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.

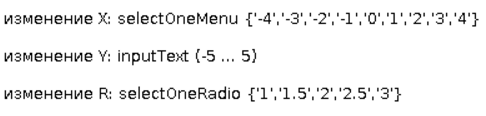
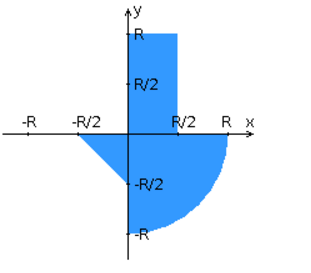
**Стартовая страница должна содержать следующие элементы:**

* "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.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html" xmlns:f="http://java.sun.com/jsf/core"

xmlns:p="http://primefaces.org/ui">

<h:head>

<title>Main page</title>

<h:outputStylesheet name="css/mainPageStyle.css"/>

<h:outputScript name="js/validateY.js"/>

</h:head>

<h:body>

<div class="wrapper" style="margin: auto;">

<div class="content" style="margin-top: 50px;">

<h:form id="mainForm" style="color: #004dff; font-family: sans-serif;">

<div style="width: 500px; float: left; clear: left">

<p style="margin-bottom: 5px;font-family: sans-serif; color: #004dff;">

Select X:</p>

<h:selectOneMenu value="#{check.x}">

<f:selectItem itemValue="-4" itemLabel="-4" />

<f:selectItem itemValue="-3" itemLabel="-3" />

<f:selectItem itemValue="-2" itemLabel="-2" />

<f:selectItem itemValue="-1" itemLabel="-1" />

<f:selectItem itemValue="0" itemLabel="0" />

<f:selectItem itemValue="1" itemLabel="1" />

<f:selectItem itemValue="2" itemLabel="2" />

<f:selectItem itemValue="3" itemLabel="3" />

<f:selectItem itemValue="4" itemLabel="4" />

</h:selectOneMenu>

<p style="margin-bottom: 0;font-family: sans-serif; color: #004dff;">

Select Y {-5;5}:

<p:inputText id="Y" required="#{param['mainForm:areaClick']==null}"

requiredMessage="Value is required"

value="#{check.y}" onchange="validateY()" onkeyup="validateY()"

onblur="validateY()"/>

<p:message for="Y" style="color: red"/>

</p>

<p style="margin-bottom: 0;font-family: sans-serif; color: #004dff;">

Select R:

<p:selectOneRadio id="R" label="Select R" required="true"

requiredMessage="Value is required" value="#{check.r}">

<f:selectItem itemValue="1" itemLabel="1"/>

<f:selectItem itemValue="1.5" itemLabel="1.5"/>

<f:selectItem itemValue="2" itemLabel="2"/>

<f:selectItem itemValue="2.5" itemLabel="2.5"/>

<f:selectItem itemValue="3" itemLabel="3"/>

<p:ajax listener="#{check.changedRadius}" update="area"/>

</p:selectOneRadio>

</p>

<p:commandButton id="submit" value="Submit" action="#{check.checkPoint}"

update="@(form),resTable, area"/>

<p style="font-size: 20px; margin-bottom: 5px">Result table:</p>

<p:dataTable id="resTable" value="#{results.results}" var="res"

styleClass="result-table"

headerClass="result-table-header"

rowClasses="result-table-odd-row, result-table-even-row"

style="width: 400px; float: left;"

scrollable="true" scrollHeight="400">

<p:column>

<f:facet name="header">X</f:facet>

#{res.xCoord}

</p:column>

<p:column>

<f:facet name="header">Y</f:facet>

#{res.yCoord}

</p:column>

<p:column>

<f:facet name="header">R</f:facet>

#{res.radius}

</p:column>

<p:column>

<f:facet name="header">In area</f:facet>

#{res.inArea}

</p:column>

</p:dataTable>

</div>

<p:graphicImage id="area" value="#{imageBean.image}" style="display: inline-block; float: right;"

cache="false" onclick="areaClick()"/>

<p:inputText id="yClick" value="#{check.yClick}" style="display: none"/>

<p:inputText id="xClick" value="#{check.xClick}" style="display: none"/>

<p:commandButton id="areaClick" style="display: none" action="#{check.checkClick}"

update="resTable, area"/>

</h:form>

<h:form style="margin-top: 30px; float: left;clear: left">

<p:commandButton id="goToStart" action="toStart" value="Start page"/>

</h:form>

</div>

</div>

<h:outputScript name="js/image.js"/>

</h:body>

</html>

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui" xmlns:f="http://java.sun.com/jsf/core">

<h:head>

<title>Start page</title>

<h:outputStylesheet name="css/startPageStyle.css"/>

</h:head>

<h:body>

<div class="wrapper">

<div class="header">

<p>Сорокин Юрий Борисович</p>

<p>Группа P3217</p>

<p>Вариант 1758</p>

</div>

<div class="content" style="text-align: center">

<div class="clock">

<h:form id="clockForm" style="color: #004dff; font-family: sans-serif;margin-bottom: 20px;">

<h:outputLabel id="clockLabel" value="#{clockTimer.date}"/>

<h:commandButton id="clockTimerBtn" style="display: none" action="#{clockTimer.curDate}">

<f:ajax render="clockLabel"/>

</h:commandButton>

<h:outputScript name="js/clock.js"/>

</h:form>

</div>

<h:form>

<p:commandButton id="submit" action="toMain" value="Main page"/>

</h:form>

</div>

</div>

</h:body>

</html>

@ManagedBean(name = "check")

@ViewScoped

public class CheckBean implements Serializable {

private double X;

private String Y;

private double R;

private String yClick;

private String xClick;

@ManagedProperty(value = "#{imageBean}")

private ImageBean imageBean;

@ManagedProperty(value = "#{results}")

private ResultsBean resultsBean;

public CheckBean() {

R = 1;

}

public void setImageBean(ImageBean imageBean) {

this.imageBean = imageBean;

}

public void setResultsBean(ResultsBean resultsBean) {this.resultsBean = resultsBean;}

public ResultsBean getResultsBean() {return resultsBean;}

public String getyClick() {

return yClick;

}

public void setyClick(String yClick) {

this.yClick = yClick;

}

public String getxClick() {

return xClick;

}

public void setxClick(String xClick) {

this.xClick = xClick;

}

public void setX(double x) {

X = x;

}

public double getX(){return X;}

public String getY() {

return Y;

}

public void setY(String y) {

Y = y;

}

public double getR() {

return R;

}

public void setR(double r) {

R = r;

}

public void checkPoint() {

double y = Double.parseDouble(Y.replace(',', '.'));

double r = R;

double x = X;

Point point = new Point(x, y, r, inArea(x, y, r));

resultsBean.addResult(point);

imageBean.addPoint(new Point(point.getxCoord(), point.getyCoord(), point.getRadius(), point.isInArea()));

imageBean.setRadius(r);

checkExistedPoints(r);

}

public void checkClick() {

double y = Double.parseDouble(yClick.replace(',', '.'));

double x = Double.parseDouble(xClick.replace(',', '.'));

y = Math.rint(100.0 \* y) / 100.0;

x = Math.rint(100.0 \* x) / 100.0;

double r = R;

Point point = new Point(x, y, r, inArea(x, y, r));

resultsBean.addResult(point);

imageBean.addPoint(new Point(point.getxCoord(), point.getyCoord(), point.getRadius(), point.isInArea()));

imageBean.setRadius(r);

checkExistedPoints(r);

}

private boolean inArea(double x, double y, double r) {

return (x>=0 & x<=r/2 & y<=r & y>=0)

|| (x>=0 & x <= r & y<=0 & y >= -r & x\*x+y\*y<r\*r)

|| (x<=0 & x>=-r/2 & y<=0 & y>=-r/2 & y>-r/2-x);

}

private void checkExistedPoints(double r) {

for (Point p : imageBean.getPoints()) {

p.setRadius(r);

p.setInArea(inArea(p.getxCoord(), p.getyCoord(), r));

}

}

public void changedRadius() {

imageBean.setRadius(R);

checkExistedPoints(R);

}

}

**Вывод**

При выполнении данной работы, я ознакомился c принципами создания приложения на базе JSF.