>image/11 iphone.jpg</image>
        <release_year>2019</release_year>
        <display>6.1"</display>
        <sales>170</sales>
      </specifications>
    </model>

  </iphone_models>
</iphone_sales>
```

Рисунок 1 - XML-документ

## DTD-документ

На рисунке 2 представлен DTD документ. Код документа приведен в приложении Б.

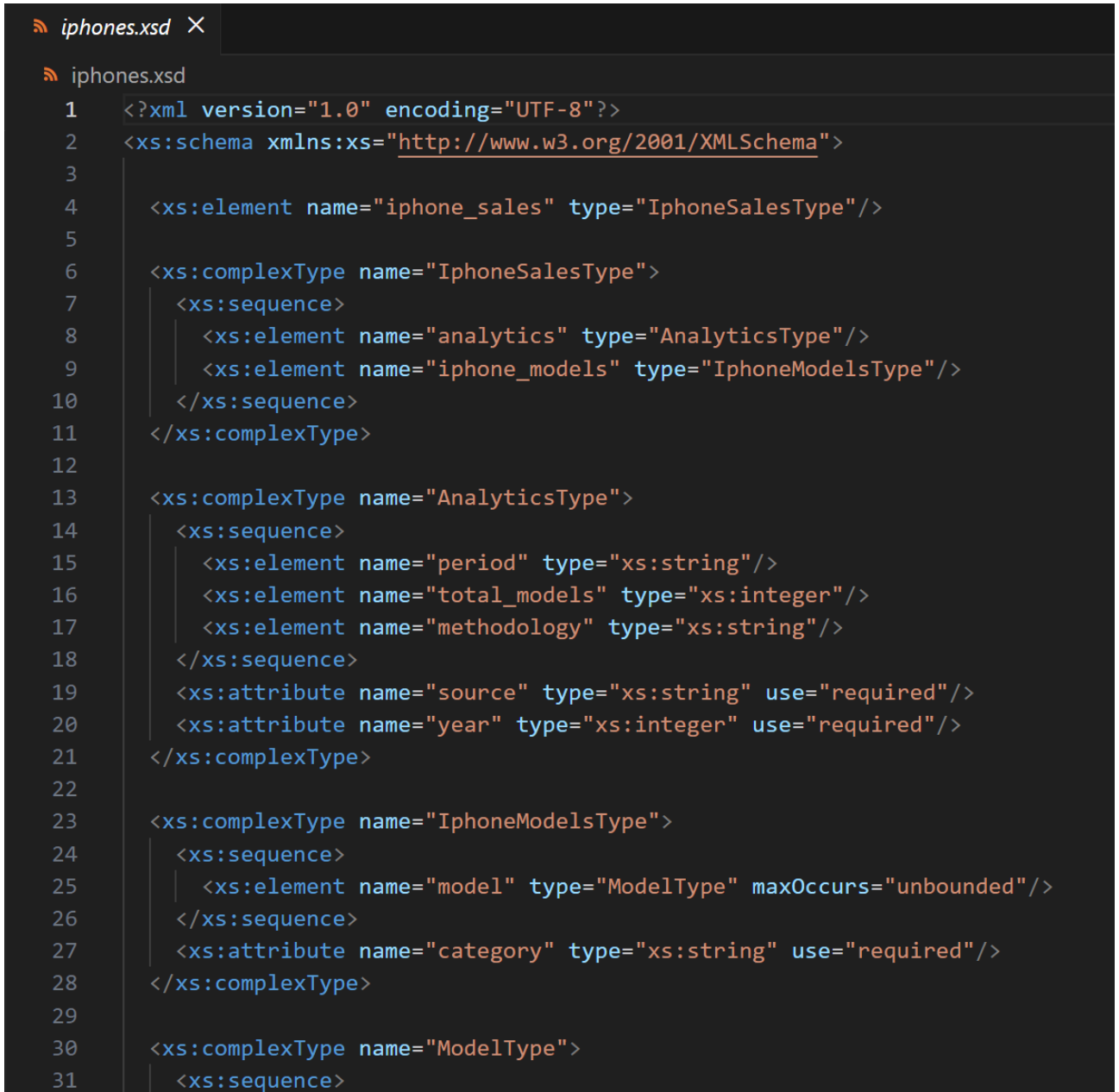
A screenshot of a code editor with a dark theme. The editor has a tab at the top labeled 'iphones.dtd' with a close button. Below the tab, the file name 'iphones.dtd' is shown. The main area contains XML DTD code with line numbers from 1 to 29 on the left. The code defines the structure for an iPhone sales document, including elements for analytics, models, and specifications.

```
1  <!ELEMENT iphone_sales (analytics, iphone_models)>
2
3  <!ELEMENT analytics (period, total_models, methodology)>
4  <!ATTLIST analytics
5  |   source CDATA #REQUIRED
6  |   year CDATA #REQUIRED
7  >
8
9  <!ELEMENT period (#PCDATA)>
10 <!ELEMENT total_models (#PCDATA)>
11 <!ELEMENT methodology (#PCDATA)>
12
13 <!ELEMENT iphone_models (model+)>
14 <!ATTLIST iphone_models
15 |   category CDATA #REQUIRED
16 >
17
18 <!ELEMENT model (specifications)>
19 <!ATTLIST model
20 |   name CDATA #REQUIRED
21 >
22
23 <!ELEMENT specifications (image, release_year, display, sales)>
24
25 <!ELEMENT image (#PCDATA)>
26 <!ELEMENT release_year (#PCDATA)>
27 <!ELEMENT display (#PCDATA)>
28 <!ELEMENT sales (#PCDATA)>
29
```

Рисунок 2 - DTD-документ

## XSD-документ

На рисунке 3 представлен XSD-документ. Код документа приведен в приложении В.

The image shows a code editor window titled 'iphones.xsd'. The code is an XSD schema defining the structure of an iPhone sales document. It starts with an XML declaration for version 1.0 and UTF-8 encoding. The schema uses the xmlns:xs namespace for XML Schema definitions. It defines a root element 'iphone\_sales' of type 'IphoneSalesType'. This type is a complex type containing a sequence of two elements: 'analytics' (of type 'AnalyticsType') and 'iphone\_models' (of type 'IphoneModelsType'). The 'AnalyticsType' is also a complex type with a sequence of three elements: 'period' (string), 'total\_models' (integer), and 'methodology' (string). It also has two required attributes: 'source' (string) and 'year' (integer). The 'IphoneModelsType' is a complex type with a sequence of one or more 'model' elements (of type 'ModelType') and a required attribute 'category' (string). The 'ModelType' is a complex type with a sequence of elements (the sequence is not fully defined in the visible code).


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4   <xs:element name="iphone_sales" type="IphoneSalesType"/>
5
6   <xs:complexType name="IphoneSalesType">
7     <xs:sequence>
8       <xs:element name="analytics" type="AnalyticsType"/>
9       <xs:element name="iphone_models" type="IphoneModelsType"/>
10    </xs:sequence>
11  </xs:complexType>
12
13  <xs:complexType name="AnalyticsType">
14    <xs:sequence>
15      <xs:element name="period" type="xs:string"/>
16      <xs:element name="total_models" type="xs:integer"/>
17      <xs:element name="methodology" type="xs:string"/>
18    </xs:sequence>
19    <xs:attribute name="source" type="xs:string" use="required"/>
20    <xs:attribute name="year" type="xs:integer" use="required"/>
21  </xs:complexType>
22
23  <xs:complexType name="IphoneModelsType">
24    <xs:sequence>
25      <xs:element name="model" type="ModelType" maxOccurs="unbounded"/>
26    </xs:sequence>
27    <xs:attribute name="category" type="xs:string" use="required"/>
28  </xs:complexType>
29
30  <xs:complexType name="ModelType">
31    <xs:sequence>
```

Рисунок 3 - XSD-документ



## XSL-документ

На рисунке 4 представлен отрывок кода XSL-документа. Полный код файла представлен в приложении Г.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3
4   <xsl:output method="html" encoding="UTF-8" indent="yes"/>
5
6   <xsl:template match="/">
7     <html>
8       <head>
9         <meta charset="UTF-8"/>
10        <title>Самые продаваемые iPhone</title>
11        <style>
12          body {
13            background-color: white;
14            font-family: Arial;
15            padding: 20px;
16          }
17          h1 {
18            text-align: center;
19          }
20          .model-cell {
21            background-color: #f0f0f0;
22            padding: 20px;
23            margin: 20px auto;
24            max-width: 500px;
25            border: 1px solid #ddd;
26          }
27          .model-image {
28            width: 100%;
29            height: auto;
30          }
31          p {
32            color: black;
33            font-size: 16px;
34            margin: 10px 0;
```

Рисунок 4 - XSL-документ

## Преобразование XML документа

На рисунке 5 и 6 представлены обрывки веб-сайта, получившегося в результате связывания XML-документа с DTD, XSD, XSL.

## Самые продаваемые iPhone



Рисунок 5-Сайт



iPhone 11

Год: 2019

Экран: 6.1"

Продажи: 170 млн.



Рисунок 6 – Сайт

## **ЗАКЛЮЧЕНИЕ**

В ходе работы был создан XML-документ, а также подготовлены DTD- и XSD-схемы и разработан XSL-файл для преобразования XML в HTML. Для этих файлов была создана отдельная ветка feature2, которая затем была загружена на удалённый репозиторий GitHub с помощью Git Bash и слита в основную ветку main. Для подготовки отчёта дополнительно создана ветка feature3.

Разработка выполнялась в среде Visual Studio Code, а запуск сайта в браузере осуществлялся через расширение Live Server.

Таким образом, выполнение данного задания помогло лучше понять процесс работы с Git и GitHub. Полученные навыки по созданию, ведению и объединению веток особенно важны для командной разработки и служат прочной основой для дальнейшего изучения более сложных техник работы с системами контроля версий.

### **Список использованных источников**

- 1) Самохвалов Э. Н., Ревунков Г. И., Гапанюк Ю. Е.\XML-технологии\Методические указания к лабораторным работам по курсу XML-технологии\ Часть 1 (4 семестр). - Москва: МГТУ им. Н. Э. Баумана, 2012 – 33 с
- 2) Лоу Х.\Популярное введение в современный Git для пользователей и разработчиков. - М.: ДМК Пресс, 2021. - 322 с.

## Приложение А

Код XML-документа

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="iphones.xsl"?>
<iphone_sales xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="iphones.xsd">

    <analytics source="Statista" year="2024">
        <period>2014-2024</period>
        <total_models>5</total_models>
        <methodology>Анализ данных международных аналитических
агентств</methodology>
    </analytics>

    <iphone_models category="top_sellers">

        <model name="iPhone 6">
            <specifications>
                <image>image/6 iphone.jpg</image>
                <release_year>2014</release_year>
                <display>4.7"</display>
                <sales>220</sales>
            </specifications>
        </model>

        <model name="iPhone 11">
            <specifications>
                <image>image/11 iphone.jpg</image>
                <release_year>2019</release_year>
                <display>6.1"</display>
```

```
<sales>170</sales>
</specifications>
</model>

<model name="iPhone 12">
  <specifications>
    <image>image/12 iphone.jpg</image>
    <release_year>2020</release_year>
    <display>6.1"</display>
    <sales>150</sales>
  </specifications>
</model>

<model name="iPhone 13">
  <specifications>
    <image>image/13 iphone .jpg</image>
    <release_year>2021</release_year>
    <display>6.1"</display>
    <sales>140</sales>
  </specifications>
</model>

<model name="iPhone 15">
  <specifications>
    <image>image/15 iphone.jpg</image>
    <release_year>2023</release_year>
    <display>6.1"</display>
    <sales>80</sales>
  </specifications>
</model>
```

</iphone\_models>

</iphone\_sales>



## Приложение Б

Код DTD-документа

```
<!ELEMENT iphone_sales (analytics, iphone_models)>
```

```
<!ELEMENT analytics (period, total_models, methodology)>
```

```
<!ATTLIST analytics
```

```
    source CDATA #REQUIRED
```

```
    year CDATA #REQUIRED
```

```
>
```

```
<!ELEMENT period (#PCDATA)>
```

```
<!ELEMENT total_models (#PCDATA)>
```

```
<!ELEMENT methodology (#PCDATA)>
```

```
<!ELEMENT iphone_models (model+)>
```

```
<!ATTLIST iphone_models
```

```
    category CDATA #REQUIRED
```

```
>
```

```
<!ELEMENT model (specifications)>
```

```
<!ATTLIST model
```

```
    name CDATA #REQUIRED
```

```
>
```

```
<!ELEMENT specifications (image, release_year, display, sales)>
```

```
<!ELEMENT image (#PCDATA)>
```

```
<!ELEMENT release_year (#PCDATA)>
```

```
<!ELEMENT display (#PCDATA)>
```

```
<!ELEMENT sales (#PCDATA)>
```

## Приложение В

### Код XSD-документа

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="iphone_sales" type="IphoneSalesType"/>

  <xs:complexType name="IphoneSalesType">
    <xs:sequence>
      <xs:element name="analytics" type="AnalyticsType"/>
      <xs:element name="iphone_models" type="IphoneModelsType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AnalyticsType">
    <xs:sequence>
      <xs:element name="period" type="xs:string"/>
      <xs:element name="total_models" type="xs:integer"/>
      <xs:element name="methodology" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="source" type="xs:string" use="required"/>
    <xs:attribute name="year" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="IphoneModelsType">
    <xs:sequence>
      <xs:element name="model" type="ModelType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="category" type="xs:string" use="required"/>
  </xs:complexType>
```

```
<xs:complexType name="ModelType">
  <xs:sequence>
    <xs:element name="specifications" type="SpecificationsType"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

```
<xs:complexType name="SpecificationsType">
  <xs:sequence>
    <xs:element name="image" type="xs:string"/>
    <xs:element name="release_year" type="xs:integer"/>
    <xs:element name="display" type="xs:string"/>
    <xs:element name="sales" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:schema>
```

## Приложение Г

Код XSL-документа

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="html" encoding="UTF-8" indent="yes"/>

  <xsl:template match="/">
    <html>
      <head>
        <meta charset="UTF-8"/>
        <title>Самые продаваемые iPhone</title>
        <style>
          body {
            background-color: white;
            font-family: Arial;
            padding: 20px;
          }
          h1 {
            text-align: center;
          }
          .model-cell {
            background-color: #f0f0f0;
            padding: 20px;
            margin: 20px auto;
            max-width: 500px;
            border: 1px solid #ddd;
          }
          .model-image {
```

```

width: 100%;
height: auto;
}
p {
color: black;
font-size: 16px;
margin: 10px 0;
}
</style>
</head>
<body>
<h1>Самые продаваемые iPhone</h1>

<xsl:for-each select="/iphone_sales/iphone_models/model">
<div class="model-cell">
<img class="model-image">
<xsl:attribute name="src">
<xsl:value-of select="specifications/image"/>
</xsl:attribute>
<xsl:attribute name="alt">
<xsl:value-of select="@name"/>
</xsl:attribute>
</img>
<p><xsl:value-of select="@name"/></p>
<p>Год: <xsl:value-of select="specifications/release_year"/></p>
<p>Экран: <xsl:value-of select="specifications/display"/></p>
<p>Продажи: <xsl:value-of select="specifications/sales"/> млн.</p>
</div>
</xsl:for-each>

```

</body>

</html>

</xsl:template>

</xsl:stylesheet>