СПб НИУ ИТМО

кафедра ИПМ

Программирование интернет-приложений

Лабораторная работа № 8

JSP-страница

Вариант 21262

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2014 г.

**Цель работы:**

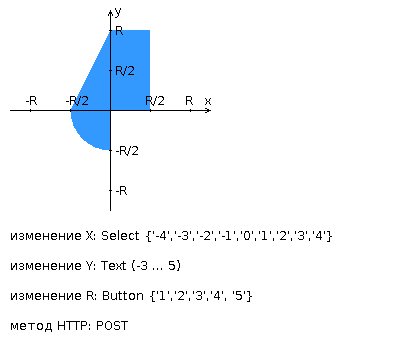
Доработать приложение из лабораторной работы №7 следующим образом:

* **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.

**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.

Разработанное веб-приложение необходимо развернуть на сервере GlassFish.



**Код Программы:**

@WebServlet(name = "**ControllerServlet**", urlPatterns = {"/ControllerServlet"})

public class ControllerServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

getServletContext().getRequestDispatcher("/WebForm.jsp").forward(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

getServletContext().getRequestDispatcher("/AreaCheckServlet").forward(request, response);

} }

**WebForm.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.util.ArrayList"%>

<!DOCTYPE html>

<html>

<head>

<title>WebLab № 8</title>

<script>

var submit\_coord = false;

function validForm(f) {

value = parseFloat(f.inputY.value);

if (submit\_coord) {

if ((f.coordX.value === "") && (f.coordY.value === ""))

{ alert('Сначала кликните точку на графике!');

} else f.submit(); }

else if (isNaN(f.inputY.value \* 1) || f.inputY.value.length === 0) {

alert('Неверный формат Y!'); }

else if (value < -3 || value > 3) {

alert('Y не попадает в заданный диапазон (-3...3)'); }

else { f.submit(); } }

function clickX(id) {

submit\_coord = false;

var X = document.getElementById(id).value;

document.getElementById("inputX").value = X; }

function clickSubmit() {

submit\_coord = true;

document.getElementById("inputX").value = ""; }

function getchar(event) {

var str = String.fromCharCode(event.charCode);

return event.char === 0 || /[-0-9.]/.test(str) || event.charCode === 0; }

function getcoord(e) {

var R = document.getElementById("inputR").value;

var x, y;

var c = document.getElementById("svg");

if (R < 1 || R > 3 || isNaN(R \* 1)) {

alert("Сначала укажите верный радиус!"); }

else {

if (e.hasOwnProperty('offsetX'))

{ x = e.offsetX;

y = e.offsetY; }

else

{ x = e.layerX - c.offsetLeft - c.offsetParent.offsetLeft - c.offsetParent.offsetParent.offsetLeft;

y = e.layerY - c.offsetTop - c.offsetParent.offsetTop - c.offsetParent.offsetParent.offsetTop; }

pos\_x = (x - 250) / 200 \* R;

pos\_y = (250-y) / 200 \* R;

document.getElementById("coordX").value = String(Math.round(pos\_x \* 100) / 100);

document.getElementById("coordY").value = String(Math.round(pos\_y \* 100) / 100); } }

</script>

</head>

<body>

<form method="POST" action="ControllerServlet" onSubmit="validForm(this); return false">

<H1>Журавлев В., гр.2120, вар.21262</H1>

<table>

<tr>

<td>

<svg id="svg" width="500" height="500" onclick="getcoord(event)">

<polygon points="250,250 50,250 250,50" style="fill:blue"/>

<path d="M250,50 A 200,200 0 0,1 450,250 L 250,250" style="fill:blue"></path>

<rect width="100" height="200" x="250" y="250" style="fill:blue"/>

<line x1="250" y1="0" x2="250" y2="500" stroke="black" stroke-width="2"/>

<line x1="0" y1="250" x2="500" y2="250" stroke="black" stroke-width="2"/>

</svg>

</td>

<td valign="top">

<div>

<label>R: </label>

<select name="inputR" id="inputR">

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

<option value="1.5">1,5</option>

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

<option value="2.5">2,5</option>

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

</select>

</div>

<br>

<div>

<label>X: </label>

<input type="Submit" id="X1" value="-4" onclick="clickX(this.id)">

<input type="Submit" id="X2" value="-3" onclick="clickX(this.id)">

<input type="Submit" id="X3" value="-2" onclick="clickX(this.id)">

<input type="Submit" id="X4" value="-1" onclick="clickX(this.id)">

<input type="Submit" id="X5" value="0" onclick="clickX(this.id)">

<input type="Submit" id="X6" value="1" onclick="clickX(this.id)">

<input type="Submit" id="X7" value="2" onclick="clickX(this.id)">

<input type="Submit" id="X8" value="3" onclick="clickX(this.id)">

<input type="Submit" id="X9" value="4" onclick="clickX(this.id)">

<input type="text" name="inputX" id="inputX" hidden>

</div>

<div>

<label>Y: </label>

<input type="text" name="inputY" id="inputY" onkeypress="return getchar(event)">

</div>

<br><br>

<div>

<label>Clicked X: </label>

<input type="text" name="coordX" id="coordX" value="" readonly>

</div>

<div>

<label>Clicked Y: </label>

<input type="text" name="coordY" id="coordY" value="" readonly>

</div>

<div><input type="Submit" name="Submit" onclick="clickSubmit()"><div>

</td>

</tr>

</table>

</form>

<%@page import='device.Point'%>

<% ServletContext context = getServletContext();

ArrayList<Point> ar = (ArrayList) context.getAttribute("data");

if (ar != null) {

out.print("<table border=\"2\" style=\"width:38%\">\n"

+ "<tr>\n"

+ " <th>X</th>\n"

+ " <th>Y</th> \n"

+ " <th>R</th>\n"

+ " <th>Result</th>\n"

+ "</tr>\n");

if (ar != null && !ar.isEmpty()) {

for (int i = 0; i < ar.size(); i++) {

out.print("<tr>\n"

+ " <td>" + (ar.get(i)).x + "</td>\n"

+ " <td>" + (ar.get(i)).y + "</td>\n"

+ " <td>" + (ar.get(i)).r + "</td>\n"

+ " <td>" + (ar.get(i)).result + "</td>\n"

+ "</tr>\n"); } }

out.print("</table>"); } %>

</body>

</html>

@WebServlet(name = "**AreaCheckServlet**", urlPatterns = {"/AreaCheckServlet"})

public class AreaCheckServlet extends HttpServlet {

public static ArrayList<Point> ar = new ArrayList();

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

float x, y, r;

try { x = Float.valueOf(request.getParameter("inputX"));

y = Float.valueOf(request.getParameter("inputY"));

} catch (Exception e) {

x = Float.valueOf(request.getParameter("coordX"));

y = Float.valueOf(request.getParameter("coordY")); }

r = Float.valueOf(request.getParameter("inputR"));

boolean result = check(x, y, r);

Point p = new Point(x, y, r, result);

ar.add(p);

getServletContext().setAttribute("data", ar);

request.getRequestDispatcher("/ControllerServlet").forward(request, response); }

static boolean check(float x, float y, float r) {

return (x >= 0) && (y >= 0) && (x \* x + y \* y <= r \* r) || ((x >= 0) && (y <= 0) && (x <= r / 2) && (y >= -r)) || ((x <= 0) && (y >= 0) && (y <= x + r)); } }

public class **Point** {

public float x, y, r; public boolean result;

public Point(float x, float y, float r, boolean result) {

this.x = x; this.y = y; this.r = r; this.result = result; } }

**Вывод:**

В ходе выполнения лабораторной работы я изучил некоторые правила работы c JSP-файлами: синтаксис, скриплеты и выражения, написание кода Java внутри JSP.

В том числе я освоил такое понятие, как контекст приложения, в котором, в данном случае хранится таблица результатов выполнения приложения и делегирование ее соответствующему JSP-файлу.

Так же я познакомился с работой с изображениями (графиками), их рисованием и получением координат точки на рисунке, по которой кликнул пользователь и последующей отправкой и обработкой полученных значений.