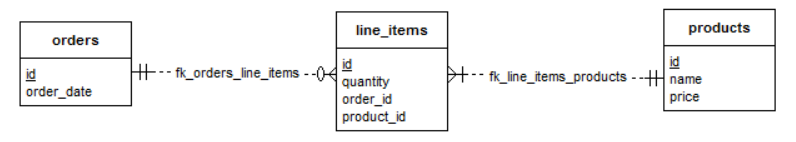
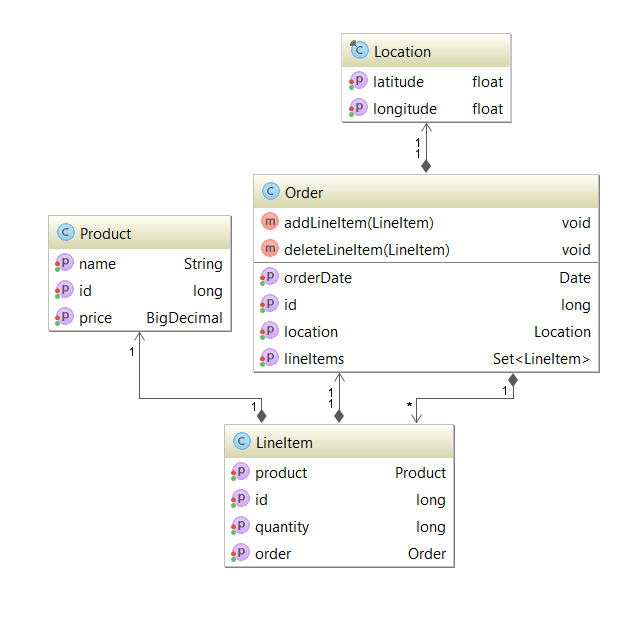
# Exercises: Online Shop

This document defines the **exercise assignments** for the ["Databases Advanced – Hibernate" course @ Software University.](https://softuni.bg/trainings/1444/databases-advanced-hibernate-october-2016)

We have to create an online shop with the following structure:



## Create Models



We have the following classes:

### **Order – Aggregation Root**

@Entity  
@Table(name = **"orders"**)  
**public class** Order **implements** Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.***IDENTITY***)  
 **private long id**;  
  
 @Column(name = **"order\_date"**)  
 **private** Date **orderDate**;  
  
 @OneToMany(mappedBy = **"order"**, targetEntity = LineItem.**class**, cascade = CascadeType.***ALL***)  
 **private** Set<LineItem> **lineItems**;  
  
 @Transient  
 **private** Location **location**;  
  
 **public** Order() {  
 **this**.setLineItems(**new** HashSet<>());  
 }

}

### **Line Item – Entity**

@Entity  
@Table(name = **"line\_items"**)  
**public class** LineItem {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.***IDENTITY***)  
 **private long id**;  
  
 @ManyToOne(cascade = CascadeType.***ALL***)  
 @JoinColumn(name = **"product\_id"**, referencedColumnName = **"id"**)  
 **private** Product **product**;  
  
 @Column(name = **"quantity"**)  
 **private long quantity**;  
  
 @ManyToOne  
 @JoinColumn(name = **"order\_id"**, referencedColumnName = **"id"**)  
 **private** Order **order**;  
  
 **public** LineItem() {  
 }

}

### **Product – Entity**

@Entity  
@Table(name = **"products"**)  
**public class** Product **implements** Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.***IDENTITY***)  
 **private long id**;  
  
 @Column(name = **"name"**)  
 **private** String **name**;  
  
 @Column(name = **"price"**)  
 **private** BigDecimal **price**;  
  
 **public** Product() {  
 }

}

### **Location – Value Object**

**public final class** Location {  
  
 **private float latitude**;  
  
 **private float longitude**;  
  
 **public** Location() {  
 Random randomGenerator = **new** Random();  
 **this**.**latitude** = randomGenerator.nextFloat();  
 **this**.**longitude** = randomGenerator.nextFloat();  
 }

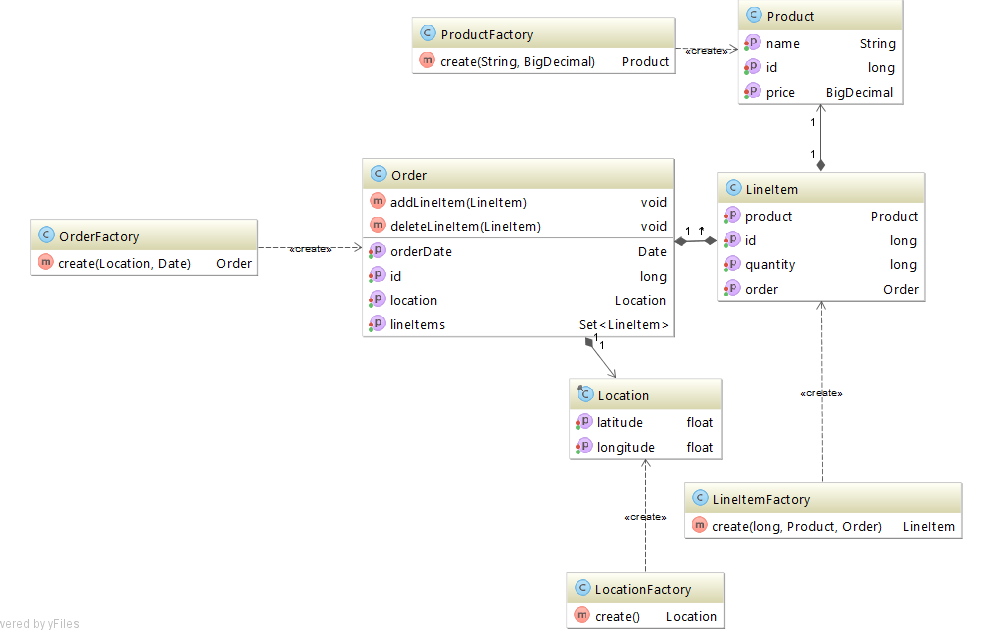
}

## Create Dto

### **ProductDto**

**public class** ProductDto {  
  
 **private** String **name**;  
  
 **private** BigDecimal **price**;  
  
 **private long quantity**;  
}

## Create Factories



### **Order Factory**

**public class** OrderFactory {  
  
 **public** Order create(Location location, Date orderDate){  
 Order order = **new** Order();  
 order.setLocation(location);  
 order.setOrderDate(orderDate);  
 **return** order;  
 }  
}

### **LineItem Factory**

**public class** LineItemFactory {  
  
 **public** LineItem create(**long** quantity, Product product, Order order){  
 LineItem lineItem = **new** LineItem();  
 lineItem.setProduct(product);  
 lineItem.setOrder(order);  
 lineItem.setQuantity(quantity);  
 **return** lineItem;  
 }  
}

### **Product Factory**

**public class** ProductFactory {  
  
 **public** Product create(String name, BigDecimal price){  
 Product product = **new** Product();  
 product.setName(name);  
 product.setPrice(price);  
 **return** product;  
 }  
}

### **Location Factory**

**public class** LocationFactory {  
  
 **public** Location create(){  
 **return new** Location();  
 }  
}

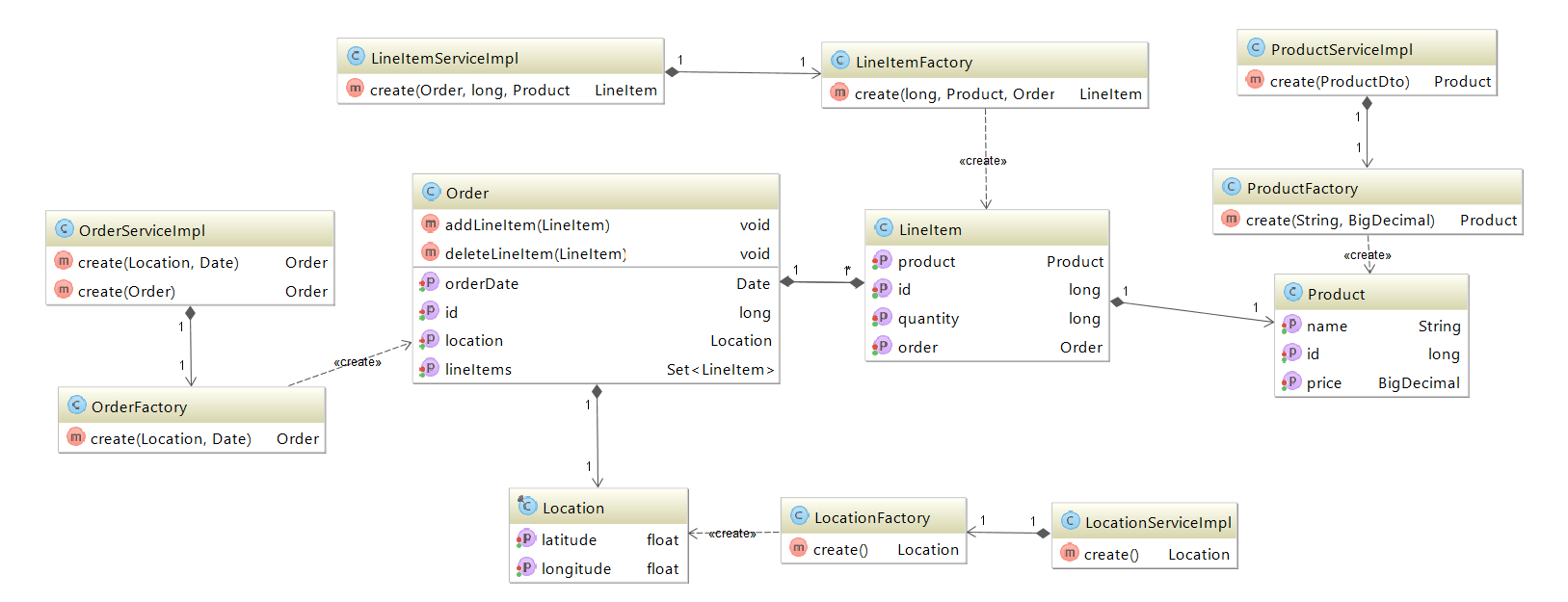
### **Create Beans**

@Configuration  
**public class** JavaConfig {  
  
 @Bean  
 **public** ProductFactory productFactory(){  
 **return new** ProductFactory();  
 }  
  
 @Bean  
 **public** LineItemFactory lineItemFactory(){  
 **return new** LineItemFactory();  
 }  
  
 @Bean  
 **public** OrderFactory orderFactoryFactory(){  
 **return new** OrderFactory();  
 }  
  
 @Bean  
 **public** LocationFactory locationFactory(){  
 **return new** LocationFactory();  
 }  
}

## Create Repository

@Repository  
**public interface** OrderRepository **extends** JpaRepository<Order, Long>{  
}

## Create Services



### **Order Service**

**public interface** OrderService {  
  
 Order create(Location location, Date date);  
  
 Order create(Order order);  
}

### **LineItem Service**

**public interface** LineItemService {  
  
 LineItem create(Order order, **long** quantity, Product product);  
}

### **Product Service**

**public interface** ProductService {  
  
 Product create(ProductDto productDto);  
}

### **Location Service**

**public interface** LocationService {  
  
 Location create();  
}

## Create Controller

@Controller  
**public class** ShopController {  
  
 @Autowired  
 **private** LocationService **locationService**;  
  
 @Autowired  
 **private** ProductService **productService**;  
  
 @Autowired  
 **private** LineItemService **lineItemService**;  
  
 @Autowired  
 **private** OrderService **orderService**;  
  
 @GetMapping(**"/shop"**)  
 **public** String shopForm(Model model) {  
 model.addAttribute(**"productDto"**, **new** ProductDto());  
 **return "shop"**;  
 }  
  
 @PostMapping(**"/shop"**)  
 **public** String shopSubmit(@ModelAttribute ProductDto productDto) {  
 Order order = **this**.registerOrder();  
 Product product = **this**.createProduct(productDto);  
 **long** quantity = productDto.getQuantity();  
 **this**.addLineItem(order, quantity , product);  
 **return "shop"**;  
 }  
  
 **private** Order registerOrder() {  
 Location location = **this**.**locationService**.create();  
 Order order = **this**.**orderService**.create(location, **new** Date());  
 **return this**.**orderService**.create(order);  
 }  
  
 **private** Product createProduct(ProductDto productDto) {  
 Product product = **this**.**productService**.create(productDto);  
 **return** product;  
 }  
  
 **private void** addLineItem(Order order, **long** quantity, Product product) {  
 LineItem lineItem = **this**.**lineItemService**.create(order, quantity, product);  
 order.addLineItem(lineItem);  
 **this**.**orderService**.create(order);  
 }  
}

## Create View

### **Dependency**

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-thymeleaf</**artifactId**>  
</**dependency**>

### **View**

<!DOCTYPE **HTML**>  
<**html xmlns:th="http://www.thymeleaf.org"**>  
<**head**>  
 <**title**>Getting Started: Serving Web Content</**title**>  
 <**meta http-equiv="Content-Type" content="text/html; charset=UTF-8"** />  
</**head**>  
 <**body**>  
 <**p th:text="'Shop Page'"** />  
 <**br**/>  
  
 <**form action="#" th:action="@{/shop}" th:object="${productDto}" method="post"**>  
 <**p**>Product Name: <**input type="text" th:field="\*{name}"** /></**p**>  
 <**p**>Product Price: <**input type="text" th:field="\*{price}"**/></**p**>  
 <**p**>Product Quantity: <**input type="text" th:field="\*{quantity}"**/></**p**>  
 <**p**><**input type="submit" value="Submit"** /></**p**>  
 </**form**>  
 </**body**>  
</**html**>