**ИТМО Кафедра Вычислительной техники**

Отчет по лабораторной работе №7

Программирование интернет-приложений

Вариант 2911

**Выполнил: студент группы P3217**

**Плюхин Дмитрий**

**Преподаватель: Гаврилов А. В.**

**2017 год**

1. **Задание к лабораторной работе**

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. Все запросы внутри приложения должны передаваться этому сервлету (по методу GET или POST в зависимости от варианта задания), остальные сервлеты с веб-страниц напрямую вызываться не должны.
* 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.
* Страница JSP, формирующая HTML-страницу с веб-формой. Должна обрабатывать все запросы, не содержащие сведений о координатах точки и радиусе области.
* Метод HTTP: POST
* Список результатов сохранять в Bean-компоненте

Разработанная страница JSP 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. Интерактивный элемент, содержащий изображение области на координатной плоскости (в соответствии с вариантом задания) и реализующий следующую функциональность:
   1. 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.
   2. В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.
   3. После проверки факта попадания точки в область изображение должно быть обновлено с учётом результатов этой проверки (т.е., на нём должна появиться новая точка).
6. Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения, HTTP-сессии или Bean-компонента в зависимости от варианта.

Страница, возвращаемая AreaCheckServlet, должна содержать:

1. Таблицу, содержащую полученные параметры.
2. Результат вычислений - факт попадания или непопадания точки в область.
3. Ссылку на страницу с веб-формой для формирования нового запроса.

Разработанное веб-приложение необходимо развернуть на сервере GlassFish.

1. **Исходный код**

**//Файл script.js**

function isKeyIncorrect(keyCode,symbol){

return (keyCode != 189 && keyCode != 8 && keyCode != 39 && keyCode != 37 && symbol != '1' && symbol != '2' && symbol != '3' && symbol != '4' &&

symbol != '5' && symbol != '6' && symbol != '7' && symbol != '8' && symbol != '9' && symbol != '0' && symbol != '.')

}

function isValueIncorrect(value,keyCode,symbol){

return ((keyCode == 189) && (value.length > 0)) || ((symbol == '.') && (value.indexOf('.') + 1));

}

function isNumberInorrect(keyCode,number){

return ((keyCode >= 48 && keyCode <= 57) || (keyCode == 190)) && ((number <= -5.0) || (number >= 5.0));

}

function isInArea(x,y,radius,dot,cell){

var xhr = new XMLHttpRequest();

var body = 'x=' + encodeURIComponent(x) +'&y=' + encodeURIComponent(y) + "&radius=" + encodeURIComponent(radius);

console.log(body);

xhr.open("POST", '/pip.lab2/home', true)

xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')

xhr.onreadystatechange = function () {

if(xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {

console.log(xhr.responseText);

if(xhr.responseText.indexOf("OUT") != -1){

dot.style.backgroundColor = "red";

cell.innerHTML = "OUT";

return false;

};

};

};

xhr.send(body);

return true;

}

function fillGraph(){

tablepontos = document.getElementById("pontos").childNodes[0];

radius = $("[name = radius]")[0].value\*1;

for (childItem in tablepontos.childNodes) {

if (tablepontos.childNodes[childItem].nodeType == 1)

if (tablepontos.childNodes[childItem].childNodes != undefined){

if (tablepontos.childNodes[childItem].childNodes[0].innerHTML == "x") continue;

dot = document.createElement('div');

dot.className = "dot";

console.log(tablepontos.childNodes[childItem]);

ml = (tablepontos.childNodes[childItem].childNodes[0].innerHTML\*80/radius+101);

mt = (-tablepontos.childNodes[childItem].childNodes[1].innerHTML\*80/radius+105);

dot.style.marginLeft = (tablepontos.childNodes[childItem].childNodes[0].innerHTML\*80/radius+101) + "px";

dot.style.marginTop = (-tablepontos.childNodes[childItem].childNodes[1].innerHTML\*80/radius+105) + "px";

if ((ml >= 0) && (ml <= 205) && (mt >= 0) && (mt <= 205)){

$("div#area").append(dot);

}

dot.style.backgroundColor = "green";

if (tablepontos.childNodes[childItem].childNodes[3].innerHTML.indexOf("OUT") != -1){

dot.style.backgroundColor = "red";

}

if (tablepontos.childNodes[childItem].childNodes[2].innerHTML != radius){

console.log("Rechecking ..."+childItem);

tablepontos.childNodes[childItem].childNodes[3].innerHTML = "IN";

dot.style.backgroundColor = "green";

isInArea(tablepontos.childNodes[childItem].childNodes[0].innerHTML,tablepontos.childNodes[childItem].childNodes[1].innerHTML,radius,dot,tablepontos.childNodes[childItem].childNodes[3]);

tablepontos.childNodes[childItem].childNodes[2].innerHTML = radius;

}

}

}

}

function insertAfter(elem, refElem) {

return refElem.parentNode.insertBefore(elem, refElem.nextSibling);

}

$(document).ready(function(){

if (document.getElementById("pontos").childNodes[0] != undefined){

fillGraph();

}

$("input#forminput").bind("keyup",function(event){

$("input#forminput")[0].value = $("input#forminput")[0].value.replace(/\,/, ".");

keyCode = event.keyCode;

symbol = event.key;

value = $("input#forminput")[0].value;

if ((isNaN(value)) || (parseFloat(value) < -5.0) || (parseFloat(value) > 5.0)){

$("input#forminput").css("background-color", "red");

}

if (!isNaN(value) && parseFloat(value) >= -5.0 && parseFloat(value) <= 5.0){

$("input#forminput").css("background-color", "white");

}

});

$("input#formsubmit").bind("click",function(event){

if ($("input#forminput")[0].value == "" || isNaN($("input#forminput")[0].value) || parseFloat($("input#forminput")[0].value) < -5.0 || parseFloat($("input#forminput")[0].value) > 5.0){

alert("Y cannot has the value!");

event.preventDefault();

}

});

$("div#area").bind("click",function(event){

if (!isNaN($("[name = radius]")[0].value)){

radius = $("[name = radius]")[0].value\*1;

x = (event.pageX - event.target.offsetLeft - 103)\*(radius/80);

y = (-(event.pageY - event.target.offsetTop) + 107)\*(radius/80);

dot = document.createElement('div');

dot.className = "dot";

console.log(dot.style.marginLeft);

dot.style.marginLeft = (event.pageX - event.target.offsetLeft - 2) + "px";

dot.style.marginTop = (event.pageY - event.target.offsetTop - 2) + "px";

this.appendChild(dot);

dot.style.backgroundColor = "green";

raw = document.createElement("tr");

cell = document.createElement("td");

cell.innerHTML = x;

raw.appendChild(cell);

cell = document.createElement("td");

cell.innerHTML = y;

raw.appendChild(cell);

cell = document.createElement("td");

cell.innerHTML = radius;

raw.appendChild(cell);

cell = document.createElement("td");

cell.innerHTML = "IN";

result = isInArea(x,y,radius,dot,cell);

raw.appendChild(cell);

document.getElementById("pontos").childNodes[0].appendChild(raw);

} else {

alert("Cannot define coordinates : radius wasn't set");

}

});

$( "[name = radius]" ).change(function() {

$("div#area > div.dot").detach();

fillGraph();

});

});

**//Файл request.jsp**

<%@page language="java" import="controller.ControllerServlet" %>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<link href="styles/style.css" rel="stylesheet"/>

<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>

<script src="scripts/script.js"></script>

<title>Silhouette checker</title>

</head>

<body>

<h1>Выполнил студент группы P3217 Плюхин Дмитрий Алексеевич вариант 2911</h1>

<form name="edit" method="POST" action="/pip.lab2/home" id="edit">

<table id="formtable">

<tr>

<td>

<label>x: </label>

</td>

<td>

<input type="checkbox" name="x1" value="-4" unchecked><label class="after">-4</label>

<input type="checkbox" name="x2" value="-3" unchecked><label class="after">-3</label>

<input type="checkbox" name="x3" value="-2" unchecked><label class="after">-2</label>

<input type="checkbox" name="x4" value="-1" unchecked><label class="after">-1</label>

<input type="checkbox" name="x5" value="0" unchecked><label class="after">0</label>

<input type="checkbox" name="x6" value="1" unchecked><label class="after">1</label>

<input type="checkbox" name="x7" value="2" unchecked><label class="after">2</label>

<input type="checkbox" name="x8" value="3" unchecked><label class="after">3</label>

<input type="checkbox" name="x9" value="4" unchecked><label class="after">4</label>

</td>

</tr>

<tr>

<td>

<label>y: </label>

</td>

<td><input type="text" name="y" id="forminput"><br>

</td>

</tr>

<tr>

<td>

<label>r: </label>

</td>

<td>

<select size="1" multiple name="radius">

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

<option value="2.5">2.5</option>

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

<option selected value="1.5">1.5</option>

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

</select>

</td>

</tr>

</table>

<div id="area">

</div>

<div>

<input type="submit" value="Submit" name="submit" class="submit" id="formsubmit">

</div>

</form>

<%=ControllerServlet.getCheckedPontosInHTMLTable()%>

</body>

</html>

**//Файл ControllerServlet.java**

package controller;

import java.io.\*;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "ControllerServlet", loadOnStartup = 1, urlPatterns = {"/home"})

public class ControllerServlet extends HttpServlet {

@Override

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

handleRequest(req, res, 1);

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

handleRequest(req, res, 0);

}

private static boolean isNumber(String s) throws NumberFormatException {

try {

Double.parseDouble(s);

return true;

} catch (NumberFormatException e) {

return false;

}

}

protected boolean isThereData(HttpServletRequest req){

if ((req.getParameter("y") == null) || (req.getParameter("radius") == null) ||

(!isNumber(req.getParameter("radius"))) || (!isNumber(req.getParameter("y")))) return false;

if ((req.getParameter("x") != null) && (isNumber(req.getParameter("x")))) return true;

for (int i = 1; i <= 9; i++){

if ((req.getParameter("x"+i) != null) && (isNumber(req.getParameter("x"+i)))) return true;

}

return false;

}

protected void handleRequest(HttpServletRequest req, HttpServletResponse res, int method) throws ServletException, IOException{

PrintWriter out = res.getWriter();

String xValue = req.getParameter("x");

String yValue = req.getParameter("y");

String nextJSP = "/pip.lab2/request.jsp";

if ((method == 0) && (isThereData(req))){

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/check");

dispatcher.forward(req,res);

}

res.sendRedirect(nextJSP);

}

public static String getCheckedPontosInHTMLTable(){

return AreaCheckServlet.getCheckedPontosInHTMLTable();

}

}

**//Файл AreaCheckServlet.java**

package controller;

import java.util.ArrayList;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.ServletConfig;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import silhouette.\*;

@WebServlet(name = "AreaCheckServlet", loadOnStartup = 1, urlPatterns = {"/check"})

public class AreaCheckServlet extends HttpServlet {

private static PontoExListBean pontosBean;

private static boolean isNumber(String s) throws NumberFormatException {

try {

Double.parseDouble(s);

return true;

} catch (NumberFormatException e) {

return false;

}

}

@Override

public void init(ServletConfig config) throws ServletException {

super.init(config);

if (pontosBean == null){

pontosBean = new PontoExListBean();

}

}

@Override

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

GeneralSilhouette slh = new GeneralSilhouette(Double.parseDouble(request.getParameter("radius")));

double y = Double.parseDouble(request.getParameter("y"));

double radius = Double.parseDouble(request.getParameter("radius"));

double x = 0.0;

ArrayList<PontoEx> pontos = new ArrayList<PontoEx>();

PrintWriter out = response.getWriter();

if (((request.getParameter("x") != null) && (isNumber(request.getParameter("x"))))){

x = Double.parseDouble(request.getParameter("x"));

pontos.add(new PontoEx(x,y,radius,slh.checkPonto(new Ponto(x,y))));

AreaCheckServlet.pontosBean.add(pontos.get(pontos.size()-1));

}

for (int i = 1; i <= 9; i++){

if ((request.getParameter("x"+i) != null) && (isNumber(request.getParameter("x"+i)))){

x = Double.parseDouble(request.getParameter("x"+i));

pontos.add(new PontoEx(x,y,radius,slh.checkPonto(new Ponto(x,y))));

AreaCheckServlet.pontosBean.add(pontos.get(pontos.size()-1));

}

}

request.setAttribute("pontos",pontos);

if (pontos.size() != 0) {

out.println("<!DOCTYPE html><html><head><meta http-equiv='Content-Type' content='text/html; charset=windows-1251'><title>Silhouette checker</title></head><body><table border='1' cellspacing='1' cellpadding='1' align='center'><tr><td>x</td><td>y</td><td>r</td><td>Location</td></tr>");

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

out.println(pontos.get(i).toHTMLTableRaw());

}

out.println("</table><div align='center'><a href='/pip.lab2/home'>Back</a></div></body></html>");

} else {

out.println("<!DOCTYPE html><html><head><meta http-equiv='Content-Type' content='text/html; charset=UTF-8'><title>Silhouette checker</title></head><body><p>There are no points</p><div align='center'><a href='pip.lab2/home'>Back</a></div></body></html>");

}

}

public static String getCheckedPontosInHTMLTable(){

String table = "<table id='pontos'><tbody><tr><td>x</td><td>y<td>r</td><td>Location</td></tr>";

//out.println("<!DOCTYPE html><html><head><meta http-equiv='Content-Type' content='text/html; charset=windows-1251'><title>Silhouette checker</title></head><body><table border='1' cellspacing='1' cellpadding='1' align='center'><tr><td valign='top' align='center' height='20px'>Params</td></tr>");

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

table = table + AreaCheckServlet.pontosBean.get(i).toHTMLTableRaw();

}

table = table + "</tbody></table>";

return table;

}

}

**//Файл PontoExListBean.java**

package controller;

import java.util.ArrayList;

import silhouette.PontoEx;

public class PontoExListBean {

private ArrayList<PontoEx> pontos;

public PontoExListBean() {

pontos = new ArrayList<PontoEx>();

}

public void add(PontoEx ponto) {

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

if ((pontos.get(i).getX() == ponto.getX()) && (pontos.get(i).getY() == ponto.getY())){

pontos.get(i).setR(ponto.getR());

pontos.get(i).setInside(ponto.getInside());

return;

}

}

pontos.add(ponto);

}

public PontoEx get(int i) {

return pontos.get(i);

}

public int size(){

return pontos.size();

}

}

1. **Развертывание на сервере Glass Fish**



1. **Вывод**

В ходе лабораторной работы были исследованы способы создания веб-приложений при помощи JSP и сервлетов. Был сделан вывод, что в контексте MVC наиболее удобное место для применения JSP – компоненты, предназначенные для формирования пользовательского интерфейса, а для сервлетов – модули, работающие исключительно на стороне сервера и обрабатывающие какие-либо данные. Был сделан вывод о том, что правильная конфигурация приложения и сервера – самая важная деталь, которой следует уделять внимание в первую очередь и продумывать до мелочей. Кроме того, был получен опыт развертывания веб приложений на удаленном сервере и сделан вывод о нетривиальности этой задачи, требующей некоторых дополнительных знаний о работе выбранной технологии и сетевой архитектуре вообще.