

№ 4. Использование сортировки с условием

Сортировка задается с помощью элементов `xsl:sort`, данных как непосредственный потомок элемента `xsl:apply-templates` ИЛИ `xsl:for-each`.

Атрибуты:

– `select` – обязательный атрибут, значением которого является выражение, называемое также ключевым выражением. Это выражение вычисляется для каждого узла обрабатываемого множества, преобразуется в строку и затем используется как значение ключа при сортировке. По умолчанию значением этого атрибута является ".", что означает, что в качестве значения ключа для каждого узла используется его строковое значение;

– `order` – необязательный атрибут, определяет порядок, в котором узлы должны сортироваться по своим ключам. Этот атрибут может принимать только два значения – "ascending", указывающее на восходящий порядок сортировки, и "descending", указывающее на нисходящий порядок. Значением по умолчанию является "ascending", то есть восходящий порядок;

– `lang` – необязательный атрибут, определяет язык ключей сортировки. Дело в том, что в разных языках символы алфавита могут иметь различный порядок, что, соответственно, должно учитываться при сортировке. Атрибут `lang` в XSLT может иметь те же самые значения, что и атрибут `xml:lang` (например: "en", "en-us", "ru" и т. д.). Если значение этого атрибута не определено, процессор может либо определять язык исходя из параметров системы, либо сортировать строки исходя из порядка кодов символов Unicode;

– `data-type` – необязательный атрибут, определяет тип данных, который несут строковые значения ключей.

К сожалению, сортировкой нельзя управлять динамически. Все атрибуты элемента `xsl:sort` должны обладать фиксированными значениями;

При обработке `xsl:choose` процессор поочередно вычисляет выражения, содержащиеся в атрибутах `test` элементов `xsl:when`, приводит их к булевому типу и выполняет содержимое первого (и только первого) элемента, тестовое выражение которого будет равно `true`. В случае если ни одно из тестовых выражений не обратилось в "истину" и в `xsl:choose` присутствует `xsl:otherwise`, процессор выполнит содержимое этого элемента. `xsl:choose` состоит из последовательности элементов `xsl:when`, за которой следует необязательный элемент `xsl:otherwise`. Каждый элемент `xsl:when` имеет единственный атрибут `test`, который задает некое выражение.

```
<xsl:choose>
  <xsl:when test="условие1">
    шаблон1
  </xsl:when>
  <xsl:when test="условие2">
    шаблон2
  </xsl:when>
```

```
<!-- ... -->
<xsl:when test="условиеN">
    шаблонN
</xsl:when>
<xsl:otherwise>
    шаблонM
</xsl:otherwise>
</xsl:choose>
```

Задание к лабораторной работе № 4

Задание 1. Создайте XML-документ, в котором будет расположена информация о ваших одногруппниках. Например: фамилия, имя, отчество, проходной балл, год рождения. Произведите сортировку по году рождения, выделите красным цветом имена тех одногруппников, которые имеют средний балл меньше 6, и зеленым тех, которые имеют средний балл равный 6.