Университет ИТМО

Кафедра ВТ

Программирование Интернет-приложений

Лабораторная работа № 8

Вариант 2526

Работу выполнил:

Студент 2 курса

Группы № 2125

Назарьев Сергей

Санкт-Петербург

2015 г.

**Цель работы:**

Доработать приложение из [лабораторной работы №7](http://helios.cs.ifmo.ru/disciplines/iaps2#lab7) следующим образом:

* **ControllerServlet** по-прежнему должен определять по URL тип ресурса, к которому обращается клиент и делегировать управление соответствующему сервлету или JSP-странице.
* Вместо WebFormServlet нужно разработать **JSP-страницу**; список требований к ней приведён ниже.
* **AreaCheckServlet** по-прежнему должен осуществлять проверку попадания точки в область на координатной плоскости. Результат проверки должен помещаться в контекст приложения, HTTP-сессию или Bean-компонент в соответствии с вариантом задания.

**Разработанная JSP-страница должна содержать:**

1. "Шапку", содержащую ФИО студента, номер группы и номер варианта.
2. Форму, отправляющую данные на сервер.
3. Набор полей для задания координат точки и радиуса области в соответствии с вариантом задания.
4. Сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы.
5. Картинку, изображающую область на координатной плоскости (в соответствии с вариантом задания). Если радиус области установлен, клик курсором мыши по этой картинке должен обрабатываться JavaScript-функцией, определяющей координаты точки, по которой кликнул пользователь. В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.
6. Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения, HTTP-сессии или Bean-компонента в зависимости от варианта.

Разработанное веб-приложение необходимо развернуть на сервере [GlassFish](http://glassfish.java.net).

**Вопросы к защите лабораторной работы:**

1. JavaServer Pages. Особенности, преимущества и недостатки по сравнению с сервлетами, область применения.
2. Жизненный цикл JSP.
3. Структура JSP-страницы. Комментарии, директивы, объявления, скриптлеты и выражения.
4. Правила записи Java-кода внутри JSP. Стандартные переменные, доступные в скриптлетах и выражениях.
5. Bean-компоненты и их использование в JSP.
6. Стандартные теги JSP. Использование Expression Language (EL) в JSP.
7. Параметры конфигурации JSP в дескрипторе развёртывания веб-приложения.

**Код программы:**

::::::::::::::

AreaCheckServlet.java

::::::::::::::

package ru.nazaryev.lab8;

import ru.nazaryev.lab8.ResultBean;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpServletRequest;

public class AreaCheckServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

try {

Double y = (Double)request.getAttribute("y");

Double x = (Double)request.getAttribute("x");

Double r = (Double)request.getAttribute("r");

String result;

Double rh = r/2d;

Boolean res = (x<=0 && y>=-r-x && y <=0)

|| (x<=0 && y>=0 && y\*y+x\*x<=r\*r)

|| (y<=rh && y>=0 && x<=r && x>=0);

result = (res ? "Tochka popala" : "Tochka ne popala");

ResultBean resultBean = (ResultBean) request.getSession().getAttribute("resultBean");

resultBean.addResult(x, y, r, result);

} catch(Exception e) {

}

request.getRequestDispatcher("main.jsp").include(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) {

}

}

::::::::::::::

ControllerServlet.java

::::::::::::::

package ru.nazaryev.lab8;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpServletRequest;

public class ControllerServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String sr = request.getParameter("r");

String sx = request.getParameter("x");

String sy = request.getParameter("y");

if(sr != null && sx != null && sy != null) {

try {

Double r = Double.parseDouble(sr);

Double x = Double.parseDouble(sx);

Double y = Double.parseDouble(sy);

if(x == null || y == null || r == null || x < -4d || x > 4d || r < 1d || r > 3d || y <= -3d || y >= 5d) {

request.setAttribute("error", "Data failed (x [-4..4], r [1..3], y (-3..5)). ");

request.getRequestDispatcher("main.jsp").include(request, response);

return;

}

request.setAttribute("x", x);

request.setAttribute("y", y);

request.setAttribute("r", r);

request.getRequestDispatcher("AreaCheckServlet").include(request, response);

} catch(NumberFormatException e) {

request.setAttribute("error", "Parsing failed.");

request.getRequestDispatcher("main.jsp").include(request, response);

return;

}

} else {

request.getRequestDispatcher("main.jsp").include(request, response);

}

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) {

}

}

::::::::::::::

ResultBean.java

::::::::::::::

package ru.nazaryev.lab8;

import java.util.\*;

public class ResultBean implements java.io.Serializable

{

public List<ResultRow> results = new ArrayList<ResultRow>();

public ResultBean() {

}

public void setResults(List<ResultRow> results) {

this.results = results;

}

public List<ResultRow> getResults() {

return results;

}

public void addResult(double x, double y, double r, String result) {

results.add(new ResultRow(x,y,r,result));

}

}

::::::::::::::

ResultRow.java

::::::::::::::

package ru.nazaryev.lab8;

public class ResultRow implements java.io.Serializable {

public double x;

public double y;

public double r;

public String result;

public ResultRow(double x, double y, double r, String result) {

this.x = x;

this.y = y;

this.r = r;

this.result = result;

}

public double getX() { return this.x; }

public double getY() { return this.y; }

public double getR() { return this.r; }

public String getResult() { return this.result; }

}

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!doctype html>

<html>

<head>

<title>Лабораторная работа</title>

<meta charset=utf8 />

<style>

input {

margin: 10px;

}

html, body {

width: 100%;

padding: 0;

margin: 0;

font-family: Tahoma;

}

.left-block {

float: left;

width: 50%;

}

.right-block {

float: right;

width: 50%;

}

.clear {

clear: both;

}

#header p {

text-align: center;

color: red;

font-size: 24px;

font-family: fantasy, sans-serif;

}

p {

background: yellow;

}

</style>

<jsp:useBean id="resultBean" class="ru.nazaryev.lab8.ResultBean" scope="session"></jsp:useBean>

</head>

<body>

<div id="header">

<p>Назарьев Сергей Сергеевич, гр. 2125, вариант 2526</p>

<p>${error}</p>

</div>

<form method=get name="myForm" onsubmit="return(validate());">

<img id="click" src="areas.png" />

<div class="left-block">

<select name="x">

<option value="-4">-4</option>

<option value="-3">-3</option>

<option value="-2">-2</option>

<option value="-1">-1</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

</select>

<input type="text" name="y" placeholder="y" required>

<input type="submit" name="submit" value="Send">

<table border=1 style="width: 100%;">

<tr><th>X</th><th>Y</th><th>R</th><th></th></tr>

<c:forEach var="item" items="${resultBean.results}">

<tr>

<td><c:out value="${item.x}"/></td>

<td><c:out value="${item.y}"/></td>

<td><c:out value="${item.r}"/></td>

<td><c:out value="${item.result}"/></td>

</tr>

</c:forEach>

</table>

</div>

<div class="right-block">

<input checked type="radio" name="r" id="r1" value="1">

<label for="r1">R = 1</label>

<input type="radio" name="r" id="r2" value="1.5">

<label for="r2">R = 1.5</label>

<input type="radio" name="r" id="r3" value="2">

<label for="r3">R = 2</label>

<input type="radio" name="r" id="r4" value="2.5">

<label for="r4">R = 2.5</label>

<input type="radio" name="r" id="r5" value="3">

<label for="r5">R = 3</label>

</div>

<div class="clear"></div>

</form>

<script>

var cx = 110, cy = 110;

var rl = 80;

function isNumber(n) {

return !isNaN(parseFloat(n)) && isFinite(n);

}

function validate()

{

var y = document.getElementsByName("y")[0].value;

if(!isNumber(y) || y >= 5 || y <= -3)

{

alert( "y должен попадать в (-3, 5)");

return false;

}

return true ;

}

function clicker(e)

{

R = document.querySelector('input[name="r"]:checked').value;

rect = clk.getBoundingClientRect();

ax = e.clientX - rect.left;

ay = e.clientY - rect.top;

x = ax - cx;

y = cy - ay;

var ourX = x \* R / rl;

var ourY = y \* R / rl;

alert("X: " + ourX + "; Y = " + ourY);

}

var clk = document.getElementById("click");

clk.addEventListener("mousedown", clicker);

</script>

</body>

</html>

**Вывод:**

В ходе данной лабораторной работы были изучены принципы построения JSP-страниц и разработка JavaBeans.