

## Лабораторная работа

# Основы разработки веб-приложений с использованием каркаса JSF 2.0

### Цель

Изучить особенности каркаса JavaServer Faces 2.0 для разработки веб-приложений (Facelets, аннотации, поддержка AJAX) и получить навыки реализации веб-приложений с использованием данного каркаса.

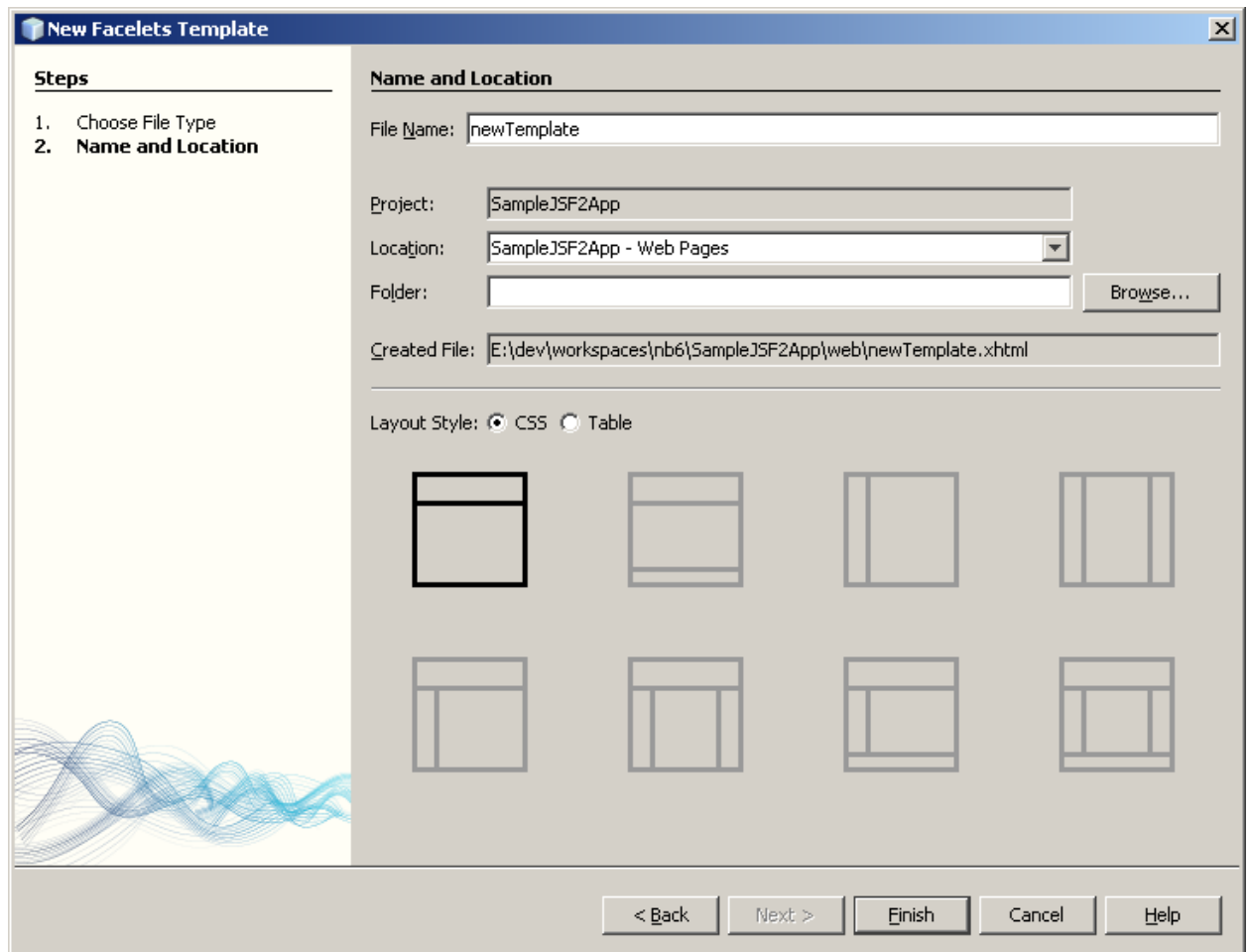
### Задание

1. Разработать веб-приложение, состоящее из двух страниц: одна страница должна обеспечивать ввод данных, а вторая – отображение результатов их обработки и возврат на первую страницу.
2. Страницы должны быть разработаны на языке XHTML с использованием технологии Facelets. Страницы приложения должны использовать общий шаблон. Для связи страниц должна использоваться неявная навигация (implicit navigation).
3. Для сбора и обработки данных должны использоваться управляемые бины контекста запроса и контекста сессии. Управляемые бины должны определяться с использованием аннотаций, содержать управляемые свойства и ограничения, описанные с помощью аннотаций Bean Validation API.
4. В веб-интерфейсе приложения должен быть реализован асинхронный запрос к серверу для получения данных без повторной загрузки текущей страницы (поддержка AJAX) средствами JSF 2.0.
5. В приложении должен быть реализован компонент интерфейс пользователя с помощью технологии Facelets. У компонента должен быть по крайней мере один атрибут.

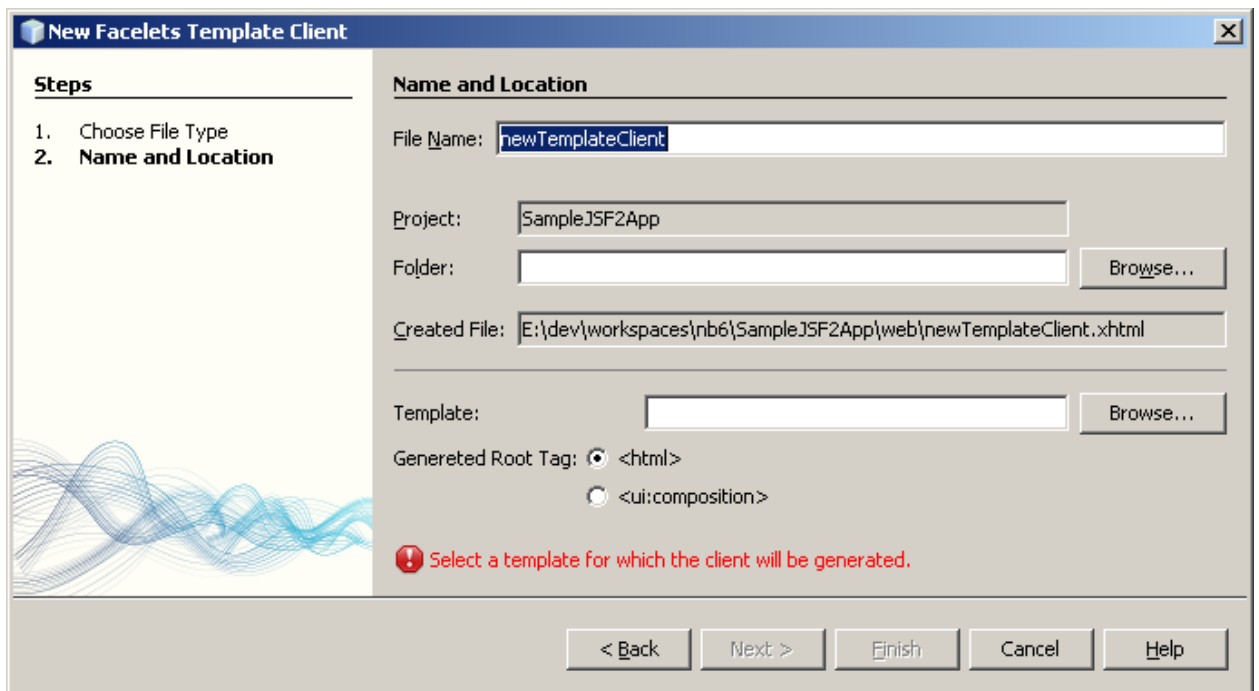
### Порядок выполнения работы

1. Создайте в среде разработки NetBeans новый проект типа **Web / Web Application**. Введите в поле **Project Name** название проекта. Нажмите кнопку **Next**. В списке Server выберите сервер приложений, поддерживающий Java EE 6, например, **GlassFish v3**. В списке Java EE Version выберите пункт **Java EE 6 Web**. Нажмите кнопку **Next**. Установите флажок для каркаса JavaServer Faces. Убедитесь, что на появившейся внизу вкладке Libraries выбран вариант **Use default library which comes with Server (JSF 2.0)**. На вкладке Configuration убедитесь, что в списке Preferred Page Language (предпочтительный язык для JSF-страниц) выбран пункт Facelets. Нажмите кнопку **Finish**.
2. Создайте классы управляемых бинов (файлы типа **JavaServer Faces / JSF Managed Bean**). Укажите имя класса в поле Class Name, пакет в поле Package, выберите в списке Scope требуемый контекст. Изучите аннотации, сгенерированные для определения управляемого бина. Добавьте в классы управляемых бинов необходимые свойства (поля и соответствующие get/set-методы) и action-методы, которые будут вызываться для обработки действия пользователя. Добавьте инициализацию свойств для управляемых бинов с помощью аннотации @ManagedProperty. Добавьте аннотации из Bean Validation API, устанавливающие ограничения на значения свойств управляемых бинов (данные аннотации определены в пакете javax.validation.constraints).

3. Создайте шаблон для Facelets (файл типа **JavaServer Faces / Facelets Template**). Укажите имя файла в поле File Name, выберите способ разметки в поле Layout Style – CSS (разделы со стилями из таблицы стилей) либо Table (обычные HTML-таблицы), а под ним выберите собственно разметку, которую будет задавать шаблон. Нажмите кнопку **Finish**. Изучите исходный код шаблона.



4. Создайте Facelets-страницы, которые используют созданный ранее шаблон (файлы типа **JavaServer Faces / Facelets Template Client**). Укажите имя файла в поле File Name, выберите созданный ранее шаблон в поле Template с помощью кнопки Browse. Нажмите кнопку **Finish**. Изучите исходный код Facelets-страниц.



5. Добавьте на Facelets-страницы форму, компоненты интерфейса пользователя, свяжите их со свойствами и методами управляемых бинов.
6. Добавьте для полей ввода данных действия валидации, например, `<f:validateBean>` для валидации по ограничениям, определенным с помощью аннотаций Bean Validation API.
7. Запустите веб-приложение, выбрав в контекстном меню проекта пункт **Run Project**. Убедитесь в правильном функционировании созданных страниц. Выполните несколько запросов, в том числе с неправильными данными. Просмотрите в окне HTTP Monitor обработанные запросы.
8. Для реализации AJAX-запроса используйте действие `<f:ajax>`. Определите, значения каких полей должны передавать на сервер и обратно при выполнении AJAX-запроса и укажите их идентификаторы в атрибутах `execute` и `render` действия `<f:ajax>`. Запустите веб-приложение и продемонстрируйте асинхронное выполнение запроса без повторной загрузки страницы.
9. Создайте новый компонент интерфейса пользователя на базе полей ввода и кнопок страницы ввода данных. Для этого создайте файл типа **JavaServer Faces / JSF Composite Component**. Укажите имя файла в поле File Name (оно будет использоваться в качестве имени действия). Введите название новой папки или выберите существующую папку в поле Folder. **Внимание:** папка обязательно должна быть вложена в каталог `resources`. Введите префикс библиотеки действий в поле Prefix. Нажмите кнопку **Finish**. Если шаг выполнен успешно, то перейдите к п.12.

**New JSF Composite Component**

**Steps**

1. Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Prefix:

The library prefix cannot be empty.

< Back   Next >   Finish   Cancel   Help

10. Если мастер создания составного компонента работает с ошибкой (пример приведен выше), то создайте обычную JSF-страницу (файл типа **JavaServer Faces / JSF Page**), использующую технологию Facelets. Остальные параметры должны быть заполнены также, как в мастере создания составного компонента.

**New JSF File**

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ Facelets

☐ JSP File (Standard Syntax)   ☐ Create as a JSP Segment

Description:

< Back   Next >   Finish   Cancel   Help

11. Добавьте в Facelets-страницу объявление пространства имен для составных компонентов:
- ```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:cc="http://java.sun.com/jsf/composite">
```
12. Определите интерфейс компонента внутри действия `<cc:interface>`. Скопируйте необходимую для компонента разметку из Facelets-страницы ввода данных внутрь действия `<cc:implementation>`. Замените в EL-выражениях обращения к

управляемым бинам на обращения к атрибутам компонента  
(`# {cc.attrs.имяАтрибута}`).

13. Замените в Facelets-странице разметку, скопированную в составной компонент, на вызов этого компонента. Запустите веб-приложение и убедитесь в его правильном функционировании.