

Национальный исследовательский университет ИТМО  
Факультет программной инженерии и компьютерной техники

## Веб-программирование Лабораторная работа №2

Работу выполнил:  
Бавыкин Роман  
Группа: Р3210  
Вариант 10201

Санкт-Петербург  
2021 г.

## 1. Текст задания.

Разработать веб-приложение на базе сервлетов и JSP, определяющее попадание точки на координатной плоскости в заданную область.

Приложение должно быть реализовано в соответствии с [шаблоном MVC](#) и состоять из следующих элементов:

- **ControllerServlet**, определяющий тип запроса, и, в зависимости от того, содержит ли запрос информацию о координатах точки и радиусе, делегирующий его обработку одному из перечисленных ниже компонентов. Все запросы внутри приложения должны передаваться этому сервлету (по методу GET или POST в зависимости от варианта задания), остальные сервлеты с веб-страниц напрямуюРазработать веб-приложение на базе сервлетов и JSP, определяющее попадание точки на координатной плоскости в заданную область.
- Приложение должно быть реализовано в соответствии с шаблоном MVC и состоять из следующих элементов:
- ControllerServlet, определяющий тип запроса, и, в зависимости от того, содержит ли запрос информацию о координатах точки и радиусе, делегирующий его обработку одному из перечисленных ниже компонентов. Все запросы внутри приложения должны передаваться этому сервлету (по методу GET или POST в зависимости от варианта задания), остальные сервлеты с веб-страниц напрямую вызываться не должны.
- AreaCheckServlet, осуществляющий проверку попадания точки в область на координатной плоскости и формирующий HTML-страницу с результатами проверки. Должен обрабатывать все запросы, содержащие сведения о координатах точки и радиусе области.
- Страница JSP, формирующая HTML-страницу с веб-формой. Должна обрабатывать все запросы, не содержащие сведений о координатах точки и радиусе области.
- Разработанная страница JSP должна содержать:
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Форму, отправляющую данные на сервер.
- Набор полей для задания координат точки и радиуса области в соответствии с вариантом задания.
- Сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы.
- Интерактивный элемент, содержащий изображение области на координатной плоскости (в соответствии с вариантом задания) и реализующий следующую функциональность:
- Если радиус области установлен, клик курсором мыши по изображению должен обрабатываться JavaScript-функцией, определяющей координаты точки, по которой кликнул пользователь и отправляющей полученные координаты на сервер для проверки факта попадания.

- В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.
- После проверки факта попадания точки в область изображение должно быть обновлено с учётом результатов этой проверки (т.е., на нём должна появиться новая точка).
- Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения, HTTP-сессии или Bean-компонента в зависимости от варианта.
- Страница, возвращаемая **AreaCheckServlet**, должна содержать:
  - Таблицу, содержащую полученные параметры.
  - Результат вычислений - факт попадания или непадения точки в область.
  - Ссылку на страницу с веб-формой для формирования нового запроса.
- Разработанное веб-приложение необходимо развернуть на сервере WildFly. Сервер должен быть запущен в standalone-конфигурации, порты должны быть настроены в соответствии с выданным portbase, доступ к http listener'у должен быть открыт для всех IP.
- ую вызываться не должны.
- **AreaCheckServlet**, осуществляющий проверку попадания точки в область на координатной плоскости и формирующий HTML-страницу с результатами проверки. Должен обрабатывать все запросы, содержащие сведения о координатах точки и радиусе области.
- **Страница JSP**, формирующая HTML-страницу с веб-формой. Должна обрабатывать все запросы, не содержащие сведений о координатах точки и радиусе области.

#### **Разработанная страница JSP должна содержать:**

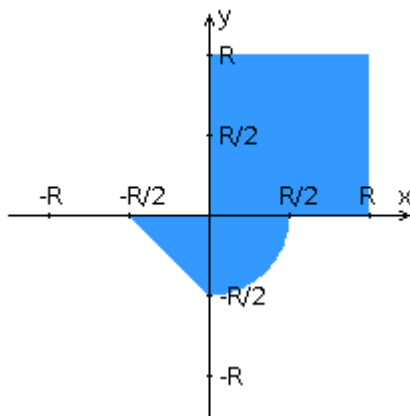
1. "Шапку", содержащую ФИО студента, номер группы и номер варианта.
2. Форму, отправляющую данные на сервер.
3. Набор полей для задания координат точки и радиуса области в соответствии с вариантом задания.
4. Сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы.
5. Интерактивный элемент, содержащий изображение области на координатной плоскости (в соответствии с вариантом задания) и реализующий следующую функциональность:
  - Если радиус области установлен, клик курсором мыши по изображению должен обрабатываться JavaScript-функцией, определяющей координаты точки, по которой кликнул пользователь и отправляющей полученные координаты на сервер для проверки факта попадания.
  - В противном случае, после клика по картинке должно выводиться сообщение о невозможности определения координат точки.

- После проверки факта попадания точки в область изображение должно быть обновлено с учётом результатов этой проверки (т.е., на нём должна появиться новая точка).
- 6. Таблицу с результатами предыдущих проверок. Список результатов должен браться из контекста приложения, HTTP-сессии или Bean-компонента в зависимости от варианта.

**Страница, возвращаемая AreaCheckServlet, должна содержать:**

1. Таблицу, содержащую полученные параметры.
2. Результат вычислений - факт попадания или непадения точки в область.
3. Ссылку на страницу с веб-формой для формирования нового запроса.

Разработанное веб-приложение необходимо развернуть на сервере [WildFly](#). Сервер должен быть запущен в standalone-конфигурации, порты должны быть настроены в соответствии с выданным portbase, доступ к http listener'у должен быть открыт для всех IP.



изменение X: Text {-3 ... 5}

изменение Y: Checkbox {'-3','-2','-1','0','1','2','3','4','5'}

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

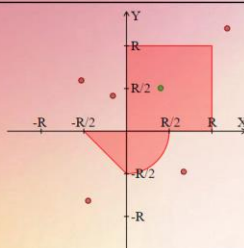
метод HTTP: GET

список результатов сохранять в HTTP-сессии

2. Исходный код программы:

<https://github.com/robqqq/Web2>

3. Разработанное Web-приложение:

X = Y = ☐ -3 ☐ -2 ☐ -1 ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5R = ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

X	Y	R	Result
3.54	3.611999885	3.0	false
-0.8	2.0700000762	5.0	false
2.68	-1.904000244	4.0	false
-0.9	-1.632000122	2.0	false
-2.12	2.376000061	4.0	false
1.6	2.016000061	4.0	true

#### 4. Выводы по работе.

Во время выполнения лабораторной работы узнал, что такое сервлеты и JSP, научился разрабатывать веб-приложения с помощью этих технологий. Изучил шаблон MVC и постарался реализовать его в своем приложении. JavaScript использовался для валидации значений на клиенте, как и в прошлой лабораторной работе, а также для динамического определения координат точки по клику на график. О JS я могу сделать вывод, что продолжение его использования заставляет меня ненавидеть его все больше.