

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.

Санкт-Петербург

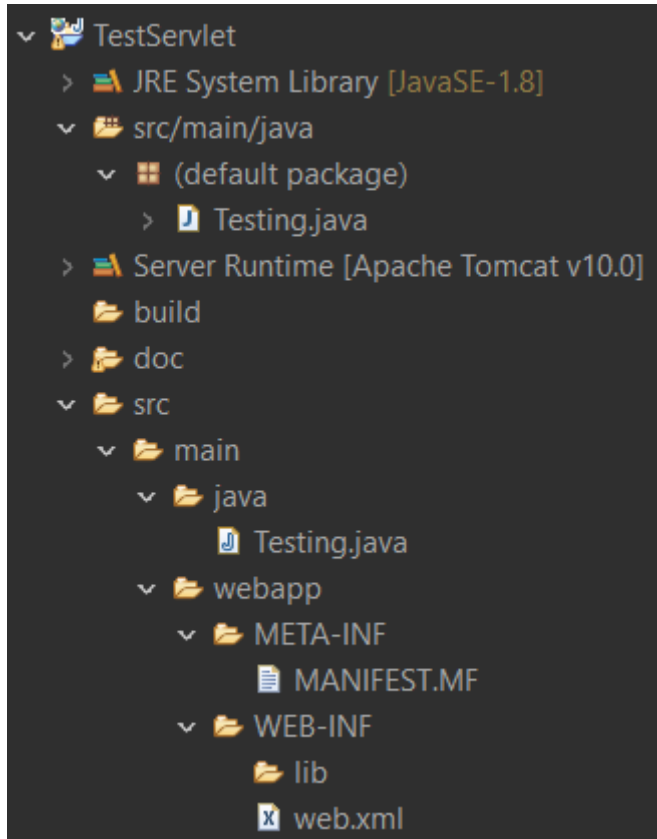
2024

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

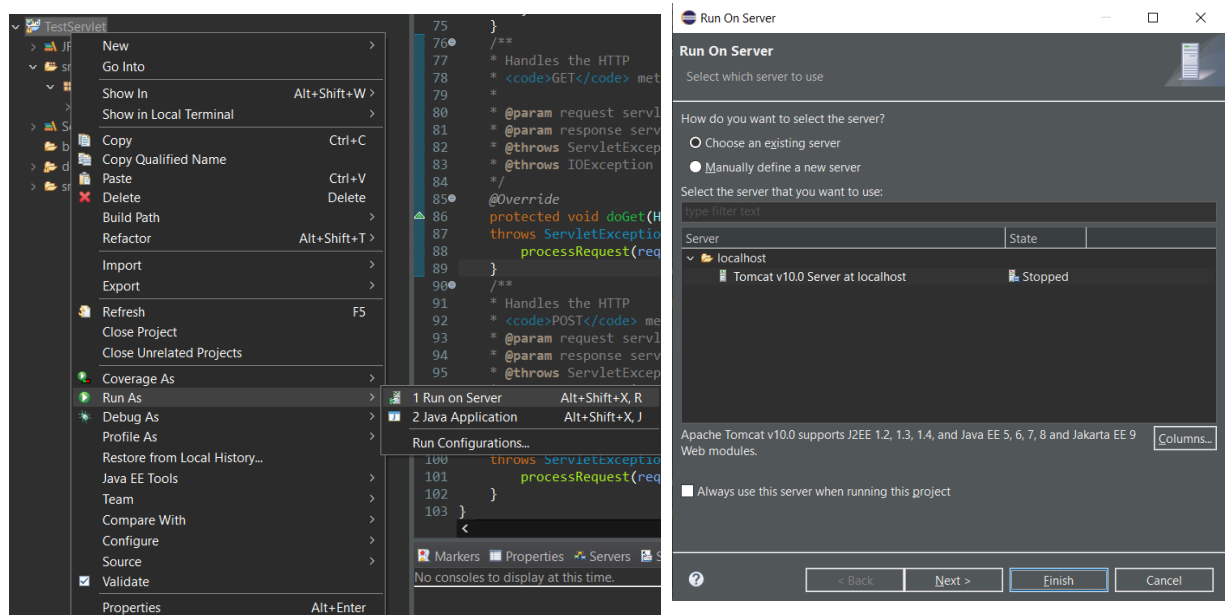
```

<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>

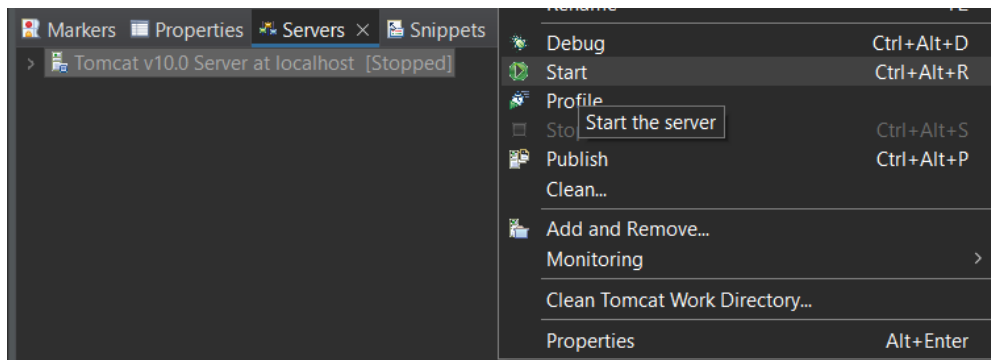
```

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

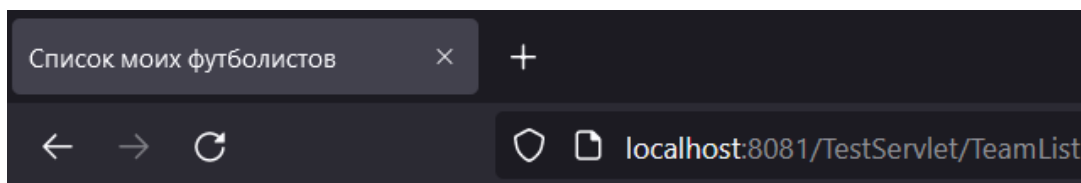


Далее сервер запускается нажатием ПКМ на сервер и выбором опции Start



Запуск web-приложения и обращение к сервлету через браузер

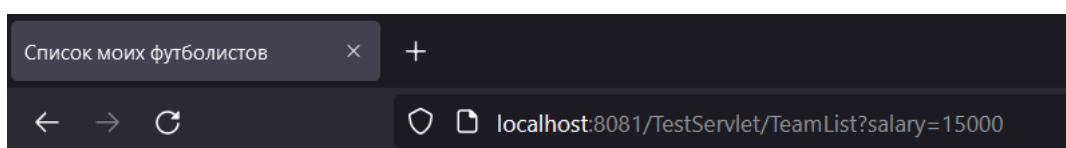
Обращаемся к сервлету через имя проекта:



Вот мои футболисты

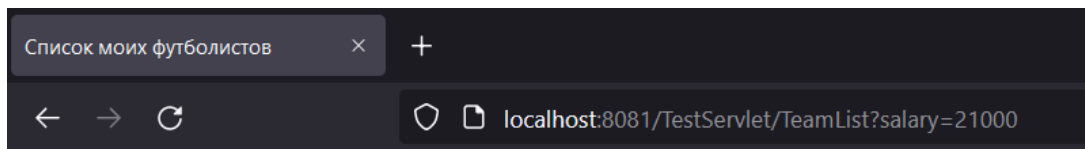
Имя фамилия	Специальность	Город	Зарплата
Билли Херрингтон	Вратарь	Махачкала	15000
Антон Чехов	Нападающий	Санкт-Петербург	30000
Илья Антонов	Полузащитник	Екатеринбург	25000
Андрей Сачков	Защитник	Вологда	19000

Передаем сервлету динамический параметр – зарплату



Вот мои футболисты с зарплатой $\geq 15000\$$

Имя фамилия	Специальность	Город	Зарплата
Билли Херрингтон	Вратарь	Махачкала	15000
Антон Чехов	Нападающий	Санкт-Петербург	30000
Илья Антонов	Полузащитник	Екатеринбург	25000
Андрей Сачков	Защитник	Вологда	19000

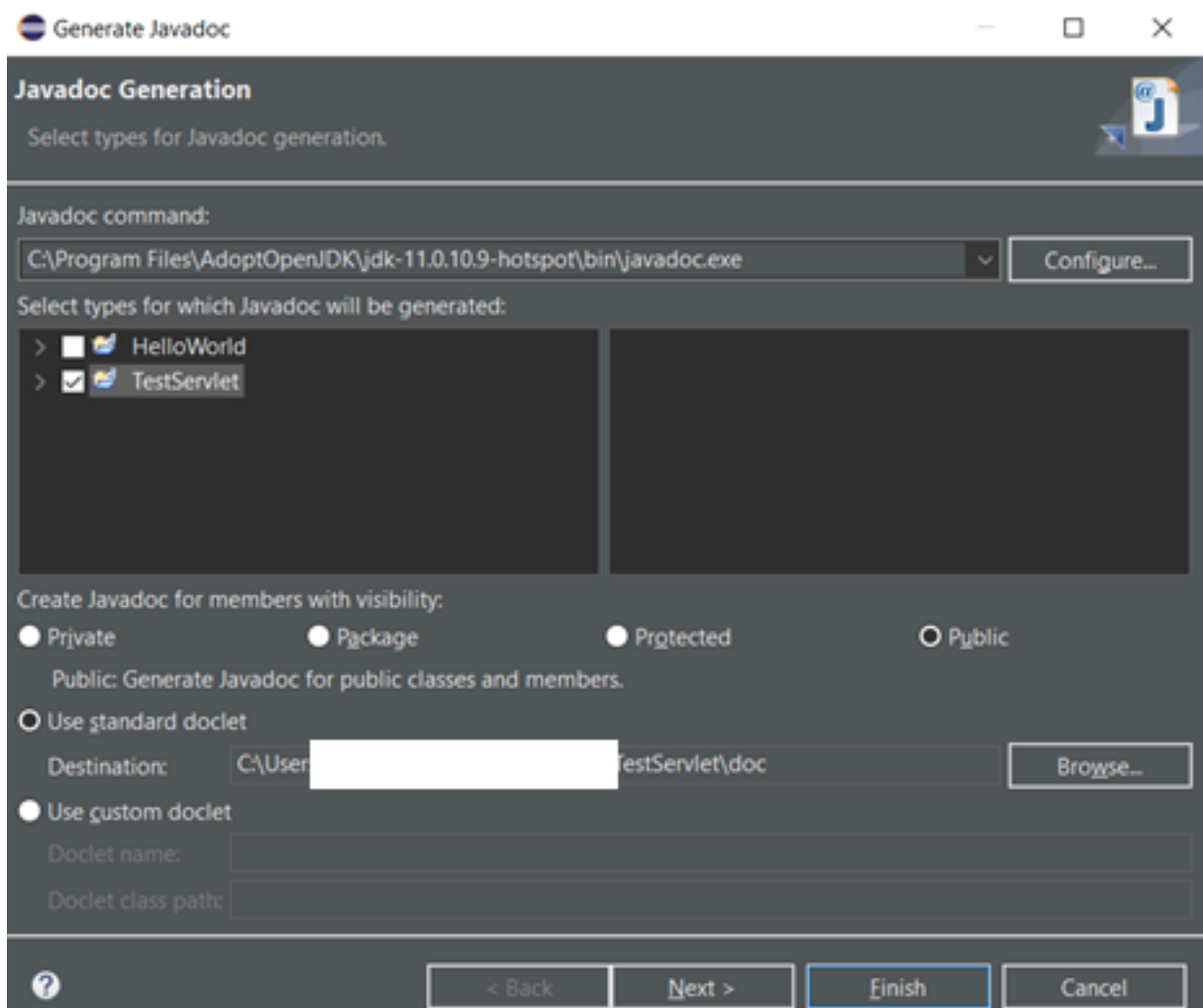


Вот мои футболисты с зарплатой $\geq 21000\$$

Имя фамилия	Специальность	Город	Зарплата
Антон Чехов	Нападающий	Санкт-Петербург	30000
Илья Антонов	Полузащитник	Екатеринбург	25000

Генерация документации

Документация генерируется с помощью Project -> Generate Javadoc



Результат:

PACKAGE
CLASS
USE
TREE
DEPRECATED
INDEX
HELP

ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
SEARCH

Class Testing

[java.lang.Object](#)
[jakarta.servlet.GenericServlet](#)
[jakarta.servlet.http.HttpServlet](#)
Testing

All Implemented Interfaces:
[jakarta.servlet.Servlet](#), [jakarta.servlet.ServletConfig](#), [java.io.Serializable](#)

```

public class Testing
extends jakarta.servlet.http.HttpServlet
Servlet implementation class Testing
See Also:
Serialized Form

```

Constructor Summary

Constructors	Description
Testing()	

Method Summary

Methods inherited from class [jakarta.servlet.http.HttpServlet](#)

[service](#)

Methods inherited from class [jakarta.servlet.GenericServlet](#)

[destroy](#), [getInitParameter](#), [getInitParameterNames](#), [getServletConfig](#), [getServletContext](#), [getServletInfo](#), [getServletName](#), [init](#), [init](#), [log](#), [log](#)

Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Testing

```

public Testing()
See Also:
HttpServlet\(\)

```

Вывод

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

- Создание сервлета:** Был разработан сервлет, который выводит список футболистов с зарплатой, превышающей указанную в запросе. Если зарплата не указана, выводится весь список.
- 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.