



Федеральное государственное автономное образовательное  
учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет ПИ и КТ

Лабораторная работа №3  
по дисциплине: «Веб-программирование»

Вариант 3552

Выполнил:

**Болорболд Аригуун,**

группа Р3211

Преподаватель:

**Егошин Алексей Васильевич**

Преподаватель:

**Инячина Диана Александровна**

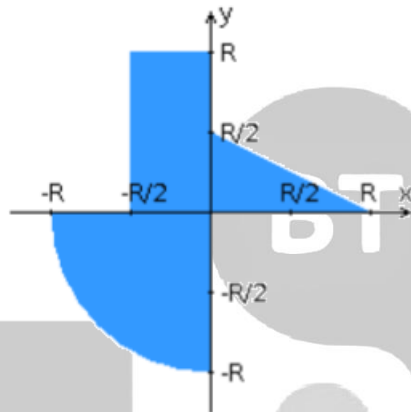
Санкт-Петербург

2023



**Задание: (порт = 21861)**

Разработать приложение на базе JavaServer Faces Framework, которое осуществляет проверку попадания точки в заданную область на координатной плоскости.



изменение X: `commandLink {'-5','-4','-3','-2','-1','0','1','2','3'}`

изменение Y: `inputText {-5 ... 3}`

изменение R: `commandLink {'1','1.5','2','2.5','3'}`

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 9 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 ORM EclipseLink.
- 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.

#### **Выполнение**

**([https://github.com/XVIIStarPlatinum/WEB\\_III\\_VAR\\_3552](https://github.com/XVIIStarPlatinum/WEB_III_VAR_3552))**

#### **Вывод:**

В ходе выполнения этой, с субъективной точки зрения, ужасной лабораторной работы я ознакомился с более сильной и гибкой реализации бэк-энда на джаве — JSF. Работа с таким фреймворком также привело к изучению таких новых понятий,

как ORM, Managed beans и JPA. Также мне удалось напрямую работать в практике с шаблонами проектирования, продвинутые способности CSS, базы данных вместо HTTP-сессии для хранения информации и его тестирование с помощью testcontainers на платформе docker. Сомневаюсь, что опыт, полученной от этой работы потребуется в практике, но это всё-таки важный опыт. Теперь будет самая актуальная работа – фреймворки.

