

Лабораторная работа №6

Знакомство с Web-технологиями

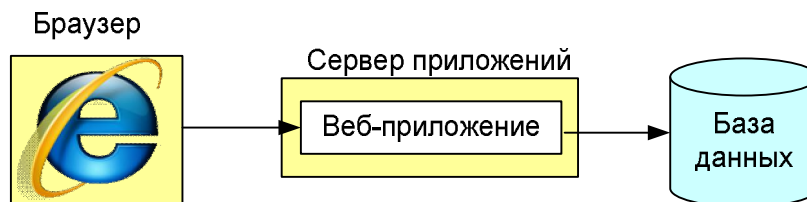
Содержание

1. Лабораторная работа 6.....	2
1.1. Постановка задачи.....	2
1.2. Требования к работе.....	2
1.3. Требования к оформлению отчета.....	2
1.4. Варианты заданий	2
2. Рекомендации по выполнению работы.....	6
2.1. Установка сервера приложений.....	6
2.2. Создание веб-приложения в среде NetBeans v7	7
2.3. Разработка Java-сервлета	10
2.4.1. Интерфейс Servlet.....	10
2.4.2. Класс HTTPServlet.....	11
2.4.3. Создание сервлета в NetBeans.....	11
2.4.4. Пример простого сервлета	13

1. Лабораторная работа 6

1.1. Постановка задачи

Разработайте веб-приложение, предоставляющее пользователю сведения об объектах (в соответствии с заданной предметной областью). Информация об объектах читается из базы данных, созданной в рамках лабораторной работы №2.



1.2. Требования к работе

Веб-приложение выполняется на сервере приложений. Доступ к веб-приложению осуществляется через стандартный браузер.

Приложение динамически формирует веб-страницу на основе данных, хранящихся в базе. Внешний вид веб-страницы должен задаваться при помощи таблиц стилей в отдельном файле CSS.

Рекомендуемый язык программирования – Java (!!! **Java EE SDK v6**).

Рекомендуемая СУБД – MySQL v5.5.

Рекомендуемое средство доступа к данным – JDBC.

Рекомендуемая среда разработки приложения – NetBeans IDE 7 for Java EE

Рекомендуемый сервер приложений – GlassFish 3

1.3. Требования к оформлению отчета

Отчет должен содержать:

- титульный лист
- постановку задачи
- исходный код программы (включая *web.xml*, таблицы *css*, и т.п.)
- описание программы (описание классов, методов, полей)

1.4. Варианты заданий

Вариант 1	
Предметная область	Карта мира
Объекты	Страны, Города
Примечание	Карта мира содержит множество <i>стран</i> . Для каждой <i>страны</i> определено множество <i>городов</i> .
Требуемые операции	Выдача полного списка городов с указанием названия страны

Вариант 2	
Предметная область	Библиотека
Объекты	Авторы, Книги
Примечание	Книги в библиотеке сгруппированы по <i>авторам</i> . У каждого <i>автора</i> имеется множество <i>книг</i> .
Требуемые операции	Выдача полного списка книг с указанием ФИО автора

Вариант 3	
Предметная область	Отдел кадров
Объекты	Подразделения, Сотрудники
Примечание	Имеется множество <i>подразделений</i> предприятия. В каждом <i>подразделении</i> работает множество <i>сотрудников</i> .
Требуемые операции	Выдача списка сотрудников с указанием названия подразделения

Вариант 4	
Предметная область	Учебный отдел
Объекты	Группы, Студенты
Примечание	Имеется множество учебных <i>групп</i> . Каждая группа включает в себя множество <i>студентов</i> .
Требуемые операции	Выдача полного списка студентов с указанием названия группы

Вариант 5	
Предметная область	Автосалон
Объекты	Производители автомобилей, Марки
Примечание	<i>Марки</i> автомобилей сгруппированы по производителям. У каждого <i>производителя</i> имеется множество <i>марок</i> .
Требуемые операции	Выдача полного списка марок с названием производителя

Вариант 6	
Предметная область	Агентство новостей
Объекты	Категории новостей, Новости
Примечание	Новости сгруппированы по <i>категориям</i> . У каждой <i>категории</i> имеется множество <i>новостей</i> .
Требуемые операции	Выдача полного списка новостей с указанием категории

Вариант 7	
Предметная область	Продуктовый магазин
Объекты	Категория продукта, Продукт
Примечание	<i>Продукты</i> в магазине сгруппированы по <i>категориям</i> . Для каждой <i>категории</i> определено множество <i>продуктов</i> .
Требуемые операции	Выдача списка продуктов с указанием категории

Вариант 8	
Предметная область	Футбол
Объекты	Команды, Игроки
Примечание	Имеется множество футбольных <i>команд</i> . Для каждой <i>команды</i> определено множество <i>игроков</i> .
Требуемые операции	Выдача полного списка игроков с указанием названия команды

Вариант 9	
Предметная область	Музыкальный магазин
Объекты	Исполнители, Альбомы
Примечание	В музыкальном магазине <i>альбомы</i> сгруппированы по <i>исполнителям</i> . Для каждого <i>исполнителя</i> задано множество <i>альбомов</i> .
Требуемые операции	Выдача полного списка альбомов с указанием исполнителя

Вариант 10	
Предметная область	Аэропорт
Объекты	Авиакомпании, Рейсы
Примечание	Имеется множество <i>авиакомпаний</i> . Для каждой <i>авиакомпании</i> определены ее <i>рейсы</i> .
Требуемые операции	Выдача полного списка рейсов с указанием названия авиакомпании

Вариант 11	
Предметная область	Файловая система
Объекты	Папки, Файлы
Примечание	Имеется множество <i>папок</i> (независимых друг от друга). Для каждой <i>папки</i> определено множество <i>файлов</i> .
Требуемые операции	Выдача списка файлов с указанием папки

Вариант 12	
Предметная область	Расписание занятий
Объекты	Дни недели, Занятия
Примечание	Имеется множество <i>дней</i> . Для каждого <i>дня</i> определен перечень <i>занятий</i> .
Требуемые операции	Выдача полного списка занятий с указанием дня

Вариант 13	
Предметная область	Записная книжка
Объекты	Календарные дни, Мероприятия
Примечание	Имеется множество <i>дней</i> . Для каждого <i>дня</i> определен перечень <i>мероприятий</i> .
Требуемые операции	Выдача полного списка мероприятий с указанием дня

Вариант 14	
Предметная область	Видеомагазин
Объекты	Жанры, Фильмы
Примечание	Имеется множество <i>жанров</i> . Для каждого <i>жанра</i> определен перечень <i>фильмов</i> .
Требуемые операции	Выдача списка фильмов с указанием жанра

Вариант 15	
Предметная область	Железная дорога
Объекты	Дороги, Станции
Примечание	Имеется множество <i>железных дорог</i> . В ведомстве каждой дороги находится множество <i>станций</i> .
Требуемые операции	Выдача полного списка станций с указанием названия дороги

Вариант 16	
Предметная область	Склад
Объекты	Секции, Товары
Примечание	<i>Товары</i> на складе сгруппированы по <i>секциям</i> . Для каждой <i>секции</i> задано множество <i>товаров</i> .
Требуемые операции	Выдача списка товаров с указанием секции

Вариант 17	
Предметная область	Кафедра университета
Объекты	Преподаватели, Дисциплины
Примечание	На кафедре имеется множество <i>преподавателей</i> . Для каждого <i>преподавателя</i> задано множество <i>дисциплин</i> .
Требуемые операции	Выдача списка дисциплин с указанием ФИО преподавателя

Вариант 18	
Предметная область	Программное обеспечение
Объекты	Производители, Программные продукты
Примечание	Программные <i>продукты</i> сгруппированы по <i>производителям</i> . Для каждого <i>производителя</i> задано множество <i>продуктов</i> .
Требуемые операции	Выдача списка продуктов с указанием производителя

Вариант 19	
Предметная область	Геометрия
Объекты	Многоугольники, Вершины
Примечание	Имеется множество <i>многоугольников</i> . Каждый <i>многоугольник</i> состоит из произвольного числа <i>вершин</i> .
Требуемые операции	Выдача полного списка многоугольников с указанием всех вершин

Вариант 20	
Предметная область	Схема метро
Объекты	Линии, Станции
Примечание	Имеется множество <i>линий</i> метрополитена. Каждая <i>линия</i> состоит из последовательности <i>станций</i> .
Требуемые операции	Выдача списка станций с указанием линии

2. Рекомендации по выполнению работы

2.1. Установка сервера приложений

Для выполнения лабораторной работы потребуется установить на компьютер сервер приложений. В данной работе рекомендуется использовать свободно распространяемый сервер GlassFish.

Дистрибутив сервера GlassFish доступен на официальном сайте компании Oracle по ссылке:

<http://www.oracle.com/technetwork/middleware/glassfish/downloads/index.html>

Сервер GlassFish также входит в дистрибутив Java EE SDK:

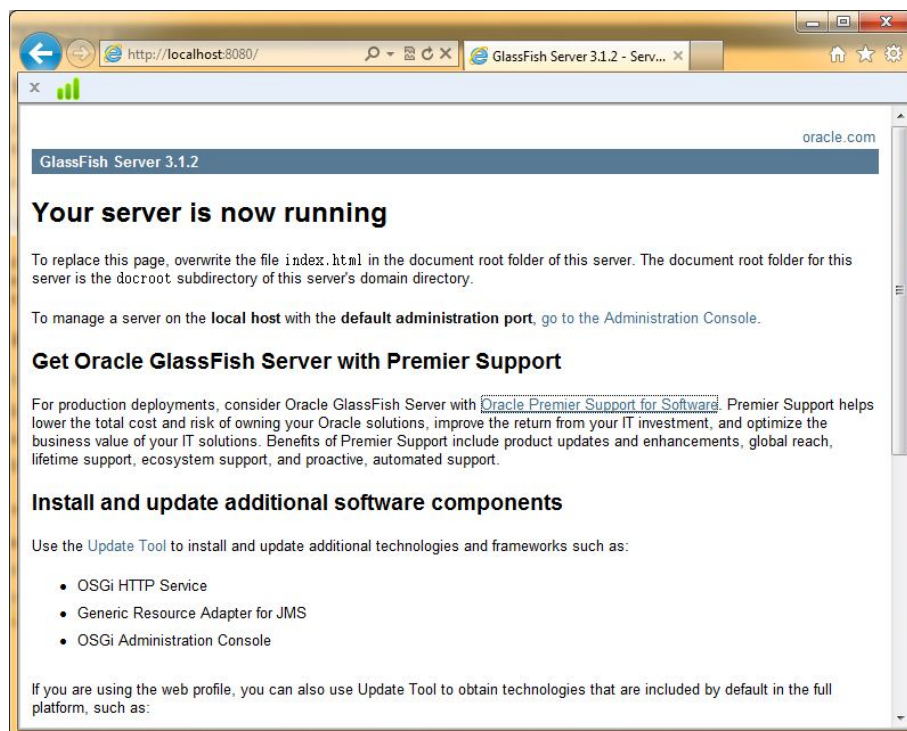
<http://www.oracle.com/technetwork/java/javaee/downloads/java-ee-sdk-6u3-jdk-7u1-downloads-523391.html>

Кроме того, GlassFish включен в дистрибутив среды разработки NetBeans IDE J2EE:

<http://netbeans.org/downloads/>

Инструкция по установке и настройке сервера приложений GlassFish v3 размещается здесь: http://docs.oracle.com/cd/E26576_01/doc.312/e24935/toc.htm

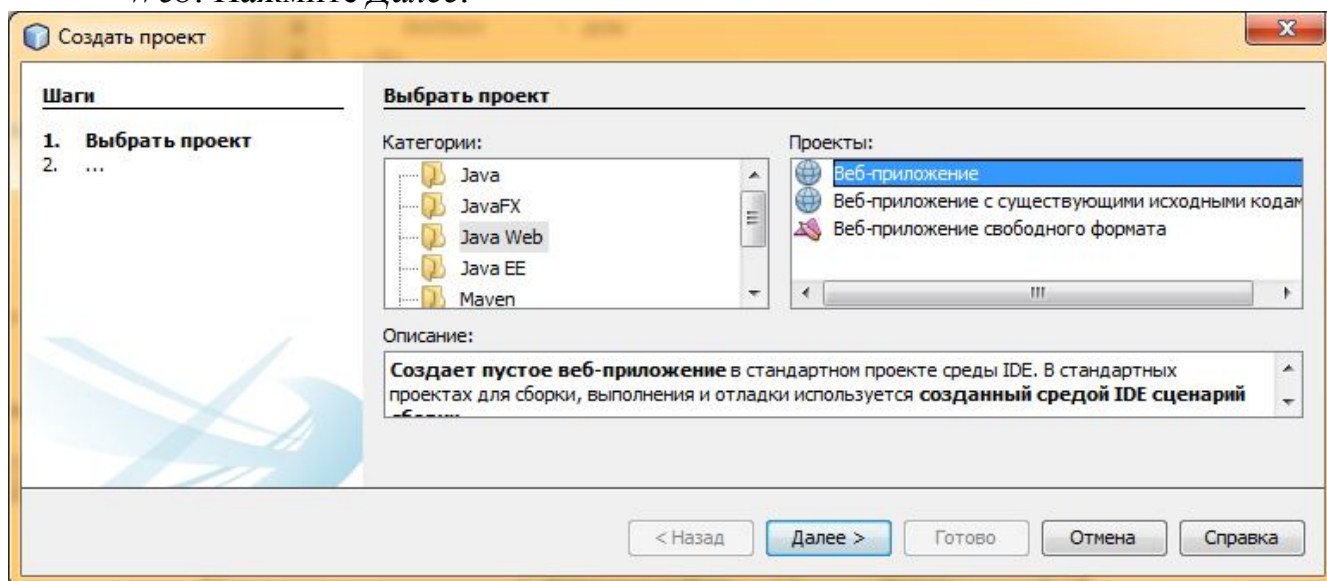
В случае успешной установки и запуска сервера, по адресу <http://localhost:8080> будет доступна приветственная страница GlassFish.



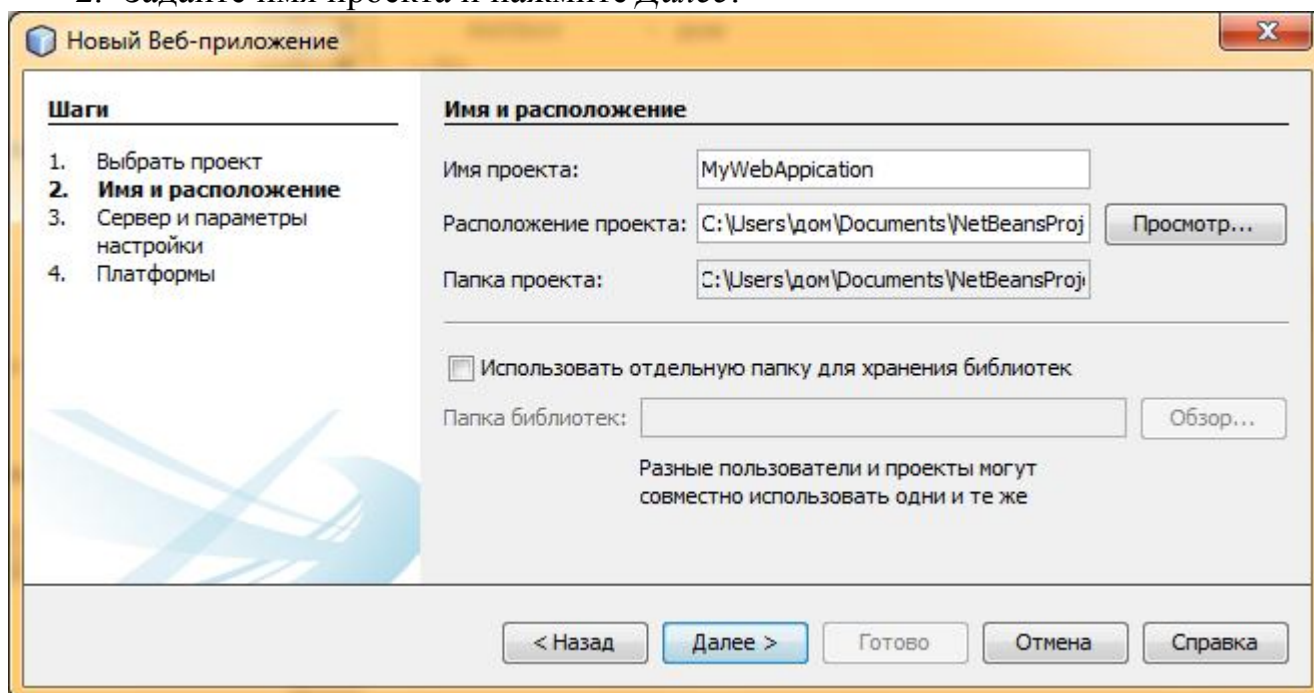
2.2. Создание веб-приложения в среде NetBeans v7

Рассмотрим процесс создания веб-приложения на примере среды NetBeans v7:

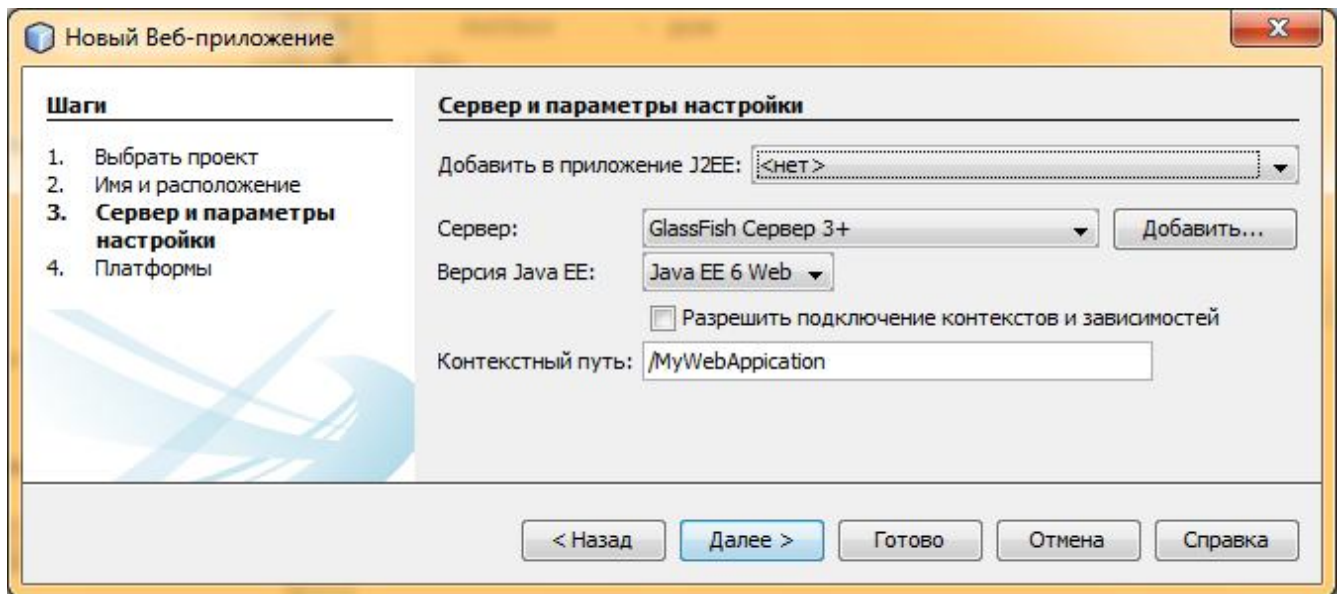
1. Запустите среду NetBeans и вызовите из главного меню команду создания нового проекта. Выберите тип проекта *Веб-приложение* категории *Java Web*. Нажмите *Далее*.



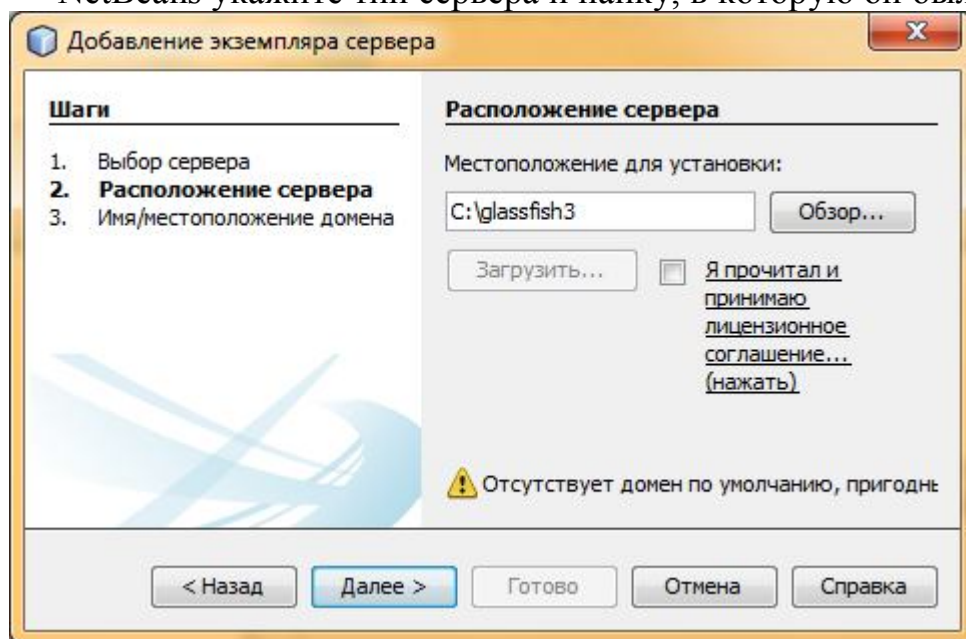
2. Задайте имя проекта и нажмите *Далее*:



3. Выберите сервер приложений, который планируется использовать для запуска веб-приложения (например, GlassFish 3).

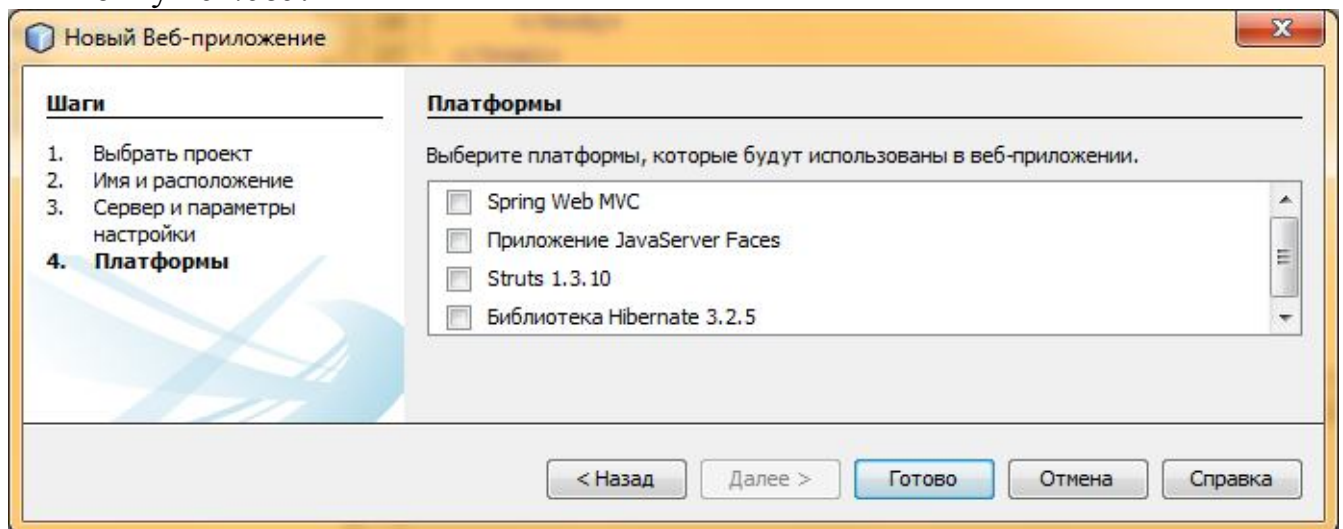


4. Если среда NetBeans не нашла сервер приложений, зарегистрируйте его самостоятельно (кнопка *Добавить*). При регистрации сервера в среде NetBeans укажите тип сервера и папку, в которую он был установлен.



5. Укажите контекстный путь. Контекстный путь определяет адрес вашего приложения в строке URI. Нажмите кнопку *Далее*.

6. Выберите при необходимости дополнительные библиотеки и нажмите кнопку *Готово*:



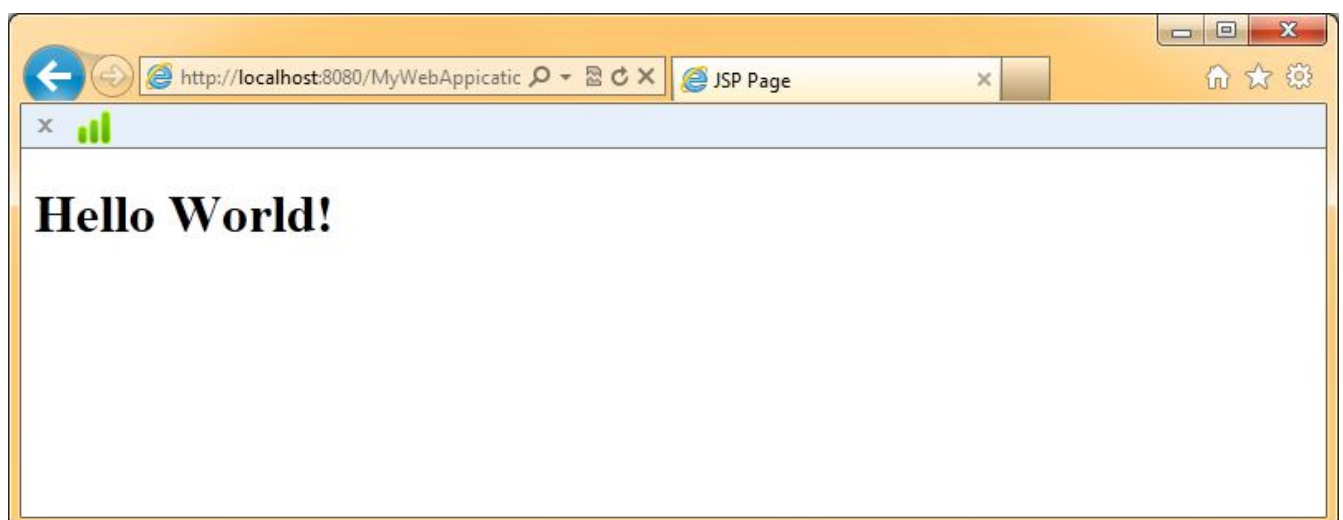
Веб-приложение создано.

Созданное в среде NetBeans веб-приложение Java по умолчанию включает в себя: стартовую JSP-страницу – `index.jsp`.

7. Запустите веб-приложение, щелкнув правой кнопкой мыши на проекте и вызвав команду *Выполнение*. NetBeans откомпилирует проект, и установит веб-приложение на заданный сервер приложений. Проверить работоспособность приложения можно через любой стандартный браузер, указав в адресной строке URI приложения. Например: <http://localhost:8080/MyWebApplication>

где:

- localhost – сетевое имя сервера
- 8080 – порт сервера приложений
- MyWebApplication – контекстный путь



2.3. Разработка Java-сервлета

Сервлет Java – класс, предназначенный для обработки вызовов HTTP и генерации содержимого веб-страницы. Сервлеты являются специализированным механизмом Java для создания Web ресурсов.

Для разработки сервлетов используется пакет `javax.servlet`.

2.4.1. Интерфейс Servlet

Все пользовательские сервлеты должны поддерживать интерфейс `javax.servlet.Servlet`. Данный интерфейс определяет методы, которые управляют сервлетом и его общением с клиентом. При написании нового сервлета, разработчик должен реализовать нужные методы этого интерфейса.

Всего интерфейс содержит пять методов.

```
public void init(ServletConfig config) throws ServletException
```

Данный метод вызывается в момент первой загрузки сервлета. Метод позволяет осуществить инициализацию ресурсов (открытие файлов, установка соединений с удаленными серверами, и т.п.). Гарантируется, что метод `init()` закончится перед любым другим обращением к сервлету. Параметр `config` содержит сведения о конфигурации сервлета.

```
public ServletConfig getServletConfig()
```

Используется для получения системной информации о конфигурации сервлета.

```
public String getServletInfo()
```

Используется для возврата строки, содержащей общие сведения о сервлете, определяемые программистом

```
public void service(ServletRequest request,  
    ServletResponse response)  
    throws ServletException, java.io.IOException
```

Метод `service()` является сердцем сервлета. Каждый запрос от клиента приводит к одному вызову метода `service()`. Этот метод читает запрос и формирует ответное сообщение при помощи своих двух аргументов `ServletRequest` и `ServletResponse` :

- Объект `ServletRequest` содержит параметры HTTP запроса (данные, полученные от клиента).
- Объект `ServletResponse` содержит ответ сервлета клиенту.

Работа метода `service()` по существу проста – он создает ответ на каждый клиентский запрос, переданный ему с сервера. Однако необходимо помнить, что могут существовать несколько параллельных запросов, обрабатываемых в одно и то же время. Если метод `service()` требует каких-либо внешних ресурсов, таких как файлы, базы данных, то необходимо гарантировать, чтобы доступ к ресурсам являлся потокозащищенным.

```
public void destroy()
```

Метод `destroy()` вызывается для освобождения всех ресурсов (например, открытые файлы и соединения с базой данных) перед выгрузкой сервлета. Этот метод может быть пустым, если нет необходимости выполнения каких-либо завершающих операций.

2.4.2. Класс *HTTPServlet*

Пакет `javax.servlet` содержит два абстрактных класса, которые реализуют интерфейс `Servlet`: класс `GenericServlet` (из пакета `javax.servlet`) и класс `HttpServlet` (из пакета `javax.servlet.http`). Эти классы предоставляют реализацию по умолчанию для всех методов интерфейса `Servlet`. Большинство разработчиков используют либо класс `GenericServlet`, либо класс `HttpServlet`, и замещают некоторые или все методы этих классов.

Сервлеты, работающие с клиентами через Web, обычно расширяют класс `HttpServlet`. Метод `service`, как правило, переопределяется, чтобы иметь возможность различать стандартные методы запросов, получаемые от Web-браузера клиента. Двумя наиболее распространенными типами запросов HTTP (их также называют методами запросов) являются `get` и `post`. Запрос `get` получает (или извлекает) информацию. Запрос `post` помещает (или отправляет) данные на сервер. Типичное применение метода `post` - отправка на сервер информации для аутентификации, или данных из формы, в которую пользователь ввел информацию.

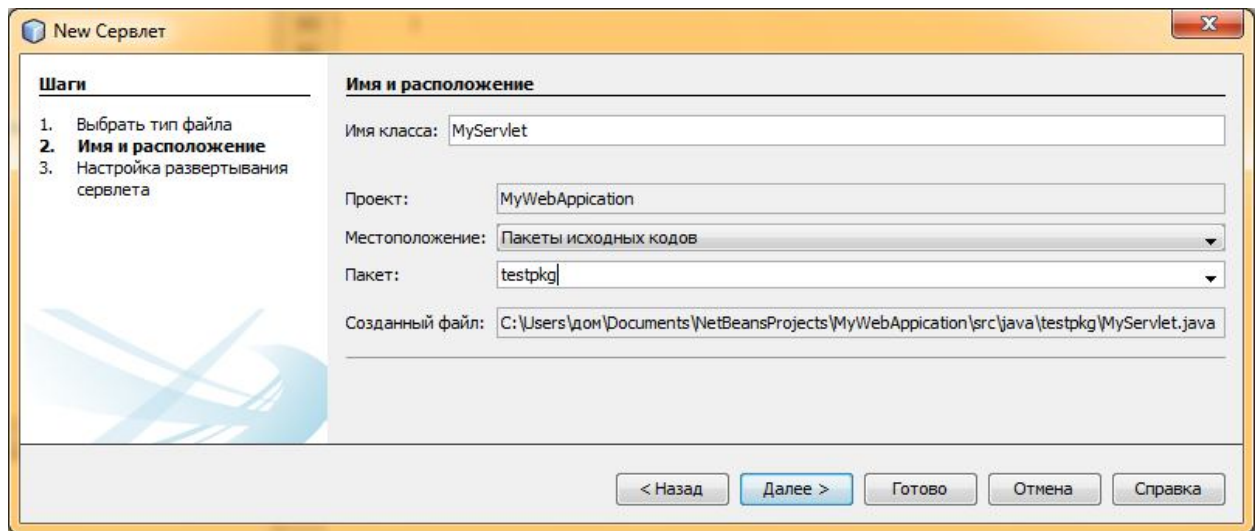
В классе `HttpServlet` в частности определены методы `doGet` и `doPost` для реакции на запросы типа `get` и `post` клиента. Эти методы вызываются методом `service` класса `HttpServlet`, который, в свою очередь, вызывается при поступлении запроса на сервер. Метод `service` сначала определяет тип запроса, а затем вызывает соответствующий метод.

Методы `doGet` и `doPost` принимают в качестве параметров объекты `HttpServletRequest` и `HttpServletResponse`, которые дают возможность осуществлять взаимодействие между клиентом и сервером. Методы интерфейса `HttpServletRequest` облегчают доступ к данным запроса. Методы интерфейса `HttpServletResponse` облегчают возврат результатов Web-клиенту в виде HTML.

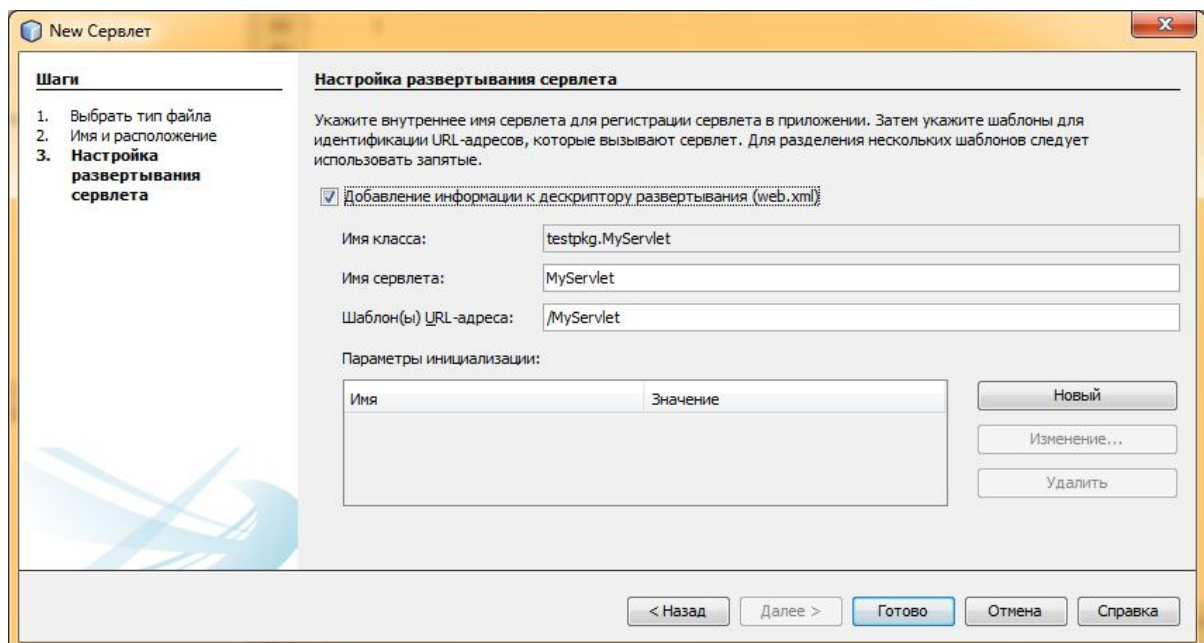
2.4.3. Создание сервлета в NetBeans

Для создания нового сервлета в среде NetBeans вызовите контекстное меню на веб-проекте и выполните команду *Новый – Сервлет*.

1. В появившемся диалоге укажите имя класса сервлета и задайте пакет. Нажмите кнопку *Далее*.



2. Выберите опцию *Добавление информации к дескриптору развертывания* и нажмите кнопку *Готово*.



В проекте будет создано два новых файла: файл с исходным кодом класса нового сервлета, унаследованного от класса `HttpServlet`, и дескриптор развертывания `web.xml`.

Дескриптор развертывания описывает классы, ресурсы и конфигурацию приложения. Данный дескриптор используется веб-сервером, в частности, для сопоставления указанного в запросе пути URL с определенным сервлетом.

2.4.4. Пример простого сервлета

Создадим простой сервлет, формирующий веб-страницу с таблицей умножения. Для оформления таблицы будем использовать стили CSS.

Сервлет (файл MyServlet.java)

```
package testpkg;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class MyServlet extends HttpServlet
{    // Обработчик HTTP-метода Get()
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Таблица умножения</title>");
            out.println("<link href =\"newcss.css\""+
                        "rel=\"stylesheet\" type=\"text/css\">");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Таблица умножения</h1>");

            // ВЫВОЖУ ТАБЛИЦУ
            out.println("<table>");
            // Вывожу заголовок
            out.println("<tr>");
            out.println("<th></th>");
            for (int i=2; i<10; i++)
                out.println("<th>"+i+"</th>");
            out.println("</tr>");
            // Вывожу содержимое
            for (int i=2; i<9; i++)
            {
                out.println("<tr>");
                out.println("<th>"+i+"</th>");
                for (int j=2; j<10; j++)
                    out.println("<td>"+i*j+"</td>");
                out.println("</tr>");
            }
            out.println("</table>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

```

        } finally {
            out.close();
        }
    }

    public String getServletInfo() {
        return "Таблица умножения";
    }
}

```

Таблица стилей (файл newcss.css)

```

table {
    font-size: 14pt;
    border: 1px solid black;
}

th {
    width: 50px;
    height: 20px;
    text-align: center;
    background-color: greenyellow;
    border: 1px solid black;
}

td {
    width: 50px;
    height: 20px;
    text-align: center;
    border: 1px solid black;
}

```

Дескриптор развертывания (файл web.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
    <servlet>
        <servlet-name>MyServlet</servlet-name>
        <servlet-class>testpkg.MyServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>MyServlet</servlet-name>
        <url-pattern>/MyServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>

```

Запуск сервлета

Разработанное веб-приложение следует откомпилировать и разместить на сервере приложений. Для запуска сервлета следует открыть браузер и перейти по ссылке вида:

`http://сервер:порт/контекстный_путь_приложения/адрес_сервлета`

