Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики

Лабораторная работа №1

148/GET

**4100**

Дымов

Шамуков

Санкт-Петербург

2012

Оглавление

Задание 3

Исходный код 4

lab1/core/Area.java 4

lab1/core/AreaFactory.java 4

lab1/core/DescartesArea.java 4

lab1/core/HitChecker.java 5

lab1/core/shape/OurPoint.java 5

lab1/core/shape/Quadrant.java 5

lab1/core/shape/Rectangle.java 6

lab1/core/shape/Sector.java 6

lab1/core/shape/Shape.java 7

lab1/core/shape/Triangle.java 7

lab1/Lab1AreaFactory.java 8

lab1/RequestParser.java 8

lab1/Servlet.java 9

index.jsp 10

style.css 12

Вывод 13

# Задание

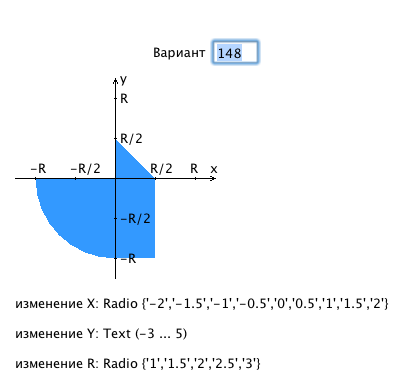
Разработать сервлет, определяющий попадание точки на координатной плоскости в заданную область и создать HTML-страницу, которая формирует данные для отправки их на обработку этим сервлетом.

Параметр R и координаты точки должны передаваться сервлету посредством HTTP-запроса по методу GET или POST в зависимости от варианта задания. Сервлет должен возвращать HTML-страницу с таблицей, содержащей полученные параметры и результат вычислений - факт попадания или непопадания точки в область.

Разработанная HTML-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. Любые некорректные значения (например, буквы в координатах точки или отрицательный радиус) должны блокироваться.
5. Каскадную таблицу стилей (CSS), устанавливающую размеры и позиционирование полей ввода.

Разработанные сервлет и HTML-страницу необходимо развернуть на сервере приложений [GlassFish](http://glassfish.java.net/" \t "_blank).



# Исходный код

\_\_\_\_\_

## lab1/core/Area.java

package com.dymsha.spip.lab1.core;

import com.dymsha.spip.lab1.core.shape.OurPoint;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:30 PM

\* To change this template use File | Settings | File Templates.

\*/

public interface Area {

boolean containPoint(OurPoint point);

}

\_\_\_\_\_

## lab1/core/AreaFactory.java

package com.dymsha.spip.lab1.core;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/11/11

\* Time: 6:36 AM

\* To change this template use File | Settings | File Templates.

\*/

public interface AreaFactory {

Area produceArea(double radius);

}

\_\_\_\_\_

## lab1/core/DescartesArea.java

package com.dymsha.spip.lab1.core;

import com.dymsha.spip.lab1.core.shape.OurPoint;

import com.dymsha.spip.lab1.core.shape.Shape;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:46 PM

\* To change this template use File | Settings | File Templates.

\*/

public class DescartesArea implements Area {

private List<Shape> shapes;

public DescartesArea() {

shapes = new ArrayList<Shape>();

}

public void addShape(final Shape shape) {

shapes.add(shape);

}

public boolean containPoint(OurPoint point) {

for (final Shape shape : shapes) {

if (shape.containPoint(point)) {

return true;

}

}

return false;

}

}

\_\_\_\_\_

## lab1/core/HitChecker.java

package com.dymsha.spip.lab1.core;

import com.dymsha.spip.lab1.core.shape.OurPoint;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA. User: evgenij Date: 3/11/11 Time: 5:38 AM To change

\* this template use File | Settings | File Templates.

\*/

public class HitChecker {

public List<OurPoint> checkPoints(final Area area, final List<OurPoint> points) {

final List<OurPoint> res = new ArrayList<OurPoint>();

for (OurPoint point : points) {

if (area.containPoint(point)) {

res.add(point);

}

}

return res;

}

public boolean checkPoint(final Area area, final OurPoint point) {

return area.containPoint(point);

}

}

\_\_\_\_\_

## lab1/core/shape/OurPoint.java

package com.dymsha.spip.lab1.core.shape;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/11/11

\* Time: 7:54 AM

\* To change this template use File | Settings | File Templates.

\*/

public class OurPoint implements Shape {

private final double x, y;

public OurPoint(final double x, final double y) {

this.x = x;

this.y = y;

}

public boolean containPoint(OurPoint point) {

return point.getX() == x && point.getY() == y;

}

public double getX() {

return x;

}

public double getY() {

return y;

}

@Override

public String toString() {

return "(" + x + ", " + y + ")";

}

}

\_\_\_\_\_

## lab1/core/shape/Quadrant.java

package com.dymsha.spip.lab1.core.shape;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/11/11

\* Time: 12:36 AM

\* To change this template use File | Settings | File Templates.

\*/

public enum Quadrant {

FIRST,

SECOND,

THIRD,

FOURTH

}

\_\_\_\_\_

## lab1/core/shape/Rectangle.java

package com.dymsha.spip.lab1.core.shape;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:43 PM

\* To change this template use File | Settings | File Templates.

\*/

public class Rectangle implements Shape {

private OurPoint apex;

private double width, height;

public Rectangle(final OurPoint apex, final double width, final double height) {

this.apex = apex;

this.width = width;

this.height = height;

}

public Rectangle(final double x, final double y, final double width, final double height) {

this.apex = new OurPoint(x,y);

this.width = width;

this.height = height;

}

public boolean containPoint(final OurPoint point) {

return ( height >= 0 ? (point.getY() >= apex.getY() && point.getY() <= apex.getY() + height)

: (point.getY() <= apex.getY() && point.getY() >= apex.getY() + height))

&& ( width >= 0 ? point.getX() >= apex.getX() && point.getX() <= apex.getX() + width

: point.getX() <= apex.getX() && point.getX() >= apex.getX() + width);

}

}

\_\_\_\_\_

## lab1/core/shape/Sector.java

package com.dymsha.spip.lab1.core.shape;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:43 PM

\* To change this template use File | Settings | File Templates.

\*/

public class Sector implements Shape {

private Quadrant quadrant;

private double radius;

public Sector(Quadrant quadrant, double radius) {

this.quadrant = quadrant;

this.radius = radius;

}

public boolean containPoint(OurPoint point) {

switch (quadrant) {

case FIRST:

if ((point.getX() < 0) || (point.getY() < 0)) {

return false;

}

break;

case SECOND:

if ((point.getX() > 0) || (point.getY() < 0)) {

return false;

}

break;

case THIRD:

if ((point.getX() > 0) || (point.getY() > 0)) {

return false;

}

break;

case FOURTH:

if ((point.getX() < 0) || (point.getY() > 0)) {

return false;

}

break;

}

return pointInDaCircle(point, radius);

}

private boolean pointInDaCircle(OurPoint point, double radius) {

if ((Math.pow(point.getX(), 2) + Math.pow(point.getY(), 2)) <= Math.pow(radius, 2)) {

return true;

}

return false;

}

}

\_\_\_\_\_

## lab1/core/shape/Shape.java

package com.dymsha.spip.lab1.core.shape;

import com.dymsha.spip.lab1.core.Area;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:42 PM

\* To change this template use File | Settings | File Templates.

\*/

public interface Shape extends Area {

}

\_\_\_\_\_

## lab1/core/shape/Triangle.java

package com.dymsha.spip.lab1.core.shape;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/10/11

\* Time: 11:44 PM

\* To change this template use File | Settings | File Templates.

\*/

public class Triangle implements Shape {

private double height, width;

public Triangle(double width, double height) {

this.height = height;

this.width = width;

}

public boolean containPoint(OurPoint point) {

if (point.getX() \* width >= 0 && point.getY() \* height >= 0) {

if (height >= 0) {

if (height \* (-1 \* point.getX()/width + 1) >= point.getY()) {

return true;

}

} else {

if (height \* (-1 \* point.getX()/width + 1) <= point.getY()) {

return true;

}

}

}

return false;

}

}

\_\_\_\_\_

## lab1/Lab1AreaFactory.java

package com.dymsha.spip.lab1;

import com.dymsha.spip.lab1.core.Area;

import com.dymsha.spip.lab1.core.AreaFactory;

import com.dymsha.spip.lab1.core.DescartesArea;

import com.dymsha.spip.lab1.core.shape.\*;

/\*\*

\* Created by IntelliJ IDEA.

\* User: evgenij

\* Date: 3/11/11

\* Time: 6:37 AM

\* To change this template use File | Settings | File Templates.

\*/

public class Lab1AreaFactory implements AreaFactory {

public Area produceArea(double radius) {

final DescartesArea lab1Area = new DescartesArea();

lab1Area.addShape(new Triangle(0.5 \* radius, 0.5 \* radius));

lab1Area.addShape(new Sector(Quadrant.THIRD, radius));

lab1Area.addShape(new Rectangle(new OurPoint(0,0), 0.5 \* radius, -1 \* radius));

return lab1Area;

}

}

\_\_\_\_\_

## lab1/RequestParser.java

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dymsha.spip.lab1;

import javax.servlet.http.HttpServletRequest;

/\*\*

\*

\* @author eugene

\*/

public class RequestParser {

private double x;

private double y;

private double r;

public RequestParser(HttpServletRequest aRequest) {

this.x = getDoubleFromString(aRequest.getParameter("x"));

this.y = getDoubleFromString(aRequest.getParameter("y"));

this.r = getDoubleFromString(aRequest.getParameter("r"));

}

public double getX() {

return this.x;

}

public double getY() {

return this.y;

}

public double getR() {

return this.r;

}

private double getDoubleFromString(String source) {

return Double.parseDouble(source.replaceFirst(",", "."));

}

}

\_\_\_\_\_

## lab1/Servlet.java

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.dymsha.spip.lab1;

import com.dymsha.spip.lab1.core.Area;

import com.dymsha.spip.lab1.core.HitChecker;

import com.dymsha.spip.lab1.core.shape.OurPoint;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author eugene

\*/

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

public class Servlet extends HttpServlet {

/\*\*

\* Processes requests for both HTTP

\* <code>GET</code> and

\* <code>POST</code> methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

/\*

\* TODO output your page here. You may use following sample code.

\*/

RequestParser requestParser = new RequestParser(request);

Lab1AreaFactory lab1AreaFactory = new Lab1AreaFactory();

Area lab1Area = lab1AreaFactory.produceArea(requestParser.getR());

HitChecker hitChecker = new HitChecker();

out.println("<html>");

out.println("<head>");

out.println("<title>Lab.1</title>");

out.println("<link href='style.css' rel='stylesheet' type='text/css'>");

out.println("</head>");

out.println("<body>");

out.println("<img src="+request.getRequestURL().substring(0, request.getRequestURL().lastIndexOf("/"))+"/var.png><br>");

out.println("Your Points:<br>");

out.println("<b>x:</b> "+requestParser.getX()+"<br>");

out.println("<b>y:</b> "+requestParser.getY()+"<br>");

out.println("<b>R:</b> "+requestParser.getR()+"<br>");

out.println("<hr>");

if (hitChecker.checkPoint(lab1Area, new OurPoint(requestParser.getX(), requestParser.getY()))) {

out.println("<i style='background:#d5f8d5;'>Success! Area contains the point!</i>");

} else {

out.println("<i style='background:#fedbd2'>Failure! Area does not contain the point.</i>");

}

out.println("<a href="+request.getRequestURL().substring(0, request.getRequestURL().lastIndexOf("/"))+"><i>Again?</i></a>");

out.println("</body>");

out.println("</html>");

} finally {

out.close();

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP

\* <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP

\* <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

\_\_\_\_\_

## index.jsp

<%--

Document : index

Created on : 13.03.2012, 17:10:38

Author : eugene

--%>

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

<!DOCTYPE html>

<html>

<head>

<title>Lab.1</title>

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

<script language="javascript">

function validateForm() {

var y = document.forms["labForm"]["y"].value;

if (y.match(/^-?\d+[\.|\,]?\d+$/)) {

return true;

}

if (y.match(/^-?\d$/)) {

return true;

}

if (y.match(/^-?\d+e-?\d+$/)) {

return true;

}

if (y < -3 || y > 5) {

document.getElementById("error").innerHTML="Out of range!";

return false;

}

document.getElementById("error").innerHTML="Incorrect data!";

return false;

}

</script>

</head>

<body>

<form name="labForm" action="Servlet" method="GET" onsubmit="return validateForm();">

<table id="mainTable" rules=all>

<tr>

<td colspan="2">Lab1.Dymov &amp; Shamukov.4100</td>

</tr>

<tr>

<td colspan="2"><div id="variant">Variant 148/GET</div></td>

</tr>

<tr>

<td id="titleRows">X</td>

<td>

<table id="insideTable">

<tr>

<td id="hoverable"><input id="x0" type="radio" name="x" value="-2" CHECKED/>

<label>-2 </label></td>

<td id="hoverable"><input id="x1" type="radio" name="x" value="-1.5"/>

<label>-1.5</label> </td>

<td id="hoverable"><input id="x2" type="radio" name="x" value="-1"/>

<label>-1</label></td>

</tr>

<tr>

<td id="hoverable"><input id="x3" type="radio" name="x" value="-0.5"/>

<label>-0.5</label> </td>

<td id="hoverable"><input id="x4" type="radio" name="x" value="0"/>

<label>0</label> </td>

<td id="hoverable"><input id="x5" type="radio" name="x" value="0.5"/>

<label>0.5</label> </td>

</tr>

<tr>

<td id="hoverable"><input id="x6" type="radio" name="x" value="1"/>

<label>1</label> </td>

<td id="hoverable"><input id="x7" type="radio" name="x" value="1.5"/>

<label>1.5</label></td>

<td id="hoverable"><input id="x8" type="radio" name="x" value="2"/>

<label>2</label></td>

</tr>

</table>

</tr>

<tr>

<td id="titleRows">Y</td><td id="hoverable"><input id="inputY" type="text" name="y" value="0.0"/></td>

</tr>

<tr height=50>

<td id="titleRows">R</td>

<td>

<table id="insideTable">

<tr>

<td id="hoverable"><input id="r0" type="radio" name="r" value="1" CHECKED/>

<label>1 </label></td>

<td id="hoverable"><input id="r1" type="radio" name="r" value="1.5"/>

<label>1.5</label> </td>

<td id="hoverable"><input id="r2" type="radio" name="r" value="2"/>

<label>2</label></td>

</tr>

<tr>

<td id="hoverable"><input id="r3" type="radio" name="r" value="2.5"/>

<label>2.5</label> </td>

<td id="hoverable"><input id="r4" type="radio" name="r" value="3"/>

<label>3</label> </td>

</tr>

</table>

</td>

</tr>

<tr>

<td id="confirmCell" colspan=2><input id="confirmButton" type="submit" value="Confirm"></td>

</tr>

<tr>

<td colspan=2><div id="error"></div></td>

</tr>

</table>

</form>

</body>

</html>

## style.css

/\*

Document : style

Created on : 13.03.2012, 17:26:03

Author : eugene

Description:

Purpose of the stylesheet follows.

\*/

root {

display: block;

}

a:link {

color: black;

text-decoration: underline;

}

a:visited {

color: black;

text-decoration: underline;

}

a:hover {

color: black;

background-color: white;

text-decoration: none;

}

#globalDiv {

width: 20%;

}

#title {

text-align:center;

font-size: 15pt;

/\* width: 20%;\*/

}

#variant {

background: #d3e4e1;

padding: 2px;

font-size: 11pt;

}

#variant:hover {

background: #b4d0cc;

}

#titleRows {

text-align:center;

width: 30pt;

}

#inputY {

width: 99%;

border:0px;

text-align:center;

}

#mainTable {

font-size: 15pt;

border:1px solid #000

}

#insideTable {

font-size: 13pt;

width: 100%;

}

#hoverable {

font-size: 10pt;

}

#hoverable:hover {

background: #f3f7f7;

}

#inputY:hover {

background: #f3f7f7;

}

#confirmButton {

width:100%;

}

#confirmCell {

background: #d3e4e1;

}

#confirmCell:hover {

background: #b4d0cc;

}

#error {

background: #609c92;

width: 100%;

text-align: center;

color: white;

}

# Вывод

Проделав лабораторную работу, мы познакомились с основами html, css, разработкой сервлетов и http.