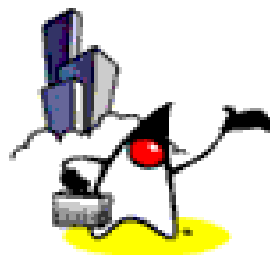




# Обзор JavaServer Faces (JSF)

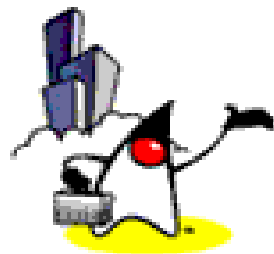


# Источники информации

- Спецификация
  - JavaServer™ Faces Specification Version 1.2
- Электронный курс “Java™ EE (J2EE) Programming (with Passion!)”  
([www.javapassion.com/j2ee](http://www.javapassion.com/j2ee))
  - Особая благодарность автору Sang Shin,  
[sang.shin@sun.com](mailto:sang.shin@sun.com)



# Архитектура веб-приложений



# Эволюция архитектуры MVC

1. Без MVC

2. MVC Model 1 (Page-centric)

3. MVC Model 2 (Servlet-centric)

4. Каркасы веб-приложений

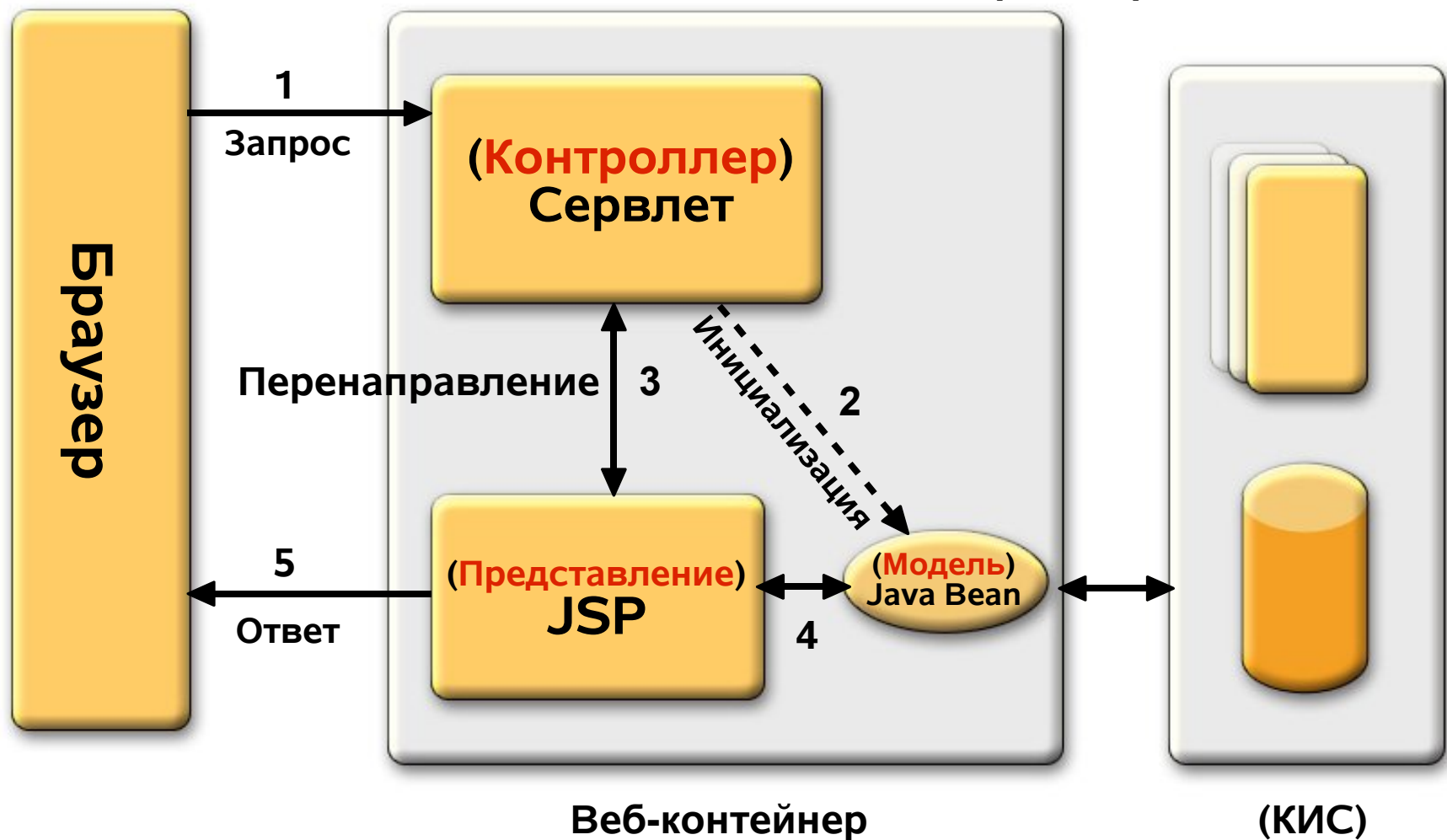
- Struts, Tapestry, ...

5. Стандартный каркас веб-приложений

- JavaServer Faces (JSR-127)

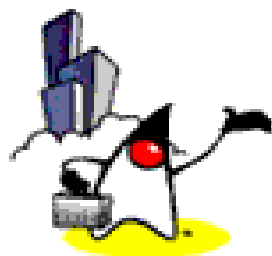
# Архитектура Model 2 (на базе сервлета)

Шаблон проектирования MVC





# Что такое JSF?



# Каркас JavaServer™ Faces (JSF) это ...

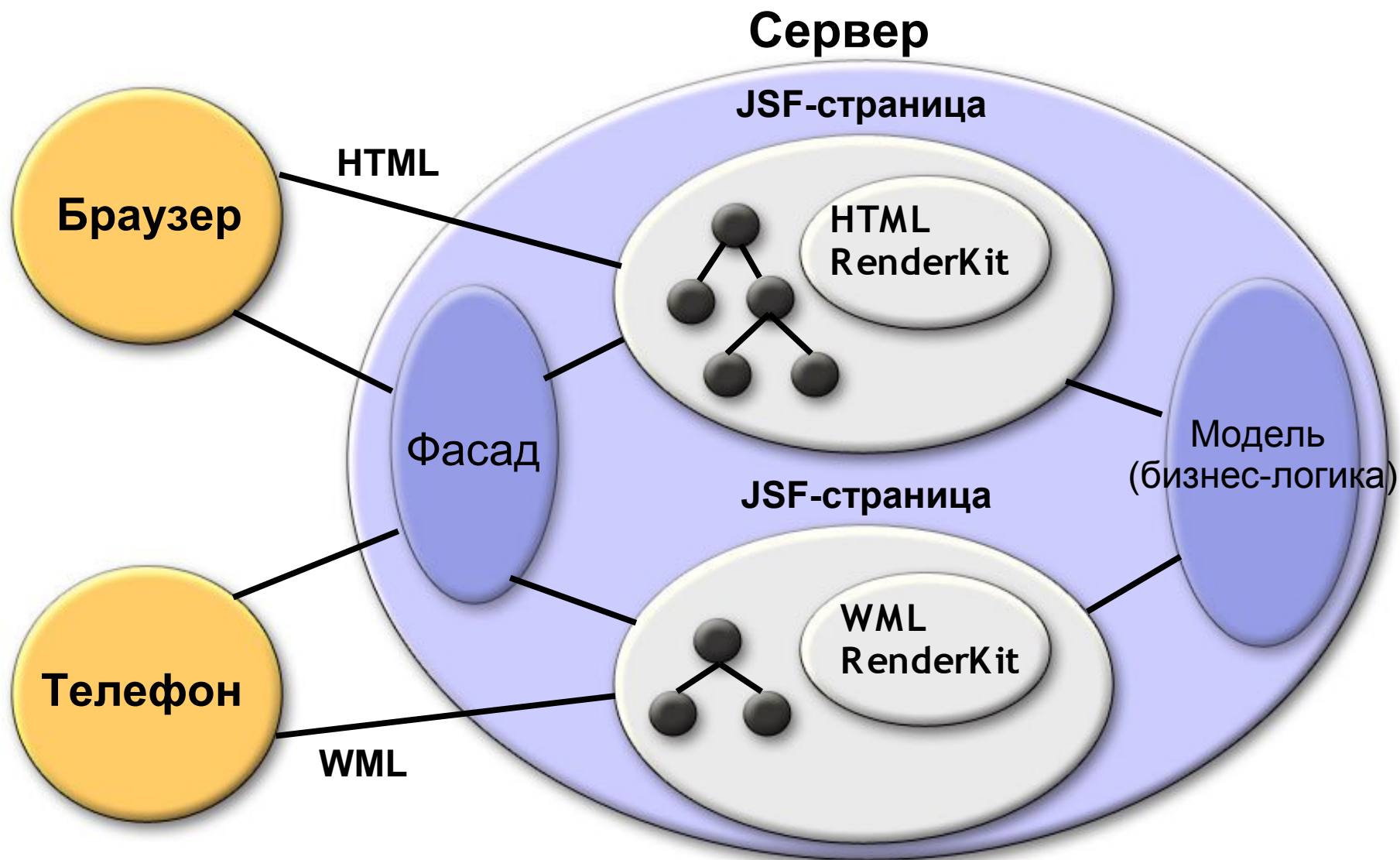
Каркас **серверных компонентов**  
**пользовательского интерфейса**  
для веб-приложений,  
построенных на технологии  
Java™.

# Что такое JSF?

- Каркас веб-приложений, построенный на компонентной модели
  - Компоненты интерфейса пользователя
  - Обработка событий
  - Интеграция с данными КИС
- Спроектирован для использования в средах разработки
  - Пример: NetBeans Visual Web Pack (VWP)
  - JSF-приложения можно разрабатывать и вручную



# Архитектура JSF

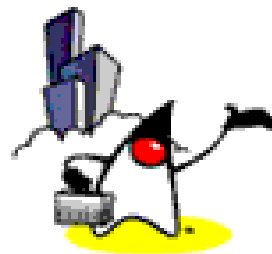


# Почему JSF?

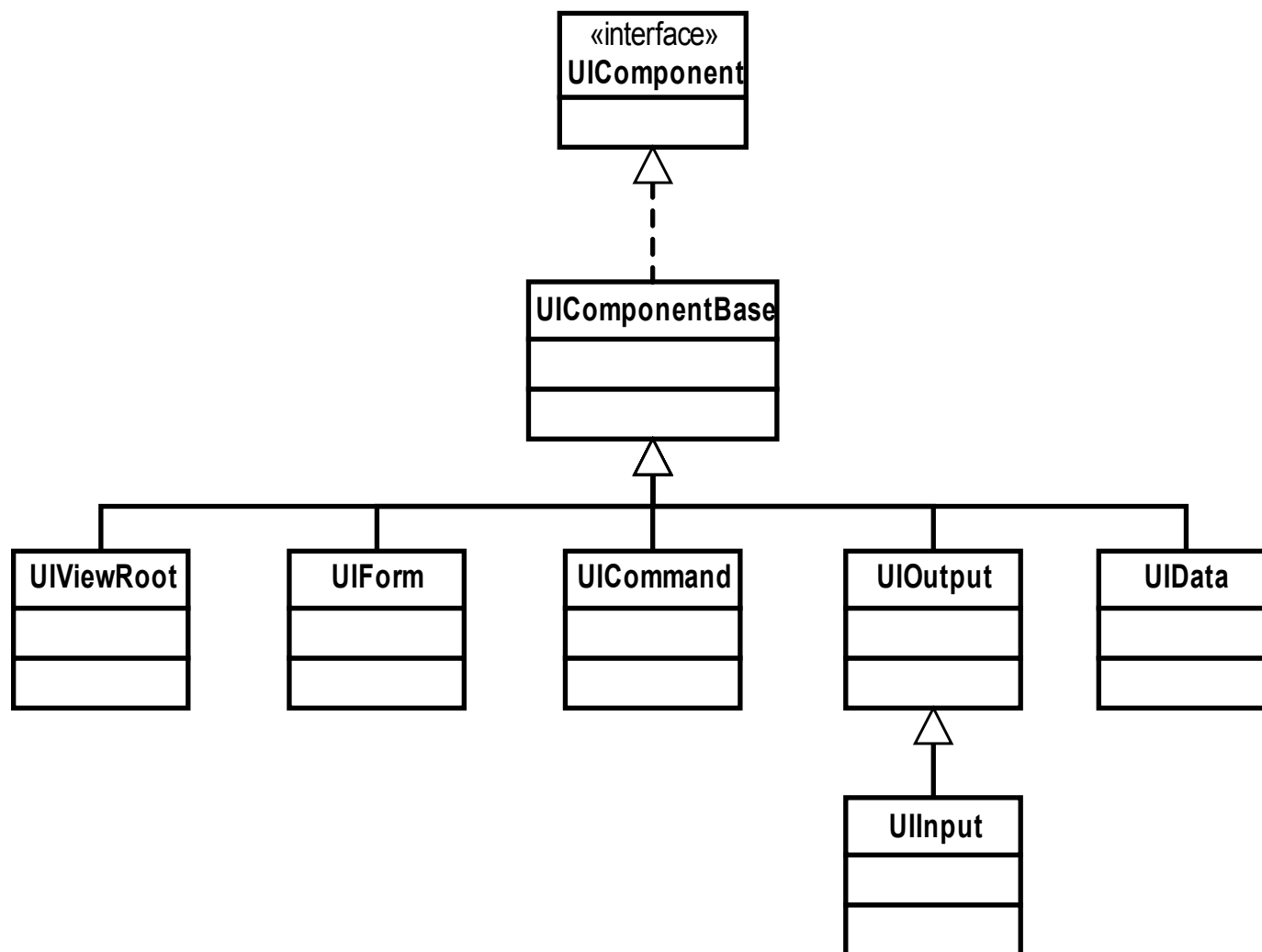
- Более высокий уровень абстракции для разработки веб-приложений
  - Событийно-ориентированная модель программирования (в отличие от модели запрос/ответ протокола HTTP)
- Шаблон MVC для веб-приложений
- Расширяемая архитектура компонентов интерфейса и их отображения
  - Независимость от клиентских устройств
- Промышленный стандарт
- Поддержка производителей



# Основные концепции JSF

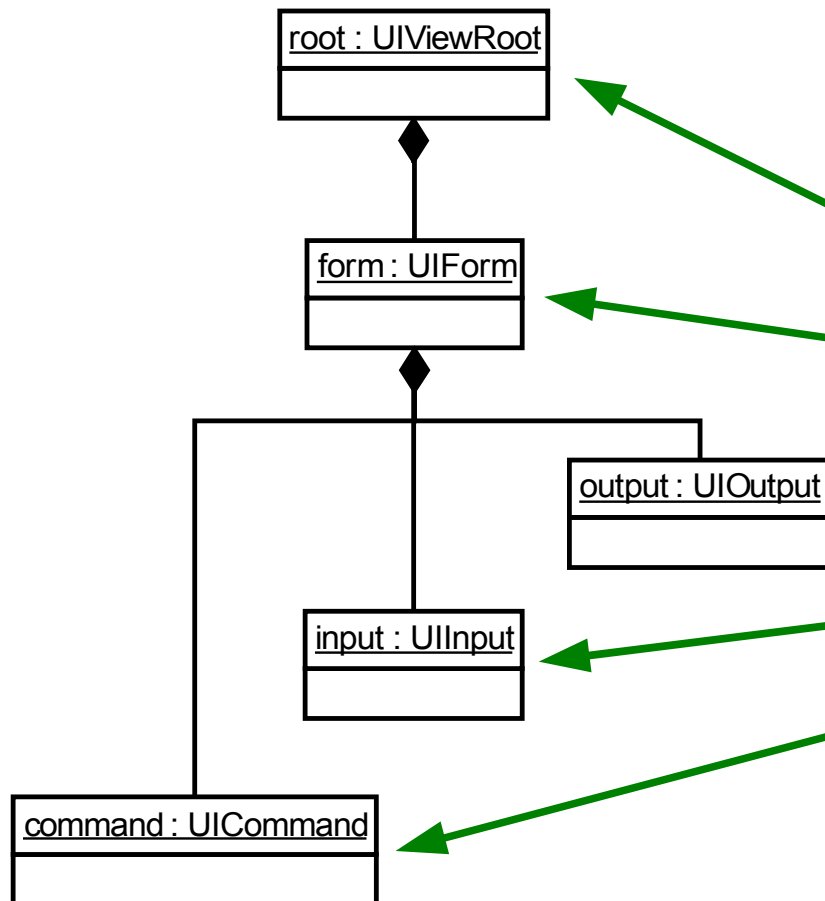


# Представление - Базовые компоненты



# Представление - Компоненты и рендереры

Дерево компонентов



JSP-страница `guess.jsp`  
с использованием рендереров  
из Basic HTML RenderKit

```
<html>
  <head>
    <meta http-equiv="Content-Type" con
    <title>Guess Number</title>
  </head>
  <body>
    <f:view>
      <h:form>
        <h1>Guess Number</h1>
        Attempt #<h:outputText/>.
        Enter integer from 0 to 9:
        <h:inputText/>
        <h:commandButton/>
      </h:form>
    </f:view>
  </body>
</html>
```

# Модель – Managed Beans

- Объекты на стороне сервера, связанные с приложением, сессией или отдельным запросом
- Содержат свойства, связанные с
  - значением компонента или
  - экземпляром компонента
- Могут содержать методы, выполняющие обработку событий и валидацию
- Для обращения к свойствам и методам используются EL-выражения с отложенным выполнением
  - `#{managedBean.property}`

# Модель – Определение Managed Bean

## Класс Managed Bean

```
public class GuessNumberBean {  
  
    private int number;  
    private int guess;  
    private int guessCount;  
    private boolean numberIsSet;  
  
    public int getNumber() {...}  
  
    public int getGuess() {...}  
  
    public void setGuess(int guess) {...}  
  
    public int getGuessCount() {...}  
  
    public String makeGuess() {...}  
  
    public String reset() {...}  
  
}
```

## Фрагмент faces-config.xml

```
<managed-bean>  
    <managed-bean-name>  
        guessBean  
    </managed-bean-name>  
    <managed-bean-class>  
        guessnumber.GuessNumberBean  
    </managed-bean-class>  
    <managed-bean-scope>  
        session  
    </managed-bean-scope>  
    <managed-property>  
        <property-name>  
            guessCount  
        </property-name>  
        <value>1</value>  
    </managed-property>  
</managed-bean>
```

# Модель – Связь с представлением

## JSP-страница guess.jsp

```
<f:view>
  <h:form>
    <h1>Guess Number</h1>
    <h:messages/>
    Attempt #<h:outputText value="#{guessBean.guessCount}"/>.
    Enter integer from 0 to 9:
    <h:inputText value="#{guessBean.guess}"/>
    <h:commandButton value="Submit" action="#{guessBean.makeGuess}"/>
  </h:form>
</f:view>
```

Отображение  
свойства

Отображение  
и изменение свойства

Метод обработки события



# Модель – Валидация и конвертация

- Валидация – проверка правильности вводимых данных
  - **Validator** – базовый класс для стандартных и определяемых в приложении валидаторов
  - Методы валидации в Managed Beans
  - Пример использования стандартного валидатора:

```
<h:inputText value="#{guessBean.guess}">  
    <f:validateLongRange minimum="0" maximum="9"/>  
</h:inputText>
```
- Конвертация – преобразование из строки в объект и наоборот
  - **Converter** – базовый класс для конвертеров, определяемых в приложении

# Контроллер – Навигация по страницам

## Фрагмент guess.jsp

```
<f:view>
  <h:form>
    <h1>Guess Number</h1>
    <h:messages/>
    Attempt #
    <h:outputText value="#{guessBean.guessCount}"/>.
    Enter integer from 0 to 9:
    <h:inputText value="#{guessBean.guess}">
      <f:validateLongRange minimum="0" maximum="9"/>
    </h:inputText>
    <h:commandButton value="Submit"
      action="#{guessBean.makeGuess}"/>
  </h:form>
</f:view>
```

## Фрагмент success.jsp

```
<f:view>
  <h:form>
    <h1>Congratulations!</h1>
    You guessed my number
    <b><h:outputText value="#{guessBean.number}"/></b>
    in attempt
    #<h:outputText value="#{guessBean.guessCount}"/>!
  </h:form>
</f:view>
```

## Фрагмент

### GuessNumberBean.java

```
public class GuessNumberBean {

    public String makeGuess() {
        if (guess == number) {
            return "success";
        } else {
            guessCount++;
            return "failure";
        }
    }
}
```

## Фрагмент faces-config.xml

```
<navigation-rule>
  <from-view-id>/guess.jsp</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/success.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>failure</from-outcome>
    <to-view-id>/guess.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

1

2

3

# Контроллер – Контекст запроса FacesContext

- Содержит деревья JSF-компонентов для запроса и ответа.
- Обеспечивает доступ к запросу, ответу, сессии и контексту веб-приложения
- Выполняет вычисление отложенных EL-выражений

# Файл конфигурации приложения

- XML-файл для конфигурирования ресурсов, необходимых при запуске приложения
  - Правила навигации, managed beans, конвертеры, валидаторы, наборы рендереров
- Обычно называется **faces-config.xml**
- Корневой элемент - `<faces-config>`:

