

# Test Task 1

Данное тестовое задание должно в результате содержать работающее веб-приложение, удовлетворяющее всем нижеописанным требованиям, и качественный исходный код.

Задание должно быть выполнено полностью самостоятельно, не копируйте и не используйте код других соискателей.

Постарайтесь создать грамотную композицию классов и форматировать код в соответствии со стандартами Java.

Стандартный стиль форматирования Java: <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

## Описание задачи

Реализовать систему ввода и отображения информации о заказах Автомастерской, включающую следующие сущности и их атрибуты:

- Клиент
  - Имя
  - Фамилия
  - Отчество
  - Телефон
- Механик
  - Имя
  - Фамилия
  - Отчество
  - Почасовая оплата
- Заказ
  - Описание
  - Клиент
  - Механик
  - Дата создания
  - Дата окончания работ
  - Стоимость
  - Статус

Заказ может иметь один из статусов: Запланирован, Выполнен, Принят клиентом

Система должна иметь следующие функции:

- Отображение списка клиентов
- Добавление нового клиента, редактирование и удаление существующего
- Отображение списка механиков
- Отображение статистической информации по количеству заказов для каждого механика
- Добавление нового механика, редактирование и удаление существующего
- Отображения списка заказов
- Фильтрация списка заказов по описанию, статусу и клиенту
- Добавление нового заказа, редактирование и удаление существующего

## Требования к реализации

Используемые технологии:

- Java SE 8
- Пользовательский интерфейс на Vaadin 8 (<https://vaadin.com>)
- Доступ к данным через JDBC или JPA
- Не используйте JPAContainer из Vaadin. Ваш код доступа к данным должен быть абстрагирован от UI.
- Сервер баз данных: HSQLDB в in-process режиме ([http://hsqldb.org/doc/2.0/guide/running-chapt.html#rgc\\_inprocess](http://hsqldb.org/doc/2.0/guide/running-chapt.html#rgc_inprocess))

Рекомендуется использовать шаблон проекта <https://github.com/Haulmont/test-task>

Требования к пользовательскому интерфейсу:

- Экраны системы:
  - Список клиентов: таблица, кнопки "Добавить", "Изменить", "Удалить"
  - Редактирование клиента: поля ввода атрибутов, кнопки "ОК", "Отменить"
  - Список механиков: таблица, кнопки "Добавить", "Изменить", "Удалить", "Показать статистику"
  - Редактирование механика: поля ввода атрибутов, кнопки "ОК", "Отменить"
  - Список заказов с фильтром: таблица, кнопки "Добавить", "Изменить", "Удалить", панель фильтра с полями "Клиент", "Статус", "Описание" и кнопкой "Применить"
  - Фильтр по описанию предполагает вывод заказов, содержащих в описании введенный в фильтр текст
  - Редактирование заказа: поля ввода атрибутов, кнопки "ОК", "Отменить"
- Экраны редактирования должны быть модальными
- Все формы ввода должны валидировать данные в соответствии с их типом и допустимыми значениями

Другие требования:

- Код доступа к данным должен быть изолирован в классах DAO.
- Каждая таблица должна иметь первичный ключ типа Long.
- Система должна иметь защиту на уровне БД от удаления клиента и механика, для которых существуют заказы
- Готовое приложение должно содержать:
  - Каталог с исходными текстами и проектным файлом Maven (pom.xml)
  - SQL-скрипт создания таблиц в базе данных в виде отдельного файла проекта (Примечание: требуется SQL скрипт создания всех необходимых объектов в БД, написанный вручную)
  - Файлы базы данных HSQLDB.Приложение должно подключаться к БД без необходимости старта отдельного сервера
- Для запуска приложения должно быть достаточно собрать его при помощи Maven, как описано для шаблона <https://github.com/Ha ulmont/test-task>.