

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

**Дисциплина «Программирование интернет-приложений»
Лабораторная работа №2**

Вариант 772

Выполнил:

Съестов Дмитрий Вячеславович
Группа Р3217

Преподаватель:

Николаев Владимир Вячеславович

Санкт-Петербург
2017

Задание

Разработать веб-приложение на базе сервлетов и JSP, определяющее попадание точки на координатной плоскости в заданную область. Приложение должно быть реализовано в соответствии с шаблоном MVC и состоять из следующих элементов:

- **ControllerServlet**, определяющий тип запроса, и, в зависимости от того, содержит ли запрос информацию о координатах точки и радиусе, делегирующий его обработку одному из перечисленных ниже компонентов.
- **AreaCheckServlet**, осуществляющий проверку попадания точки в область на координатной плоскости и формирующий HTML-страницу с результатами проверки. Должен обрабатывать все запросы, содержащие сведения о координатах точки и радиусе области. Все запросы внутри приложения должны передаваться этому сервлету по методу GET, остальные сервлеты с веб-страниц напрямую вызываться не должны.
- Страница JSP, формирующая HTML-страницу с веб-формой. Должна обрабатывать все запросы, не содержащие сведений о координатах точки и радиусе области. Разработанная страница JSP должна содержать:
 - "Шапку", содержащую ФИО студента, номер группы и номер варианта.
 - Форму, отправляющую данные на сервер.
 - Набор полей для задания координат точки и радиуса области в соответствии с вариантом задания.
 - Сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы.
 - Интерактивный элемент, содержащий изображение области на координатной плоскости (в соответствии с вариантом задания) и реализующий следующую функциональность:
 - Если радиус области установлен, клик курсором мыши по изображению должен обрабатываться JavaScript-функцией, определяющей координаты точки, по которой кликнул пользователь и отправляющей полученные координаты на сервер для проверки факта попадания.
 - В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.
 - После проверки факта попадания точки в область изображение должно быть обновлено с учётом результатов этой проверки (т.е., на нём должна появиться новая точка).
 - Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения.

ControllerServlet (Controller)

```
public class ControllerServlet extends HttpServlet
{
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        RequestDispatcher dispatcher;

        String rString = request.getParameter("r"),
            xString = request.getParameter("x"),
            yString = request.getParameter("y");

        float r;
        try
        {
            assert (rString != null && xString != null && yString != null);
            r = Float.parseFloat(rString);
            Float.parseFloat(xString);
            Float.parseFloat(yString);
            assert (r >= 0);
        }
        catch (Exception exception)
        {
            dispatcher = request.getRequestDispatcher("/index.jsp");
            dispatcher.forward(request, response);
            return;
        }

        dispatcher = request.getRequestDispatcher("/check");
        dispatcher.forward(request, response);
    }
}
```

AreaCheckServlet (Model)

```
public class AreaCheckServlet extends HttpServlet
{
    private boolean checkArea(float r, Point p) throws IOException
    {
        Point center = new Point(0, 0);

        //четверть окружности радиуса R в I четверти
        if (p.x > 0 && p.y > 0 && Utils.distance(center, p) < r)
            return true;

        //Прямоугольник R/2 x R в IV четверти
        if (p.x > 0 && p.x < r / 2 && p.y < 0 && p.y > -r)
            return true;

        ///треугольник в II четверти
        Point a = new Point(-r, 0), b = new Point(0, r);
        double abc = Utils.triangleArea(a, b, center),
            abp = Utils.triangleArea(a, b, p),
            acp = Utils.triangleArea(a, p, center),
            bcp = Utils.triangleArea(p, b, center);

        return (abp != 0 && acp != 0 && bcp != 0 && Utils.doubleEquals(abc, abp + acp + bcp));
    }
}
```

```

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    float r = Float.parseFloat(request.getParameter("r"));
    float x = Float.parseFloat(request.getParameter("x"));
    float y = Float.parseFloat(request.getParameter("y"));

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

    PrintWriter out = response.getWriter();
    //если это AJAX-запрос, возвращаем yes/no
    boolean ajax = "XMLHttpRequest".equals(request.getHeader("X-Requested-With"));
    if(ajax)
    {
        response.setContentType("text");
        out.print(result ? "Yes" : "No");
        return;
    }

    //выводим страницу
    response.setContentType("text/html");
    out.print("<!DOCTYPE html>" +
        "<html lang=\"en\">" +
        "<style type=\"text/css\">" +
        "div {" +
        "    margin: 5% auto 5%;" +
        "}" +
        "table {" +
        "    border: outset black;" +
        "    background-color: white;" +
        "    margin: inherit;" +
        "    text-align: center;" +
        "}" +
        "</style>" +
        "<head>" +
        "    <meta charset=\"UTF-8\">" +
        "    <title>Results</title>" +
        "</head>" +
        "<body>" +
        "    <div>" +
        "        <table border=\"1\">" +
        "            <tr>" +
        "                <th>Radius</th>" +
        "                <th>X</th>" +
        "                <th>Y</th>" +
        "                <th>Included?</th>" +
        "            </tr>" +
        "            <tr>" +
        "                <td>" + r + "</td>" +
        "                <td>" + x + "</td>" +
        "                <td>" + y + "</td>" +
        "                <td>" + (result ? "Yes" : "No") + "</td>" +
        "            </tr>" +
        "        </table>" +
        "    </div>" +
        "    <div style=\"text-align: center\">" +

```

```

        "        <button onclick=\"location.href='./';\">Return</button>" +
        "    </div>" +
        "</body>" +
        "</html>");
out.close();

//сохраняем данные в контекст
ServletContext context = getServletContext();

ArrayList<Float> rList = (ArrayList)context.getAttribute("r");
ArrayList<Float> xList = (ArrayList)context.getAttribute("x");
ArrayList<Float> yList = (ArrayList)context.getAttribute("y");
ArrayList<Boolean> resList = (ArrayList)context.getAttribute("result");

if(rList == null || xList == null || yList == null || resList == null)
{
    rList = new ArrayList<Float>(10);
    xList = new ArrayList<Float>(10);
    yList = new ArrayList<Float>(10);
    resList = new ArrayList<Boolean>(10);
}

rList.add(r);
xList.add(x);
yList.add(y);
resList.add(result);

context.setAttribute("r", rList);
context.setAttribute("x", xList);
context.setAttribute("y", yList);
context.setAttribute("result", resList);
}
}

```

JSP-страница (View)

```

<%@ page import="java.util.ArrayList" %>
<style> <@include file="/style.css" %> </style>
<script src="https://code.jquery.com/jquery-2.2.3.min.js"></script>
<!DOCTYPE html>
<html lang="en">
<meta charset="UTF-8">
<head>
    <title>Laboratory work #2</title>
</head>
<body>
    <header id="index_header">
        <h1>Laboratory work #2</h1>
        <h2>Variant 772</h2>
        <h3>Dmitry Sjestov</h3>
        <h3>Group P3217</h3>
        <h3>Teacher: V.V.Nikolaev</h3>
    </header>

    

    <div class="outer" id="main_block">
        <div class="inner" id="canvas_block">

```

```

<canvas id="canvas" width="705px" height="705px"> </canvas>
<script>
    var canvas = document.getElementById("canvas"),
        context = canvas.getContext("2d");
    context.font = "24px Arial";
    context.textAlign = "center";
    const center = 353, cellSize = 70.5, dotRadius = 7;

    function drawArea(r) {
        context.clearRect(0, 0, cellSize * 10, cellSize * 10);
        const areaSize = cellSize * r ^ 0;
        var radiusDefined = typeof(r) !== "undefined";
        if (radiusDefined) {
            context.fillStyle = "#FFCC00";
            context.beginPath();
            context.moveTo(center - areaSize, center);
            context.lineTo(center, center);
            context.arc(center, center, areaSize, 0, Math.PI * 1.5, true);
            context.fill();
            context.closePath();
            context.fillRect(center, center, areaSize / 2 ^ 0, areaSize);
        }
        if (radiusDefined) drawCoordinates(r, areaSize);
        drawAxis();
        drawPoints(r);
    }

    function drawAxis() {
        var axis = document.getElementById("axis");
        context.drawImage(axis, 0, 0);
    }

    function drawCoordinates(r, areaSize) {
        context.fillStyle = "Black";
        var offset = r === 5 ? 10 : 0;
        context.fillText(r, center - areaSize + offset, center - 20);
        context.fillText(r, center + areaSize - offset, center - 20);
        context.fillText(r, center + 20, center - areaSize + 2 * offset);
        context.fillText(r, center - 20, center + areaSize);
        context.fillText((r * 0.5).toString(), center + areaSize / 2 ^ 0, center +
30);
    }

    function drawPoints(r) {
        $('table > tbody > tr').each(function(index, element) {
            var x = parseFloat(element.cells[1].innerHTML),
                y = parseFloat(element.cells[2].innerHTML);
            $.ajax({ method: "GET", url: "/check?r=" + r + "&x=" + x + "&y=" + y })
                .always(function (data) {
                    var result = data === "Yes";
                    context.fillStyle = (result ? "Green" : "Red");
                    context.beginPath();
                    context.arc(center + x * cellSize ^ 0, center - y * cellSize ^ 0,
dotRadius, 0, 2 * Math.PI);
                    context.fill();
                    context.closePath();
                });
        });
    }
}

```

```

function updateCanvas() {
    var input = document.forms["form"]["r_field"].value;
    var r = parseInt(input);
    var radiusDefined = input.length === 1 && r >= 1 && r <= 5;
    drawArea(radiusDefined ? r : undefined);
}

function getMousePos(canvas, e) {
    var rect = canvas.getBoundingClientRect();
    return {
        x: e.clientX - rect.left,
        y: e.clientY - rect.top
    };
}

canvas.addEventListener("click", canvasClickEvent, false);
function canvasClickEvent(e) {
    var input = document.forms["form"]["r_field"].value;
    var r = parseInt(input);
    if (input.length === 1 && r >= 1 && r <= 5)
    {
        var coordinates = getMousePos(canvas, e);
        document.forms["form"]["x_field"].value = (coordinates.x - center) /
cellSize;

        document.forms["form"]["y_field"].value = (center - coordinates.y) /
cellSize;

        document.forms["form"].submit();
    }
    else alert("Unable to get coordinates")
}
}
</script>
</div>

<div class="inner" style="padding-left: 150px">
    <div id="validation_block">
        <form name="form" action="./check" onsubmit="return validate()" method="get">
            <p id="info_field">Parameters:</p>
            R: <input type="text" id="r_field" name="r" oninput="updateCanvas();">
            X: <input type="text" id="x_field" name="x">
            Y: <input type="text" id="y_field" name="y">
            <input type="submit" value="Send">
        </form>
        <script>
            function setInfoText(description) {
                document.getElementById("info_field").innerHTML = description;
                return false;
            }

            function validate() {
                const int_regex = /^0|-?[1-9]\d*$/;
                const real_regex = /^0|-?(?:([1-9]\d*(?:\.\d+)?)|0\.\d+)$/;

                if (!int_regex.test(document.forms["form"]["r_field"].value))
                    return setInfoText("Incorrect input in radius field");

                if (!real_regex.test(document.forms["form"]["x_field"].value))
                    return setInfoText("Incorrect input in X field");
            }
        </script>
    </div>
</div>

```

```

        if (!real_regex.test(document.forms["form"]["y_field"].value))
            return setInfoText("Incorrect input in Y field");

        var r = parseInt(document.forms["form"]["r_field"].value);
        if (r < 1 || r > 5) return setInfoText("Radius must be an integer
between 1 and 5");

        var x = parseFloat(document.forms["form"]["x_field"].value);
        if (x < -5 || x > 5) return setInfoText("X must be between -5 and 5");

        var y = parseFloat(document.forms["form"]["y_field"].value);
        if (y < -5 || y > 5) return setInfoText("Y must be between -5 and 5");
    }
</script>
</div>
</div>
</div>
<div>
    <table id="table" border="1">
        <thead>
            <tr>
                <th>Radius</th>
                <th>X</th>
                <th>Y</th>
                <th>Included?</th>
            </tr>
        </thead>

        <tbody>
            <%
                ServletContext context = session.getServletContext();
                try {
                    ArrayList<Float> r = (ArrayList) context.getAttribute("r"),
                        x = (ArrayList) context.getAttribute("x"),
                        y = (ArrayList) context.getAttribute("y");
                    ArrayList<Boolean> result = (ArrayList) context.getAttribute("result");
                    int count = r.size();
                    for (int i = 0; i < count; i++) {
                        String resultStr = result.get(i) ? "Yes" : "No";
                        out.println("<tr>" +
                            "    <td>" + r.get(i) + "</td>" +
                            "    <td>" + x.get(i) + "</td>" +
                            "    <td>" + y.get(i) + "</td>" +
                            "    <td class=\"" + resultStr + "Row\">" + resultStr +
"</td>" +
                            "</tr>");
                    }
                }
                catch (Exception e) {}
            %>
        </tbody>
    </table>
</div>

<div style="text-align: center">
    <button onclick="clearHistory()" style="margin: auto">Clear history</button>

```



```
<script>
    function clearHistory() {
        var table = document.getElementById("table");
        while(table.rows.length > 1) table.deleteRow(1);
        $.ajax({ method: "GET", url: "/clear" });
        updateCanvas();
    }
</script>
</div>
</body>
</html>
```

Вывод

В ходе выполнения данной работы я научился разрабатывать веб-приложения на основе сервлетов, соответствующие модели MVC, а также обрабатывать события с помощью Javascript и выполнять AJAX-запросы.