

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И.УЛЬЯНОВА (ЛЕНИНА)**

Кафедра ВТ

**ОТЧЕТ
по лабораторной работе №3
по дисциплине «Java программирование
интернет-приложений»
Тема: Построение WEB-приложений с использованием
технологии сервлетов**

Преподаватель

Репин С.А.

Павловский М.Г.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

1. Цель работы	3
2. Выполнение работы	3
3. Вывод	5
Исходный код сервлета	6
Исходный код web.xml	9
Исходный код pom.xml	9
Приложение А. Файл pom.xml	13

1. ЦЕЛЬ РАБОТЫ

Знакомство с технологией построения Web-приложений на основе сервлетов.

2. ВЫПОЛНЕНИЕ РАБОТЫ

В качестве предметной области взята тема из курсового проекта по объектно-ориентированному программированию на языке Java: разработка средства управления аптеков. Разработан сервлет, отображающий список лекарств в аптеке (ID, имя, количество единиц, цена единицы) и способный фильтровать лекарства по максимальной цене.

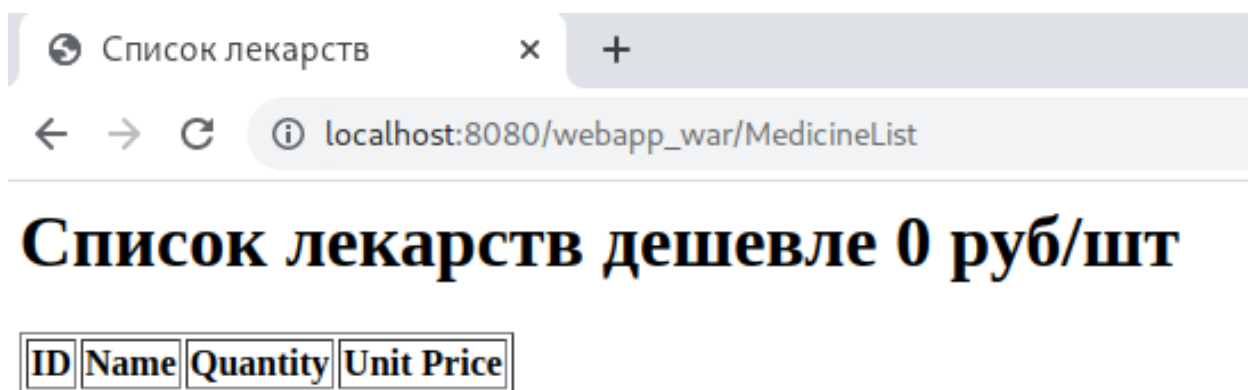


Рис. 2.1 Лекарства дешевле 0 (запрос без параметра)

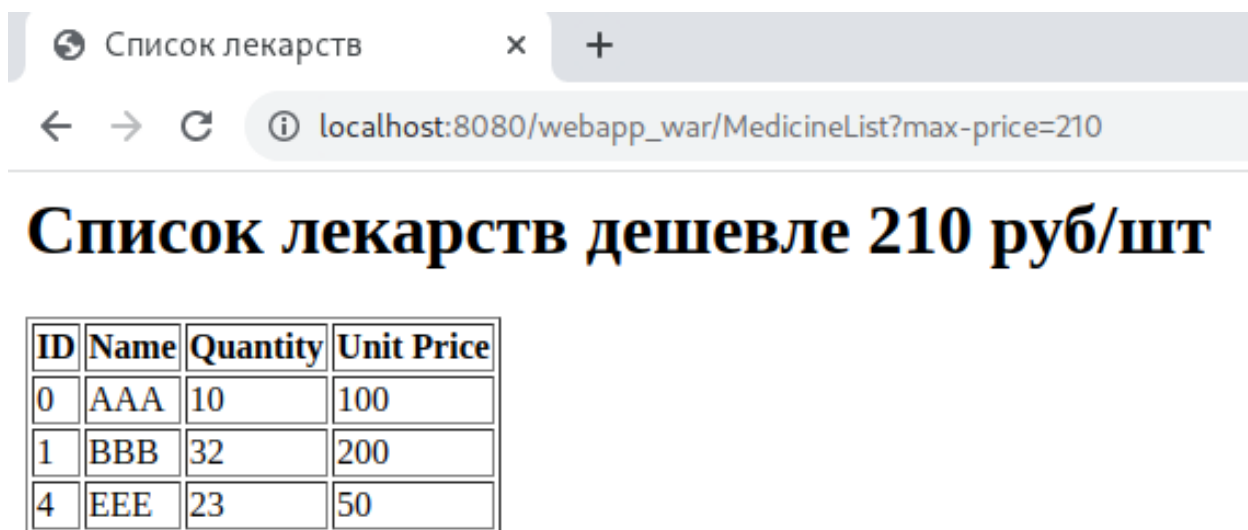
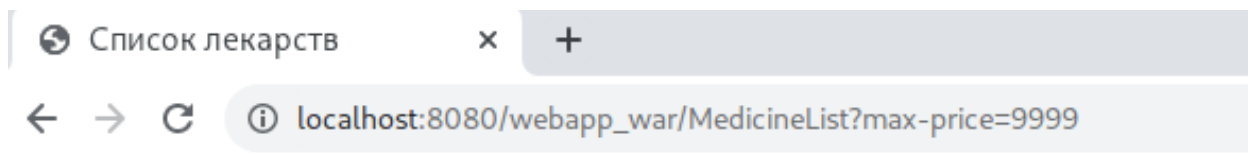


Рис. 2.2 Лекарства дешевле 210



Список лекарств дешевле 9999 руб/шт

ID	Name	Quantity	Unit Price
0	AAA	10	100
1	BBB	32	200
2	CCC	1	220
3	DDD	56	1340
4	EEE	23	50

Рис. 2.3 Все доступные лекарства

Список лекарств x MedicineList (webapp Ma x +

← → ↻ ⓘ localhost:63342/webapp/target/site/apidocs/app/servlet/MedicineList.html

PACKAGE **CLASS** USE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package app.servlet

Class MedicineList

java.lang.Object[ⓘ]
 jakarta.servlet.GenericServlet
 jakarta.servlet.http.HttpServlet
 app.servlet.MedicineList

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

```
@WebServlet(name="MedicineList",
            value="/MedicineList")
public class MedicineList
    extends jakarta.servlet.http.HttpServlet
```

Сервлет, отображающий список лекарств.

See Also:
[Serialized Form](#)

Constructor Summary

Constructors	Description
MedicineList()	Конкструктор.

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	
protected void	doGet (jakarta.servlet.http.HttpServletRequest request, jakarta.servlet.htt	
protected void	doPost (jakarta.servlet.http.HttpServletRequest request, jakarta.servlet.ht	
protected void	processRequest (jakarta.servlet.http.HttpServletRequest request, jakarta.se	

Methods inherited from class jakarta.servlet.http.HttpServlet

Рис. 2.4 Документация, сгенерированная JavaDoc

3. ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки создания простых сервлетов, способных обрабатывать GET и POST

HTTP-запросы с параметрами. Был изучен инструмент JavaDoc для генерации документации.

ИСХОДНЫЙ КОД СЕРВЛЕТА

```
package app.servlet;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;

/**
 * Лекарство в аптеке.
 */
class Medicine {
    public int id;
    public String name;
    public int quantity;
    public int unitPrice;

    /**
     * Конкструктор.
     * @param id ID
     * @param name Имя
     * @param quantity Количество на складе
     * @param unitPrice Цена единицы
     */
    public Medicine(int id, String name, int quantity, int unitPrice) {
        this.id = id;
        this.name = name;
        this.quantity = quantity;
        this.unitPrice = unitPrice;
    }

    /**
     * Возвращает список строковых значений полей класса.
     * @return Список строк
     */
    public List<String> fieldsToStringList() {
        return List.of(
            String.valueOf(this.id),
            this.name,
            String.valueOf(this.quantity),

```

```

        String.valueOf(this.unitPrice)
    );
}

/**
 * Сервлет, отображающий список лекарств.
 */
@WebServlet(name = "MedicineList", value = "/MedicineList")
public class MedicineList extends HttpServlet {
    /**
     * Список лекарств.
     */
    private final List<Medicine> medicines;

    /**
     * Конструктор.
     */
    public MedicineList() {
        super();
        medicines = new ArrayList<>();
        medicines.add(new Medicine(0, "AAA", 10, 100));
        medicines.add(new Medicine(1, "BBB", 32, 200));
        medicines.add(new Medicine(2, "CCC", 1, 220));
        medicines.add(new Medicine(3, "DDD", 56, 1340));
        medicines.add(new Medicine(4, "EEE", 23, 50));
    }

    /**
     * Выполняет GET и POST HTTP-запросы.
     *
     * @param request Запрос к сервлету
     * @param response Ответ сервлета
     * @throws ServletException Внутренняя ошибка
     * @throws IOException Ошибка ввода вывода
     */
    protected void processRequest(HttpServletRequest request,
                                   HttpServletResponse response)
        throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html;charset=UTF-8");

        var max_price_str = request.getParameter("max-price");
        var max_price = (max_price_str == null)
            ? 0
            : Integer.parseInt(max_price_str);
        var table_content = buildTableHtml(max_price);

        try (PrintWriter out = response.getWriter()) {
            out.println(
                "<html>"
                + "<head><title>Список лекарств</title></head>"

```

```

        + "<body>"
            + "<h1>"
                + "Список лекарств дешевле "
                + max_price + " руб/шт"
            + "</h1>"
            + "<table border='1'>"
            + "<tr>"
                + "<td><b>ID</b></td>"
                + "<td><b>Name</b></td>"
                + "<td><b>Quantity</b></td>"
                + "<td><b>Unit Price</b></td>"
            + "</tr>"
            + table_content
            + "</table>"
        + "</body>" +
        "</html>"
    );
}

}

/**
 * Создает содержимое таблицы для всех лекарств, стоимость единицы
 * которых меньше, чем 'max_price'.
 * @param max_price Максимальная цена (не включая)
 * @return HTML-содержимое таблицы
 */
private String buildTableHtml(int max_price) {
    StringBuilder sb = new StringBuilder();
    for (var med : this.medicines) {
        if (med.unitPrice >= max_price) {
            continue;
        }
        sb.append("<tr>");
        for (String field : med.fieldsToStringList()) {
            sb.append("<td>");
            sb.append(field);
            sb.append("</td>");
        }
        sb.append("</tr>");
    }
    return sb.toString();
}

/**
 * Выполняет GET HTTP-запрос.
 *
 * @param request Запрос к сервлету
 * @param response Ответ сервлета
 */
@Override
protected void doGet(HttpServletRequest request,
    HttpServletResponse response)

```



```

        throws ServletException, IOException {
            processRequest(request, response);
        }

    /**
     * Выполняет POST HTTP-запрос.
     *
     * @param request Запрос к сервлету
     * @param response Ответ сервлета
     */
    @Override
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

ИСХОДНЫЙ КОД WEB.XML

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>
    <servlet>
        <servlet-name>BooksList</servlet-name>
        <display-name>BooksList</display-name>
        <servlet-class>app.servlet.MedicineList</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>BooksList</servlet-name>
        <url-pattern>/BooksList</url-pattern>
    </servlet-mapping>
</web-app>

```

ИСХОДНЫЙ КОД POM.XML

```

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

```

```

        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://m
<modelVersion>4.0.0</modelVersion>

<groupId>org.example</groupId>
<artifactId>webapp</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>

<name>webapp Maven Webapp</name>
<url>https://www.example.com</url>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
</properties>

<dependencies>
    <dependency>
        <!--
        Зависимость "Фреймворк JUnit", необходимый для создания различных
        тестов приложения. Полное название: "junit:junit", версия 4.11,
        область действия "test".
        -->
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.2</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>jakarta.platform</groupId>
        <artifactId>jakarta.jakartaee-web-api</artifactId>
        <version>9.0.0</version>
        <scope>provided</scope>
    </dependency>
</dependencies>

<build>
    <finalName>webapp</finalName>
    <pluginManagement>
        <plugins>
            <!--
            Плагин для очистки артефактов сборки (файлов, создаваемых при
            сборке проекта).
            -->
            <plugin>
                <artifactId>maven-clean-plugin</artifactId>
                <version>3.1.0</version>
            </plugin>
            <!--
            Плагин для копирования ресурсов в папку 'target/'.

```

```

-->
<plugin>
  <artifactId>maven-resources-plugin</artifactId>
  <version>3.0.2</version>
</plugin>
<!--
Плагин для компиляции исходников проекта.
-->
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
</plugin>
<!--
Плагин для запуска тестов и генерации отчетов по тестированию.
-->
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.1</version>
</plugin>
<!--
Плагин для сборки всего необходимого для работы приложения в
WAR-архив (архив веб-приложения).
-->
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.2.2</version>
</plugin>
<!--
Плагин для добавления артефактов в локальный репозиторий.
-->
<plugin>
  <artifactId>maven-install-plugin</artifactId>
  <version>2.5.2</version>
</plugin>
<!--
Плагин для копирования артефактов в удаленный репозиторий.
-->
<plugin>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>2.8.2</version>
</plugin>
<!--
Плагин для JavaDoc.
-->
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>3.1.1</version>
  <configuration>
    <javadocExecutable>${java.home}/bin/javadoc</javadocExecutable>
  </configuration>
  <executions>

```

```
        <execution>
          <id>attach-javadocs</id>
          <goals>
            <goal>javadoc</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</pluginManagement>
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <configuration>
      <source>10</source>
      <target>10</target>
    </configuration>
  </plugin>
</plugins>
</build>
</project>
```

ПРИЛОЖЕНИЕ А. ФАЙЛ POM.XML

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>webapp</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>webapp Maven Webapp</name>
  <url>https://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <!--
        Зависимость "Фреймворк JUnit", необходимый для создания различных
        тестов приложения. Полное название: "junit:junit", версия 4.11,
        область действия "test".
      -->
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <finalName>webapp</finalName>
    <pluginManagement>
      <plugins>
        <!--
          Плагин для очистки артефактов сборки (файлов, создаваемых при
          сборке проекта).
        -->
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
```

```

</plugin>
<!--
Плагин для копирования ресурсов в папку 'target/'.
-->
<plugin>
  <artifactId>maven-resources-plugin</artifactId>
  <version>3.0.2</version>
</plugin>
<!--
Плагин для компиляции исходников проекта.
-->
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
</plugin>
<!--
Плагин для запуска тестов и генерации отчетов по тестированию.
-->
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.1</version>
</plugin>
<!--
Плагин для сборки всего необходимого для работы приложения в
WAR-архив (архив веб-приложения).
-->
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.2.2</version>
</plugin>
<!--
Плагин для добавления артефактов в локальный репозиторий.
-->
<plugin>
  <artifactId>maven-install-plugin</artifactId>
  <version>2.5.2</version>
</plugin>
<!--
Плагин для копирования артефактов в удаленный репозиторий.
-->
<plugin>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>2.8.2</version>
</plugin>
</plugins>
</pluginManagement>
</build>
</project>

```