

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

Кафедра информатики и прикладной математики

**Дисциплина «Программирование интернет-приложений»  
Лабораторная работа №3**

**Вариант 773**

**Выполнил:**

Съестов Дмитрий Вячеславович  
Группа Р3217

**Преподаватель:**

Николаев Владимир Вячеславович

Санкт-Петербург  
2018

## Задание

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:**

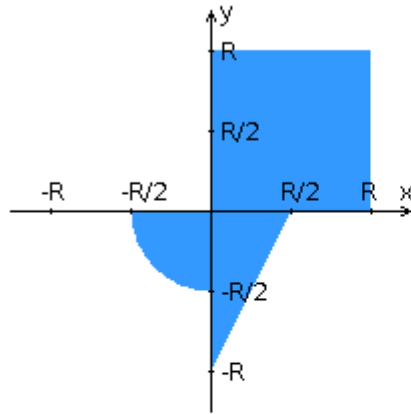
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Интерактивные часы, показывающие текущие дату и время, обновляющиеся раз в 10 секунд.
- Ссылку, позволяющую перейти на основную страницу приложения.

**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** (префикс "ace") и **PrimeFaces** (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 зависеть от факта попадания / непадения 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:**

- Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle.
- Для доступа к БД необходимо использовать JPA с ORM-провайдером на усмотрение студента.
- Для управления списком результатов должен использоваться **Session-scoped Managed Bean**.
- Конфигурация управляемых бинов должна быть задана с помощью параметров в конфигурационном файле.
- Правила навигации между страницами приложения должны быть заданы в отдельном конфигурационном файле.



изменение X: selectOneMenu {'-4','-3','-2','-1','0','1','2','3','4'}

изменение Y: inputText {-5 ... 5}

изменение R: p:slider {2 ... 5}, шаг изменения - 0.1

## Стартовая страница

```
<?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://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core">

    <f:view>
        <link type="text/css" rel="stylesheet" href="style.css" />
        <h:head>
            <title> Laboratory work #3 </title>
        </h:head>

        <h:body onload="startClock();">
            <div id="header">
                <h1 id="lab_title">Laboratory work #3</h1>
                <h2>Variant 773</h2>
                <h3>Dmitry Sjestov</h3>
                <h3>Group P3217</h3>
                <h3>Teacher: V.V.Nikolaev</h3>
            </div>

            <div class="centered">
                <h:outputLabel id="date_label"/> <br/>
                <h:outputLabel id="time_label"/>
            </div>

            <div class="centered">
                <h:link value="Main page" outcome = "main"/>
            </div>
        </h:body>
    </f:view>

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

## Главная страница

```
<?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://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:p="http://primefaces.org/ui">
    <h:head>
        <title> Laboratory work #3 </title>
        <link type="text/css" rel="stylesheet" href="style.css" />
        <h:outputScript library="primefaces" name="jquery/jquery.js" target="head" />
    </h:head>
    <h:body>
        <div class="outer" id="main_block">
            <div class="inner" id="canvas_block">
                <canvas id="canvas" width="705px" height="705px"/>
                <h:outputScript library="js" name="canvas.js"/>
            </div>

            <div class="inner" style="padding-left: 150px">
                <div id="validation_block">
                    <h:form id="form" method="get" action="#{resultManagerBean.submit}">
                        <h:outputText id="output" value="R = #{resultManagerBean.r}" /> <br/>
                        <h:inputHidden id="r_field" value="#{resultManagerBean.r}" />
                        <p:slider for="r_field" display="output" displayTemplate="R = {value}"
                            minValue="1" maxValue="5" step="1"
                            onSlideEnd="updateCanvas()"/>

                        <h:inputHidden id="x_field" value="#{resultManagerBean.x}"/>

                        <h:outputText value="X = " />
                        <p:selectOneMenu id="x_menu" onchange="setXField()" style="height: 20px">
                            <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" />
                        </p:selectOneMenu> <br/>

                        <h:outputText value="Y = " />
                        <p:inputText id="y_field" value="#{resultManagerBean.y}" label="Y"
                            size="5">
                            <f:convertNumber locale="us"/>
                            <f:validateDoubleRange minimum="-5" maximum="5"/>
                        </p:inputText> <br/>
                        <h:message for="y_field" style="color:red"/> <br/>

                        <h:commandButton id="submit" value="Submit"
                            action="#{resultManagerBean.submit}"/>
                    </h:form>
                </div>
            </div>
        </div>
    </h:body>
</html>
```

```

<div>
    <p:dataTable id="resultsTable" value="#{resultManagerBean.resultList}" var="result">
        <p:column headerText="X">
            <h:outputText value="#{result.x}"/>
        </p:column>
        <p:column headerText="Y">
            <h:outputText value="#{result.y}"/>
        </p:column>
        <p:column headerText="R">
            <h:outputText value="#{result.r}"/>
        </p:column>
        <p:column headerText="Result">
            <h:outputText value="#{result.hit ? 'Yes' : 'No'}"/>
        </p:column>
    </p:dataTable>
</div>

<div class="centered">
    <h:form id="historyForm">
        <h:commandButton value="Clear" action="#{resultManagerBean.clearHistory}"/>
    </h:form>
</div>

<div class="centered">
    <h:link outcome = "index">Start page</h:link>
</div>
</h:body>
</html>

```

## ResultManagerBean

```

public class ResultManagerBean implements Serializable
{
    private double x, y, r;
    private final SessionFactory sessionFactory;

    public ResultManagerBean()
    {
        r = 1;
        try
        {
            ServletContext ctx = (ServletContext)
                FacesContext.getCurrentInstance().getExternalContext().getContext();

            String configPath = ctx.getRealPath("/") + "WEB-INF/hibernate.cfg.xml";
            sessionFactory = new Configuration().configure(new
                File(configPath)).buildSessionFactory();
        }
        catch (Throwable ex)
        {
            System.err.println("Failed to create SessionFactory: " + ex);
            throw new SessionException(ex.getMessage());
        }
    }
}

```

```

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

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

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

public boolean checkArea(double x, double y)
{
    Point p = new Point(x, y);
    Point center = new Point(0, 0);

    //четверть окружности радиуса R/2 в III четверти
    if (p.x <= 0 && p.y <= 0 && Utils.distance(center, p) <= r/2)
        return true;

    //Квадрат R x R в I четверти
    if (p.x >= 0 && p.x <= r && p.y >= 0 && p.y <= r)
        return true;

    ///треугольник в IV четверти
    Point a = new Point(r/2, 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));
}

public void submit()
{
    boolean hit = checkArea(x, y);

    ResultEntity result = new ResultEntity();
    result.setX(x);
    result.setY(y);
    result.setR(r);
    result.setHit(hit);

    Transaction tx = null;
    try
    {
        Session session = sessionFactory.openSession();
        tx = session.beginTransaction();
        session.flush();
        session.save(result);
        tx.commit();
    }
    catch (Throwable e)
    {
        if (tx != null) tx.rollback();
        e.printStackTrace();
    }
}

```

```

public List getResultList()
{
    Session session;
    List resultList = new ArrayList<ResultEntity>();

    try
    {
        session = sessionFactory.openSession();
        Query query = session.createQuery("FROM ResultEntity ORDER BY id DESC");
        resultList = query.list();
    }
    catch (Throwable e)
    {
        e.printStackTrace();
    }

    return resultList;
}

public void clearHistory()
{
    Transaction tx = null;
    try
    {
        Session session = sessionFactory.openSession();
        tx = session.beginTransaction();
        Query query = session.createQuery("DELETE FROM ResultEntity");
        query.executeUpdate();
        tx.commit();
    }
    catch (Throwable e)
    {
        if (tx != null) tx.rollback();
        e.printStackTrace();
    }
}
}

```

### **Вывод**

В ходе выполнения данной работы я ознакомился с принципами работы объектно-реляционного преобразования, научился сохранять объекты Java в базу данных и наоборот, а также работать с фреймворком JSF и Java-бинами.