

Мы увидели, как сделать XSLT более похожим на программирование. Как насчет добавления к нему фактического программирования? Давайте взглянем на добавление функциональности Java к таблице стилей XSLT.

Во-первых, следует отметить, что хотя механизм расширения является частью рекомендаций XSLT, рассматриваемая здесь реализация специфична для процессора Xalan XSLT. Основные идеи практически одинаковы для других процессоров, однако для уточнения деталей вам придется сверяться с документацией.

Далее мы создадим расширение, которое позволит нам масштабировать рецепт на несколько порций.

### Элементы расширения

Расширение XSLT производится с помощью различных методик. Первая - это использование элементов `extension`. Элемент `extension` – это элемент в пространстве имен, который указывает на класс Java. Взгляните, например, на следующий элемент `extension` (см. листинг 30).

### Листинг 30. Использование элемента `extension`

```
<xsl:stylesheet version="1.0"
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

```
xmlns="http://www.w3.org/TR/xhtml1/strict"
```

```
xmlns:scaler = "com.backstop.RecipeScaler"
```

```
extension-element-prefixes="scaler">
```

```
<xsl:template match="/">
```

```
<html>
```

```
<head>
```

```
<title>Recipes</title>
```

```
</head>
```

```
<body>
```

```
<scaler:scaleMessage servings="2" />
```

```
<xsl:apply-templates select="/recipes/recipe"/>
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

```
...
```

Мы создали пространство имен, которое соответствует классу `com.backstop.RecipeScaler`, который включает в себя статический метод под названием `scaleMessage` (см. листинг 31).

### Листинг 31. Класс `RecipeScaler`

```
package com.backstop;
```

```
public class RecipeScaler {
```

```
public static String scaleMessage (
```

```
org.apache.xalan.extensions.XSLProcessorContext context,
```

```
org.w3c.dom.Element thisElement){
```

```
return "This recipe has been scaled by a factor of " +
```

```
thisElement.getAttribute("servings") + ".";
```

}

}

Добравшись до элемента, процессор видит префикс пространства имен `scaler:` и знает, что он обозначен как префикс элемента `extension`, и таким образом понимает, какой класс обозначен в определении пространства имен. Вызываемый им метод отвечает локальному имени элемента - `scaleMessage`. Сам метод получает два аргумента, из которых мы фактически используем один. Параметр `context` ссылается на контекст процессора, который позволяет взглянуть на элементы, относящиеся к элементу `extension`, однако мы просто займемся самим элементом `extension`. Так как мы получаем этот элемент как параметр метода, мы можем извлечь значения любых атрибутов, добавленных к этому элементу, таких как `servings` в данном случае. Текст, возвращенный методом, добавляется к выводу на месте элемента `extension`.

Это означает, что если применить таблицу стилей, мы получим результаты, показанные на рисунке 10.

**Рисунок 10. Результаты элемента `extension`**

```
tail version: 1.0 encoding: UTF-8 ?xml xmlns: "http://www.w3.org/TR/xhtml1/2
trict">head<title>Recipe</title></head>body</body>This recipe has been scaled by a
factor of 2. <h2>Cook gosh</h2><h3>Ingredients</h3><p>
1 pound hamburger</p>
1 pound elbow macaroni</p>
2 cups brown sugar</p>
1 bag chopped onion</p>
1 teaspoon dried dill</p>
</p></h3><directions></h3><ol>
<li>Brown the hamburger.</li>
<li>Add onions and cook until transparent.</li>
<li>Add brown sugar and dill.</li>
<li>Cook and drain pasta.</li>
<li>Combine meat and pasta.</li>
</ol><h2>B balanced breakfast</h2><h3>Ingredients</h3><p>
1 cup cereal</p>
1 glass orange juice</p>
2 slices toast</p>
</p></h3><directions></h3><ol>
<li>Combine cereal and milk in bowl.</li>
<li>Add all ingredients to table.</li>
</ol></body></html>
```

```
tail version: 1.0 encoding: UTF-8 ?xml xmlns: "http://www.w3.org/TR/xhtml1/2
trict">head<title>Recipe</title></head>body</body>This recipe has been scaled by a
factor of 2. <h2>Cook gosh</h2><h3>Ingredients</h3><p>
2 pound hamburger</p>
2 pound elbow macaroni</p>
4 cups brown sugar</p>
2 bag chopped onion</p>
2 teaspoon dried dill</p>
</p></h3><directions></h3><ol>
<li>Brown the hamburger.</li>
<li>Add onions and cook until transparent.</li>
<li>Add brown sugar and dill.</li>
<li>Cook and drain pasta.</li>
<li>Combine meat and pasta.</li>
</ol><h2>B balanced breakfast</h2><h3>Ingredients</h3><p>
2 cup cereal</p>
2 glass orange juice</p>
4 slices toast</p>
</p></h3><directions></h3><ol>
<li>Combine cereal and milk in bowl.</li>
<li>Add all ingredients to table.</li>
</ol></body></html>
```

Важно быть осторожным, чтобы не переопределить существующие элементы. В

Примере ниже мы использовали элемент `<ol>` для того, чтобы добавить рецепты, как