

СПБ НИУ ИТМО
«УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ»

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
Вариант №505

Выполнили:

Ощепков А.А.
Буланцов А.М.
группа: Р3202

Преподаватель:

Николаев В.В.

Санкт-Петербург 2017

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 / 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:

Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle. Для доступа к БД необходимо использовать протокол JDBC без каких-либо дополнительных библиотек. Для управления списком результатов должен использоваться Application-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.

Текст программы

```

1      private MainModel model;
2
3
4      @ManagedProperty("point")
5      private Point point;
6
7      @PostConstruct
8      void init(){
9          //          FacesContext fc = FacesContext.getCurrentInstance();
10         //          model = fc.getApplication().evaluateExpressionGet(fc,"#{
model}",MainModel.class);
11         point = new Point(0, 0, 0.1);
12     }
13
14     public AreaInteractionController(){
15
16     }
17     public Point getPoint() {

```

```

18         return point;
19     }
20
21     public void setPoint(Point point) {
22         this.point = point;
23     }
24
25     public void handle_sususu() throws Exception{
26         model.addPoint(point);
27     }
28
29     public MainModel getModel() {
30         return model;
31     }
32
33     public void setModel(MainModel model) {
34         this.model = model;
35     }
36 }

```

```

1 <?xml version='1.0' encoding='UTF-8'?>
2 <faces-config version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5     http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">
6
7     <managed-bean>
8         <managed-bean-name>clock </managed-bean-name>
9         <managed-bean-class>smt.Beans.ClockController </managed-bean-
class>
10         <managed-bean-scope>session </managed-bean-scope>
11     </managed-bean>
12
13     <managed-bean>
14         <managed-bean-name>model </managed-bean-name>
15         <managed-bean-class>smt.Beans.MainModel </managed-bean-class>
16         <managed-bean-scope>application </managed-bean-scope>
17         <managed-property>
18             <property-name>database </property-name>
19             <value>#{dbImpl}</value>
20         </managed-property>
21     </managed-bean>
22     ...

```

Выводы

В результате выполнения данной лабораторной работы, мы познакомились с технологией JSF, впервые на практике использовали DI, сделали ещё один шагок в болота энтерпрайза.