Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

*Дисциплина «Сервис-ориентированная архитектура»*

**Лабораторная работа №1**

Вариант 1706

Группа: P3417

Выполнил: Ореховский А.

Проверил: Усков И. В.

Санкт-Петербург,

2020г

# Задание

Разработать веб-сервис на базе сервлета, реализующий управление коллекцией объектов, и клиентское веб-приложение, предоставляющее интерфейс к разработанному веб-сервису. В коллекции необходимо хранить объекты класса Product, описание которого приведено ниже:

**public class** Product {  
 */\*\*  
 \* Поле идентификатора  
 \* Поле не может быть null  
 \* Значение поля должно быть больше 0  
 \* Значение этого поля должно быть уникальным  
 \* Значение этого поля должно генерироваться автоматически  
 \*/* **private** Long **id**;  
  
 */\*\*  
 \* Поле имени  
 \* Поле не может быть null  
 \* Строка не может быть пустой  
 \*/* **private** String **name**;  
  
 */\*\*  
 \* Поле координат  
 \** ***@see*** *Coordinates  
 \* Поле не может быть null  
 \*/* **private** Coordinates **coordinates**;  
  
 */\*\*  
 \* Поле даты создания  
 \* Поле не может быть null  
 \* Значение этого поля должно генерироваться автоматически  
 \*/* **private** LocalDateTime **creationDate**;  
  
 */\*\*  
 \* Поле цены  
 \* Поле может быть null  
 \* Значение поля должно быть больше 0  
 \*/* **private** Integer **price**;  
  
 */\*\*  
 \* Поле номера части  
 \* Длина строки должна быть не меньше 27  
 \* Значение этого поля должно быть уникальным  
 \* Поле может быть null  
 \*/* **private** String **partNumber**;  
  
 */\*\*  
 \* Поле цены производителя  
 \*/* **private double manufactureCost**;  
  
 */\*\*  
 \* Поле еденицы измерения  
 \** ***@see*** *UnitOfMeasure  
 \* Поле может быть null  
 \*/* **private** UnitOfMeasure **unitOfMeasure**;  
  
 */\*\*  
 \* Поле владельца  
 \** ***@see*** *Person  
 \* Поле может быть null  
 \*/* **private** Person **owner**;

}

**public class** Coordinates {  
 */\*\*  
 \* Поле x  
 \* Значение поля должно быть больше -965  
 \*/* **private int x**;  
  
 */\*\*  
 \* Поле y  
 \*/* **private long y**;

}

**public class** Person {  
 */\*\*  
 \* Поле имени  
 \* Поле не может быть null  
 \* Строка не может быть пустой  
 \*/* **private** String **name**;  
  
 */\*\*  
 \* Поле веса  
 \* Поле может быть null  
 \* Значение поля должно быть больше 0  
 \*/* **private** Integer **weight**;  
  
 */\*\*  
 \* Поле национальности  
 \** ***@see*** *Country  
 \* Поле не может быть null  
 \*/* **private** Country **nationality**;  
  
 */\*\*  
 \* Поле местоположения  
 \** ***@see*** *Location  
 \* Поле не может быть null  
 \*/* **private** Location **location**;

}

**public class** Location {  
 */\*\*  
 \* Поле x  
 \*/* **private float x**;  
  
 */\*\*  
 \* Поле y  
 \* Поле не может быть null  
 \*/* **private** Double **y**;  
  
 */\*\*  
 \* Поле z  
 \*/* **private double z**;

}

**public enum** UnitOfMeasure {  
 ***CENTIMETERS***,  
 ***GRAMS***,  
 ***MILLIGRAMS***;  
}

**public enum** Country {  
 ***USA***,  
 ***CHINA***,  
 ***INDIA***,  
 ***THAILAND***;  
}

## Веб-сервис должен удовлетворять следующим требованиям:

* API, реализуемый сервисом, должен соответствовать рекомендациям подхода RESTful.
* Необходимо реализовать следующий базовый набор операций с объектами коллекции: добавление нового элемента, получение элемента по ИД, обновление элемента, удаление элемента, получение массива элементов.
* Операция, выполняемая над объектом коллекции, должна определяться методом HTTP-запроса.
* Операция получения массива элементов должна поддерживать возможность сортировки и фильтрации по любой комбинации полей класса, а также возможность постраничного вывода результатов выборки с указанием размера и порядкового номера выводимой страницы.
* Все параметры, необходимые для выполнения операции, должны передаваться в URL запроса.
* Данные коллекции, которыми управляет веб-сервис, должны храниться в реляционной базе данных.
* Информация об объектах коллекции должна передаваться в формате **xml**.
* В случае передачи сервису данных, нарушающих заданные на уровне класса ограничения целостности, сервис должен возвращать код ответа http, соответствующий произошедшей ошибке.
* Веб-сервис должен быть "упакован" в веб-приложение, которое необходимо развернуть на сервере приложений **Tomcat**.

## Помимо базового набора, веб-сервис должен поддерживать следующие операции над объектами коллекции:

* Рассчитать среднее значение поля manufactureCost для всех объектов.
* Удалить все объекты, значение поля owner которого эквивалентно заданному.
* Удалить один (любой) объект, значение поля price которого эквивалентно заданному.

Эти операции должны размещаться на отдельных URL.

## Требования к клиентскому приложению:

* Клиентское приложение может быть написано на любом веб-фреймворке, который можно запустить на сервере helios.
* Клиентское приложение должно обеспечить полный набор возможностей по управлению объектами коллекции, предоставляемых веб-сервисом -- включая сортировку, фильтрацию и постраничный вывод.
* Клиентское приложение должно преобразовывать передаваемые сервисом данные в человеко-читаемый вид -- параграф текста, таблицу и т. д.
* Клиентское приложение должно информировать пользователя об ошибках, возникающих на стороне сервиса, в частности, о том, что сервису были отправлены невалиданые данные.

Веб-сервис и клиентское приложение должны быть развёрнуты на сервере helios.

# Код сервиса

## Back-end

Серверная часть кода доступна по ссылке:

<https://github.com/sempaw/soa.lab1>

## Front-end

Клиентская часть кода доступна по ссылке:

<https://github.com/sempaw/itmo.soa.lab1.client>

# Доступ к приложению

Для корректной работы клиентского приложения необходимо пробросить порт:

> ssh -fNL 6780:localhost:6780 s225123@helios.cs.ifmo.ru -p 2222

После успешного проброса портов приложение доступно по адресу <http://localhost:6780/>

# Итоги

По итогам данной ЛР я разработал и задеплоил приложение на гелиосе. Повторил принцип разработки веб-приложений на базе Java Servlet-ов, а также работу с Angular. Среди прочего я повторил правила хорошего тона REST-ful подхода, а также конвенцию по именованию URI. «Открытием» для меня стали CriteriaQuery-запросы, которые поддерживает Hibernate, хотя, как по мне, Linq-запросы гораздо более удобны; как жаль, что их нет в Java.