

**Национальный исследовательский университет компьютерных
технологий, механики и оптики**

Факультет ПИиКТ

**Веб-программирование.
Лабораторная работа №3.**

Вариант №30609

Работу выполнил: Кулаков Никита

Группа: Р3230

Преподаватель: Каюков И.А.

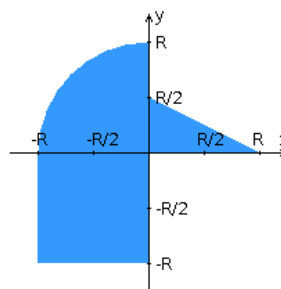
Город: Санкт-Петербург

2021 год

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.



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

изменение Y: inputText (-3 ... 3)

изменение R: p:spinner (0.1 ... 3), шаг изменения - 0.1

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 5 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 PostgreSQL.
- 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 аннотаций.
- Правила навигации между страницами приложения должны быть заданы в отдельном конфигурационном файле.

Ссылка на репозиторий: <https://github.com/zubrailx/web-lab-3>

Альтернативная ссылка: <https://github.com/zubrailx/university/tree/main/year-2/Web-programming/lab-3>

Вывод:

В ходе выполнения уже третьей лабораторной работы я познакомился с разработкой Java EE приложений с помощью фреймворка JSF. На фронте использовались JSF-шаблоны, страницы написаны в XHTML файлах. В качестве компонентов были использованы элементы из неймспейса jsf/facelets, primefaces, jsf/html. Все обращения к серверу посредством кнопок и svg listenera поддерживают ajax.

На стороне сервера был использован bean компонент для вывода и сохранения данных от пользователя, bean, являющийся контроллером для передачи управления на model и взаимодействия с бд PostgreSQL. Для визуализации таблицы используются DTO объекты, модель работает с Entity объектами. Конвертация объектов осуществляется с помощью конвертера. Для доступа к бд использовался JPA с ORM-провайдером Hibernate.

Чтобы никто не решил глупым образом воспользоваться аккаунтом на гелиосе, логин и пароль для осуществления соединения с бд загружаются из файла. Сахароза