**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

Кафедра вычислительной техники

**Отчет по лабораторной работе №2  
по дисциплине «Web-программирование»**

**Тема: построение web-приложений на основе сервлетов**

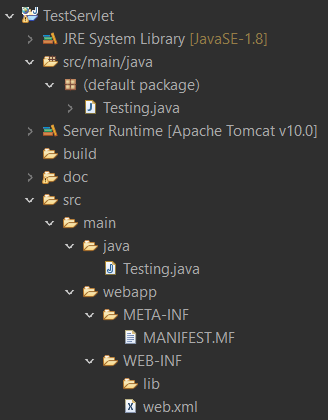
|  |  |
| --- | --- |
| Студент гр. 2310 | Альсакма О.С.М |
| Преподаватель | Павловский М.Г. |

**Цель работы**

Знакомство с технологией построения Web-приложений на основе сервлетов

Предметная область – та же, что и в курсовой работе: футбольная команда

**Структура web-приложения**

****

**Сервлет web-приложения**

Создадим сервлет, который выводит список футболистов, зарплата которых больше указанной в запросе. Если зарплата не указана – весь список

import java.io.IOException;

import java.io.PrintWriter;

import jakarta.servlet.ServletException;

//import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

//@WebServlet("/TeamList")

/\*\*

\* Servlet implementation class Testing

\*/

public class Testing extends HttpServlet {

private static final long serialVersionUID = 1L;

private Object[][] team;

private String[] roles;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Testing() {

super();

team = new Object[][] {

{"Билли Херрингтон", 0, "Махачкала", 15000},

{"Антон Чехов", 1, "Санкт-Петербург", 30000},

{"Илья Антонов", 2, "Екатеринбург", 25000},

{"Андрей Сачков", 3, "Вологда", 19000}

};

roles = new String[] {"Вратарь", "Нападающий", "Полузащитник", "Защитник"};

}

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse

response)

throws ServletException, IOException {

// Задание типа кодировки для параметров запроса

request.setCharacterEncoding("utf-8");

// Чтение параметра name из запроса

String salary = request.getParameter("salary");

// Задание типа содержимого для ответа (в том числе кодировки)

response.setContentType("text/html;charset=UTF-8");

// Получение потока для вывода ответа

PrintWriter out = response.getWriter();

try {

// Создание HTML-страницы

out.println("<html>");

out.println("<head><title>Список моих футболистов</title></head>");

out.println("<body>");

out.println("<h1>Вот мои футболисты" + ((salary == null)? " ": " с зарплатой >= "

+ salary + "$") + "</h1>");

out.println("<table border='1'>");

out.println("<tr><td><b>Имя фамилия</b></td>"

+ "<td><b>Специальность</b></td>"

+ "<td><b>Город</b></td>"

+ "<td><b>Зарплата</b></td></tr>");

for (Object[] temp : team)

if (salary == null || (int)temp[3] >= Integer.parseInt(salary))

out.println("<tr><td>" + temp[0] + "</td><td>" + roles[(int)temp[1]] + "</td><td>"

+ temp[2] + "</td><td>" + Integer.toString((int)temp[3]) + "</td></tr>");

out.println("</table>");

out.println("</body>");

out.println("</html>");

} finally {

// Закрытие потока вывода

out.close();

}

}

/\*\*

\* Handles the HTTP

\* <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP

\* <code>POST</code> method.

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

}

Теперь нужно сделать разметку сервлета, что сервер смог его найти. Для этого в web.xml были добавлены следующие строки

<**servlet**>

<**servlet-name**>Testing</**servlet-name**>

<**servlet-class**>Testing</**servlet-class**>

</**servlet**>

<**servlet-mapping**>

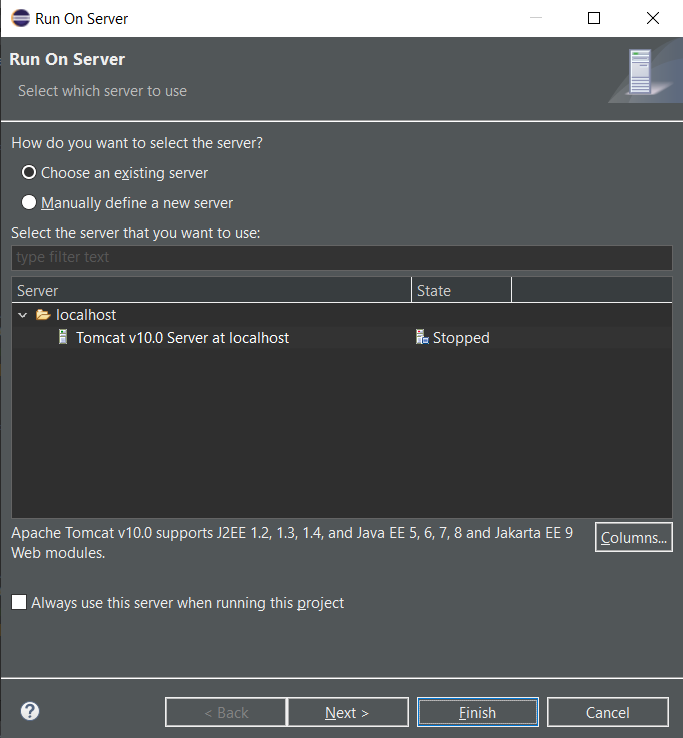
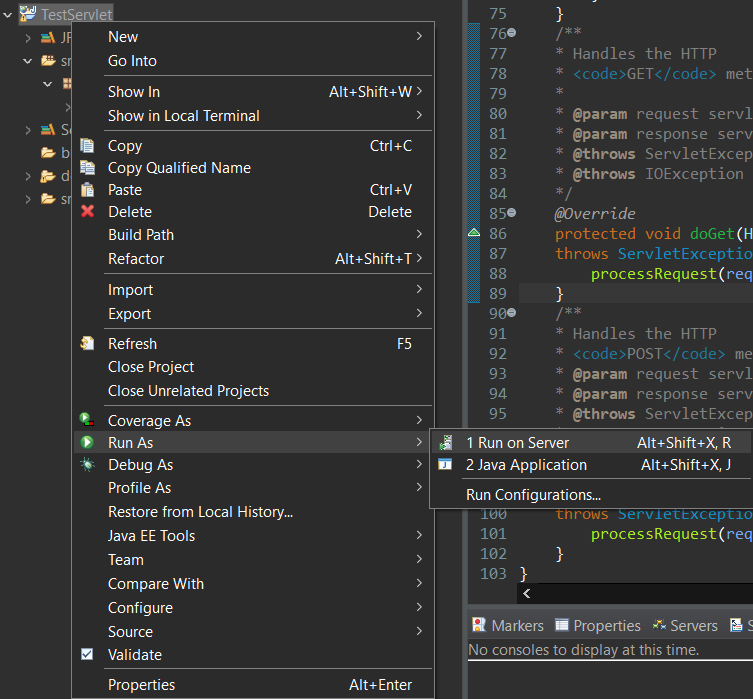
<**servlet-name**>Testing</**servlet-name**>

<**url-pattern**>/TeamList</**url-pattern**>

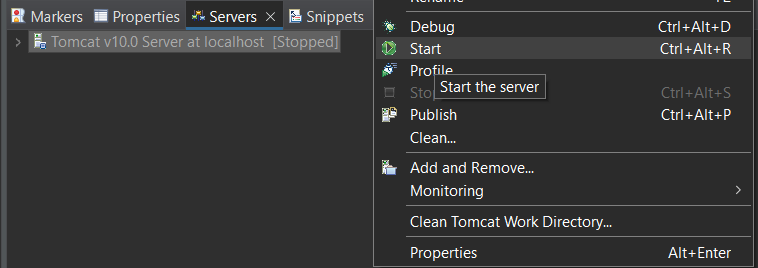
</**servlet-mapping**>

Здесь servlet-name может быть любым (главное, чтобы в обоих блоках имя было одинаковым), servlet-class должен иметь имя java-класса, где этот сервлет находится, а в url-pattern отображается адрес сервлета

**Размещение проекта на сервере Tomcat**

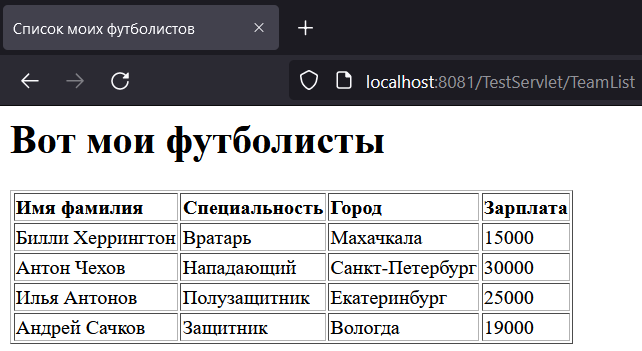


Далее сервер запускается нажатием ПКМ на сервер и выбором опции Start

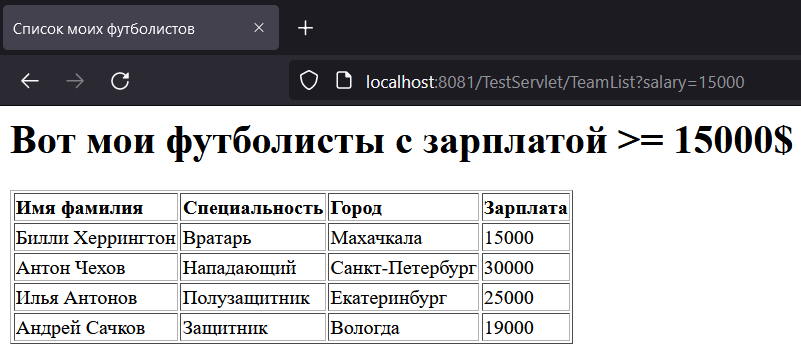


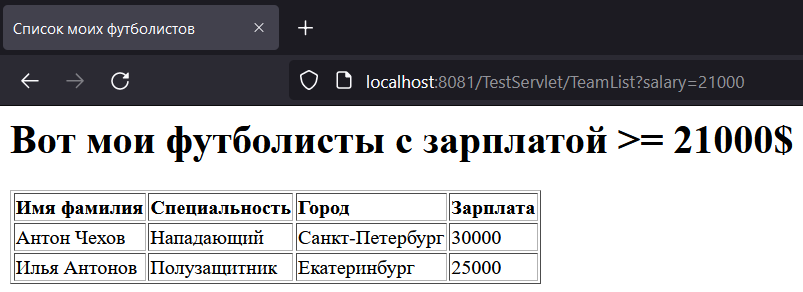
**Запуск web-приложения и обращение к сервлету через браузер**

Обращаемся к сервлету через имя проекта:



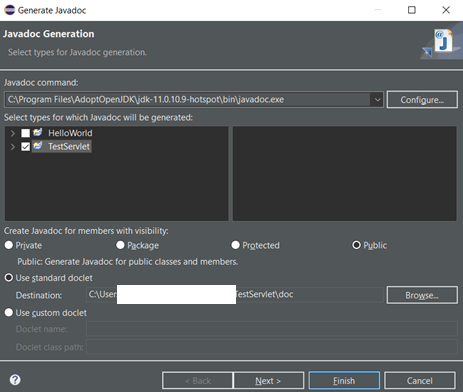
Передаем сервлету динамический параметр – зарплату



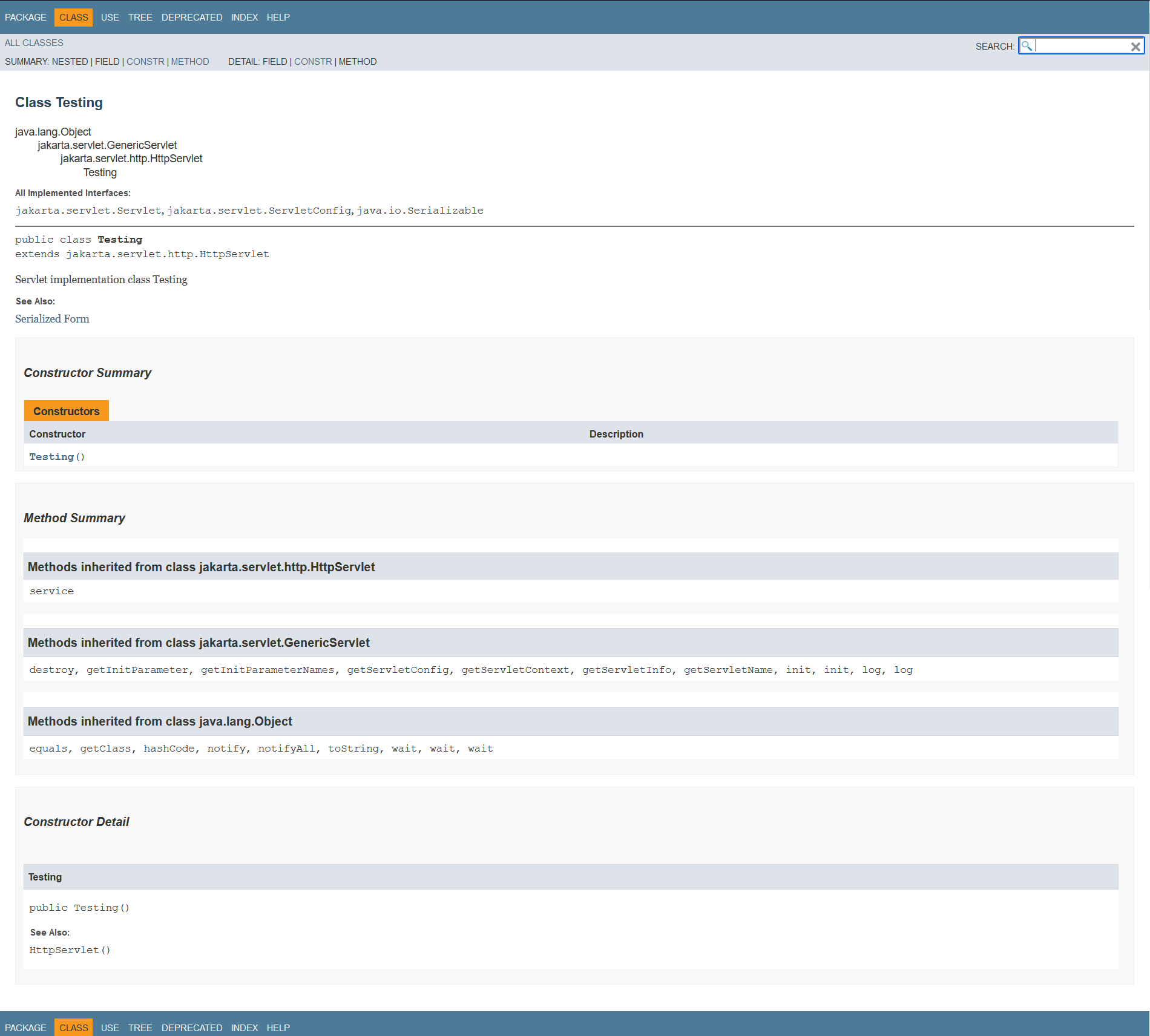


**Генерация документации**

Документация генерируется с помощью Project -> Generate JavaDoc



Результат:



**Вывод**

В ходе выполнения лабораторной работы были изучены и освоены основы построения Web-приложений на основе сервлетов. В частности, были выполнены следующие шаги:

1. **Создание сервлета**: Был разработан сервлет, который выводит список футболистов с зарплатой, превышающей указанную в запросе. Если зарплата не указана, выводится весь список.
2. **Настройка сервлета**: Для корректной работы сервлета были добавлены записи в файл web.xml, чтобы сервер Tomcat мог найти и запустить сервлет.
3. **Размещение проекта на сервере Tomcat**: Проект был размещен на сервере Tomcat, и сервлет был успешно запущен.
4. **Тестирование сервлета**: Сервлет был протестирован через браузер, где были переданы динамические параметры (зарплата) для фильтрации списка футболистов.
5. **Генерация документации**: Была сгенерирована документация для сервлета с использованием инструмента JavaDoc.

Таким образом, в результате выполнения лабораторной работы было успешно создано и протестировано Web-приложение на основе сервлетов, а также сгенерирована документация для кода.