Университет ИТМО

Кафедра вычислительной техники

Программирование интернет-приложений

Лабораторная работа № 9

Вариант 1758

Выполнил:

Сорокин Юрий

Группа Р3217

Преподаватель:

Харитонова А.Е.

СПб

2016 г.

### Лабораторная работа #9

### Внимание! У разных вариантов разный текст задания!

Разработать приложение на базе JavaServer Faces Framework, которое осуществляет проверку попадания точки в заданную область на координатной плоскости.

Приложение должно включать в себя 2 facelets-шаблона - стартовую страницу и основную страницу приложения, а также набор управляемых бинов (managed beans), реализующих логику на стороне сервера.

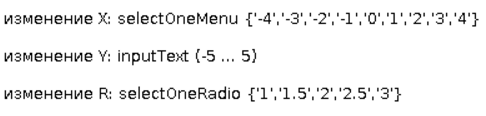
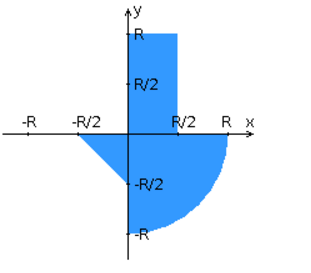
**Стартовая страница должна содержать следующие элементы:**

* "Шапку", содержащую ФИО студента, номер группы и номер варианта.
* Интерактивные часы, показывающие текущие дату и время, обновляющиеся раз в 7 секунд.
* Ссылку, позволяющую перейти на основную страницу приложения.

**Основная страница приложения должна содержать следующие элементы:**

* Набор компонентов для задания координат точки и радиуса области в соответствии с вариантом задания. Может потребоваться использование дополнительных библиотек компонентов - [ICEfaces](http://www.icesoft.org/java/projects/ICEfaces/overview.jsf" \t "_blank) (префикс "ace") и [PrimeFaces](http://www.primefaces.org/" \t "_blank)(префикс "p"). Если компонент допускает ввод заведомо некорректных данных (таких, например, как буквы в координатах точки или отрицательный радиус), то приложение должно осуществлять их валидацию.
* Динамически обновляемую картинку, изображающую область на координатной плоскости в соответствии с номером варианта и точки, координаты которых были заданы пользователем. Клик по картинке должен инициировать сценарий, осуществляющий определение координат новой точки и отправку их на сервер для проверки её попадания в область. Цвет точек должен зависить от факта попадания / непопадания в область. Смена радиуса также должна инициировать перерисовку картинки.
* Таблицу со списком результатов предыдущих проверок.
* Ссылку, позволяющую вернуться на стартовую страницу.

**Дополнительные требования к приложению:**

* Для хранения списка результатов должен использоваться Session-scoped Managed Bean.
* Конфигурация управляемых бинов должна быть задана с помощью аннотаций.
* Правила навигации между страницами приложения должны быть заданы в отдельном конфигурационном файле.

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ page isELIgnored="false" %>

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

<jsp:useBean id="result" class="hz.ResultBean" scope="session" />

<html>

<head>

<meta charset="UTF-8">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">

</script>

<title>ПИП Лаба №8</title>

<style>

<%@include file="style.css" %>

</style>

</head>

<body>

<script>

<%@include file="script.js"%>

</script>

<div class="wrapper">

<header>

<p>Сорокин Юрий Борисович</p>

<p>Группа P3217</p>

<p>Вариант 1754</p>

</header>

<div class="content">

<div class="inputs">

<form method="GET" id="forma" action="../lab8/main">

<p>X value:<select name="XSelector">

<option>-4</option>

<option>-3</option>

<option>-2</option>

<option>-1</option>

<option>0</option>

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

</select></p>

<p>Y value {-3..3}:

<input type="text" id="textY" name="YSelector" placeholder="-3..3" onchange="validateForm()" onkeydown="javascript:if(13==event.keyCode){return false;}" required>

</p>

<p>R value:

<input type="checkbox" name="RSelector1" id="RSelector1" onclick="set(1)" value="1">1

<input type="checkbox" name="RSelector2" id="RSelector2" onclick="set(2)" value="1.5">1.5

<input type="checkbox" name="RSelector3" id="RSelector3" onclick="set(3)" value="2">2

<input type="checkbox" name="RSelector4" id="RSelector4" onclick="set(4)" value="2.5">2.5

<input type="checkbox" name="RSelector5" id="RSelector5" onclick="set(5)" value="3">3

</p>

<input type="hidden" id="hiddenR" name="hiddenR" value="" disabled="true">

<input type="hidden" id="hiddenY" name="hiddenY" value="" disabled="true">

<button type="submit" id="ok" name="submit" disabled="true">OK</button>

</form>

</div>

<div class="area">

<p><img name='img' src="http://i.imgur.com/cCdmLea.png" id="graph" style="position: relative;"/></p>

</div>

</div>

<c:if test="${result.results!=null}">

<table>

<tr>

<td colspan="4">Результат обработки формы</td>

</tr>

<tr>

<th>Координата X</th>

<th>Координата Y</th>

<th>Радиус R</th>

<th>Результат</th>

</tr>

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

<tr>

<td>${result.x}</td>

<td>${result.y}</td>

<td>${result.r}</td>

<td>${result.isEntered}</td>

</tr>

</c:forEach>

</table>

</c:if>

</div>

</body>

</html>

public class AreaCheckServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

req.setCharacterEncoding("UTF-8");

resp.setCharacterEncoding("UTF-8");

String XBuf = req.getParameter("XSelector").replace(',', '.');

String YBuf = req.getParameter("YSelector").replace(',', '.');

ArrayList<String> RBuf = new ArrayList<String>();

for(int i=1;i<=5;i++) if (req.getParameter("RSelector"+i) != null) RBuf.add(req.getParameter("RSelector"+i));

float X = Float.NaN;

float Y = Float.NaN;

float R = Float.NaN;

int size = RBuf.size();

try {

X = Float.parseFloat(XBuf);

Y = Float.parseFloat(YBuf);

if(RBuf.size()==1) R = Float.parseFloat(RBuf.get(0).replace(',','.'));

if (!validate(X,Y,R))throw new NumberFormatException();

} catch (NumberFormatException e) {}

ResultBean resultsBean = (ResultBean)req.getSession().getAttribute("result");

if (resultsBean == null) {

resultsBean = new ResultBean();

}

ArrayList<Result> resultsList = resultsBean.getResults();

resultsList.add(new Result(X,Y,R, inFigure(X, Y, R)));

resultsBean.setResults(resultsList);

req.getSession().setAttribute("result", resultsBean);

RequestDispatcher rd = req.getRequestDispatcher("index.jsp");

rd.forward(req, resp);

}

boolean inFigure(float X, float Y, float R) {

return (Y <=0 && Y>=-R && X>=-R/2 && X<=0) || //left bot

(Y >= 0 && X >= 0 && (X \* X + Y \* Y) <= R\*R/4) || // right top

(Y>=0 && X<=0 && Y<=X+R); //left top

}

boolean validate(float X, float Y, float R) {

return (!Float.isNaN(X) && !Float.isNaN(Y) && !Float.isNaN(R) && Y <= 3 && Y >= -3 && (R==1 || R==1.5 || R==2 || R==2.5 || R==3) && (X==-4 || X==-3 || X==-2 || X==-1 || X==0 || X==4 || X==3 || X==2 || X==1) );

}

}

public class Result {

private double x;

private double y;

private double r;

private boolean isEntered;

public Result(double x, double y, double r, boolean isEntered) {

this.x = x;

this.y = y;

this.r = r;

this.isEntered = isEntered;

}

@Override

public boolean equals(Object obj) {

if (this == obj) return true;

if (obj == null || getClass() != obj.getClass()) return false;

Result res = (Result) obj;

return (this.x == res.x && this.y == res.y && this.r == res.r && this.isEntered == res.isEntered);

}

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

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

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

public boolean getIsEntered() { return this.isEntered; }

}

public class ResultBean implements Serializable {

ArrayList<Result> results;

String test;

public String getTest() {

return this.test;

}

public void setTest(String test) {

this.test = test;

}

public ResultBean() {

this.results = new ArrayList<Result>();

}

public ArrayList<Result> getResults() {

return this.results;

}

public void setResults(ArrayList<Result> results) {

this.results = results;

}

@Override

public boolean equals(Object obj) {

if (this == obj) return true;

if (obj == null || getClass() != obj.getClass()) return false;

ResultBean bean = (ResultBean) obj;

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

if (this.results.get(i) != bean.results.get(i)) return false;

}

return true;

}

@Override

public int hashCode() {

return results.size();

}

@Override

public String toString() {

return "Массив из " + results.size() + " результатов";

}

}

**Вывод**

При выполнении данной работы, я ознакомился созданием JSP страниц.