

Standard Dialect.

Синтаксис.

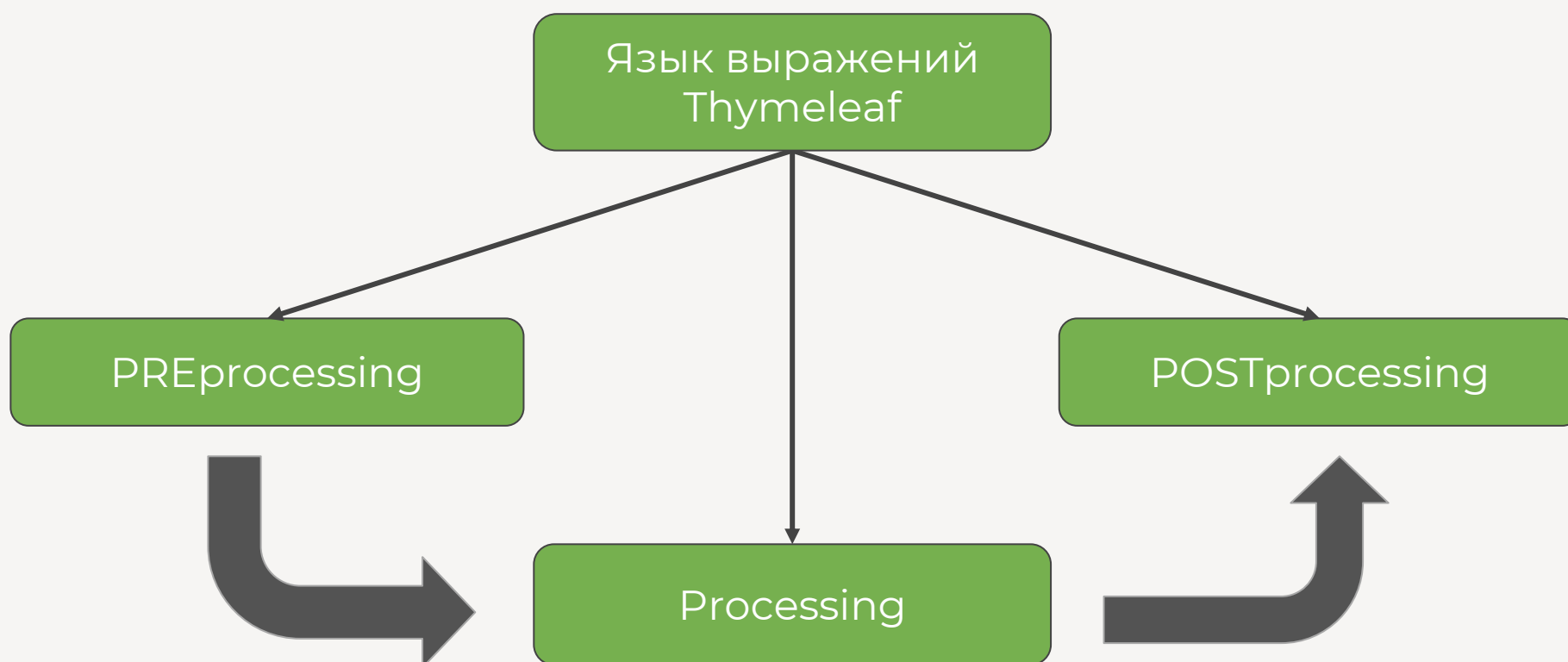
Операторы

Кирилл Кошаев

Java-разработчик в «Газпром информ»

Язык выражений Thymeleaf

Представляет собой набор функций, условно разделяемых на 3 категории



Standard Dialect

Thymeleaf атрибуты начинаются с сочетания `th:`, которое называется префиксом диалекта, и это означает, что все теги, атрибуты и, соответственно, выражения, обрабатываемые этим диалектом, будут начинаться с такого префикса

```
<div class="Slider-box Cards">
  <div class="Slider-item" th:each="book : ${bookData}">
    <div class="Slider-content">
      <div class="Card"><a class="Card-picture" href="#"></a>
      <div class="Card-content">
        <strong class="Card-title"><a href="#" th:text="${book.title}"></a>
        </strong>
        <div class="Card-description" th:text="${book.author}">
        </div>
        <div class="Card-cost"><span class="Card-priceOld"
th:text="${book.priceOld}"></span>
          <span class="Card-price" th:text="${book.price}"></span>
        </div>
      </div>
    </div>
  </div>
</div>
```

thymeleaf атрибут

thymeleaf-префикс

thymeleaf-префикс

thymeleaf атрибут

Простые выражения

- Variable Expressions: `${...}`
- Selection Variable Expressions: `*{...}`
- Message Expressions: `#{...}`
- Link URL Expressions: `@{...}`
- Fragment Expressions: `~{...}`

Standard Dialect

Помимо приведённых выше конструкций, мы можем пользоваться в выражениях основными литералами, типичными для большинства языков программирования

- Текст: 'one text', 'Another one!',...
- Числа: 0, 34, 3.0, 12.3,...
- Логические литералы: true, false
- Null : null

Messages #{. . .}

Message Expressions дают возможность связать контент, отображаемый в разметке шаблона, со свойствами, которые мы можем определить в property файлах

```
some.property file ->
```

```
welcome.message=What a lovely day today: {0} !
```

```
markup ->
```

```
<p th:utext="#{welcome.message}">What a lovely day today: </p>
```

```
markup ->
```

```
<p th:text="#{welcome.message (${today})}">
```

```
    What a lovely day today: 14 march 1992 !
```

```
</p>
```

Variables `${. . .}`

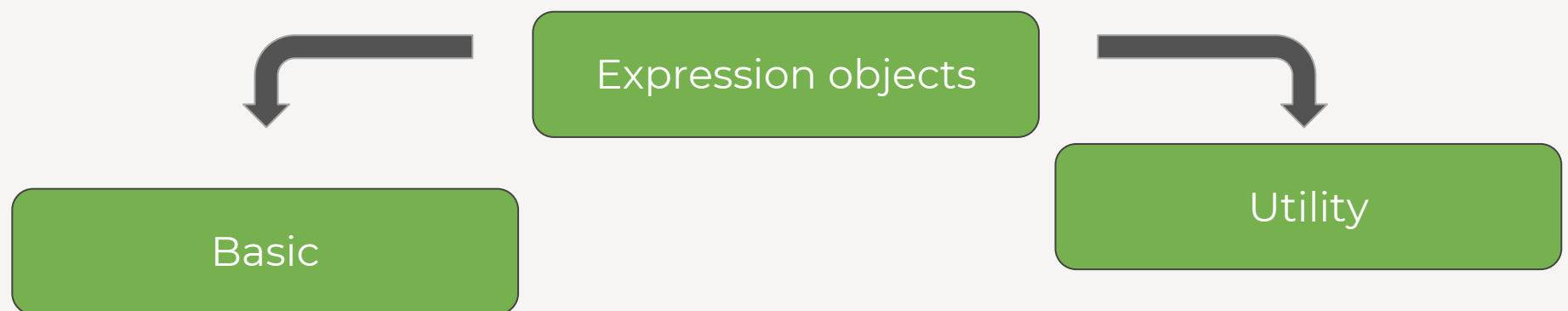
В качестве значения выражения переменной мы можем указывать текст, цифры и другие литералы, а также объекты.

Для того, чтобы иметь доступ к свойствам переданного объекта, мы можем воспользоваться следующими вариантами

```
//Доступ к свойствам при помощи dot-notation.  
${person.father.name}  
//Доступ к свойствам при помощи скобок ([])  
${person['father']['name']}  
//Если объект является Map, можно смешивать dot и скобки.  
${countriesByCode.ES}  
${personsByName['Stephen Zucchini'].age}  
//Доступ по индексу и dot notation  
${personsArray[0].name}  
//Вызов метода с и без аргументов  
${person.createCompleteName()}  
${person.createCompleteNameWithSeparator('-')}
```

Expression objects

Для придания гибкости при составлении выражений в Thymeleaf используется большое количество Expression-объектов, доступ к которым осуществляется с помощью символа #



#ctx
#vars
#locale
#request
#response
#session
#servletContext

Expression objects

К служебным, т. е. к expression utility объектам, относятся те объекты, которые представляют различные вспомогательные методы



**Спасибо
за внимание!**