

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

Кафедра информатики и прикладной математики

## **Программирование интернет приложений**

Лабораторная работа 8

“Java Server Pages”

Вариант: 989



Проверил: **Гаврилов А.В.**

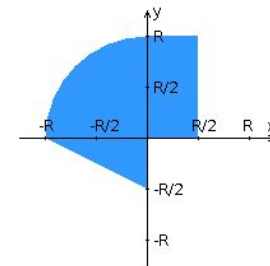
Старался: **Шкаруба Н.Е.**

Группа **Р3218**

2016г

Доработать приложение из [лабораторной работы №7](#) следующим образом:

- **ControllerServlet** по-прежнему должен определять по URL тип ресурса, к которому обращается клиент и делегировать управление соответствующему сервлету или JSP-странице.
- Вместо WebFormServlet нужно разработать **JSP-страницу**; список требований к ней приведён ниже.
- **AreaCheckServlet** по-прежнему должен осуществлять проверку попадания точки в область на координатной плоскости. Результат проверки должен помещаться в контекст приложения, HTTP-сессию или Bean-компонент в соответствии с вариантом задания.



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

изменение Y: Text {'-5 ... 3'}

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

метод HTTP: GET

список результатов сохранять в контексте приложения

**Разработанная JSP-страница должна содержать:**

1. "Шапку", содержащую ФИО студента, номер группы и номер варианта.
2. Форму, отправляющую данные на сервер.
3. Набор полей для задания координат точки и радиуса области в соответствии с вариантом задания.
4. Сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы.
5. Картинку, изображающую область на координатной плоскости (в соответствии с вариантом задания). Если радиус области установлен, клик курсором мыши по этой картинке должен обрабатываться JavaScript-функцией, определяющей координаты точки, по которой кликнул пользователь. В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.
6. Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения, HTTP-сессии или Bean-компонента в зависимости от варианта.

Разработанное веб-приложение необходимо развернуть на сервере [GlassFish](#).

## Результат:

Исходный код доступен на: [github.com/SigmaOne/ITMO](https://github.com/SigmaOne/ITMO)