

Национальный исследовательский университет ИТМО
Факультет программной инженерии и компьютерной техники

Веб-программирование Лабораторная работа №3

Работу выполнил:
Бавыкин Роман
Группа: Р3210
Вариант 10315

Санкт-Петербург
2021 г.

1. 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:

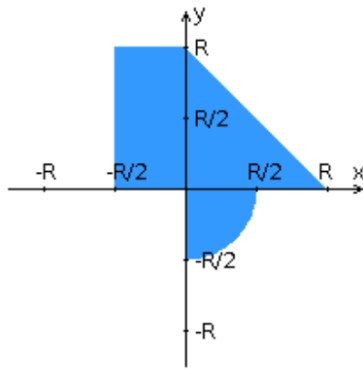
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Интерактивные часы, показывающие текущие дату и время, обновляющиеся раз в 8 секунд.
- Ссылку, позволяющую перейти на основную страницу приложения.

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 PostgreSQL.
- Для доступа к БД необходимо использовать ORM EclipseLink.
- Для управления списком результатов должен использоваться Application-scoped Managed Bean.
- Конфигурация управляемых бинов должна быть задана с помощью аннотаций.
- Правила навигации между страницами приложения должны быть заданы в отдельном конфигурационном файле.



изменение X: p:slider {-5 ... 5}, шаг изменения - 1

изменение Y: inputText {-3 ... 3}

изменение R: selectOneRadio {'1','2','3','4','5'}

2. Исходный код программы:

<https://github.com/robqqq/Web3>

3. Разработанное Web-приложение:

Лабораторная работа №3
Бавыкин Роман Алексеевич, Р3210
Вариант 10315

23:59:32
09.11.2021

To main page

Лабораторная работа №3
Бавыкин Роман Алексеевич, Р3210
Вариант 10315

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

To index page

X	Y	R	Result
-0.9475	1.9680001831054688	4.0	true
-1.7875	1.9080001831054688	4.0	true
1.2725	1.2480001831054688	4.0	true
0.4525	-0.9139998168943312	4.0	true
-1.1275	0.6480001831054687	4.0	true

4. Выводы по работе.

Во время выполнения лабораторной работы изучил работу Java Server Faces Framework и реализовал на его основе web-приложения. Изучил сторонние библиотеки JSF на примере PrimeFaces. Изучил спецификацию JPA и использовал в проекте ее реализацию ORM EclipseLink. Научился работать с управляемыми бинами.

JSF отличается от других спецификаций для разработки Web-приложений с MVC архитектурой тем, что основывается на основе компонентов. За счет того, что в страницу разметки нельзя встраивать Java код, в отличие от JSP, а все взаимодействие происходит с помощью бинов, получается более сильное отделение представления от модели и контроллера.

Спецификация JPA позволяет облегчить работу с базами данных, в отличие от использования JDBC напрямую, поскольку позволяет определить сущности в Java, которые будут соответствовать строкам в таблице базы данных, что позволяет в удобном виде сохранять объекты в базу данных и доставать их оттуда.

Осуждаю JS, следующую лабу буду писать на TypeScript.s