

```

package com.example.springsecurityapplication.models;

import org.aspectj.weaver.ast.Or;

import javax.persistence.*;
import javax.validation.constraints.Min;
import javax.validation.constraints.NotEmpty;
import javax.validation.constraints.NotNull;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;

@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "title", nullable = false, columnDefinition = "text",
unique = true)
    @NotEmpty(message = "Наименование товара не может быть пустым")
    private String title;

    @Column(name = "description", nullable = false, columnDefinition =
"text")
    @NotEmpty(message = "Описание товара не может быть пустым")
    private String description;

    @Column(name = "price", nullable = false)
    @Min(value = 1, message = "Цена не может быть отрицательной или
нулевой")
    @NotNull(message = "Цена товара не может быть отрицательной или
нулевой")
    private float price;

    @Column(name = "warehouse", nullable = false)
    @NotEmpty(message = "Склад по нахождению товара не может быть
пустым")
    private String warehouse;

    @Column(name = "seller", nullable = false)
    @NotEmpty(message = "Продавец товара не может быть пустым")
    private String seller;

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY,
mappedBy = "product")
    private List<Image> imageList = new ArrayList<>();

    @ManyToOne(optional = false)
    private Category category;

    @ManyToMany()
    @JoinTable(name = "product_cart", joinColumns = @JoinColumn(name =
"product_id"), inverseJoinColumns = @JoinColumn(name = "person_id"))
    private List<Person> personList;

    @OneToMany(mappedBy = "product")
    private List<Order> orderList;

    private LocalDateTime dateTimeOfCreated;

```

```
// Будет заполняться дата и время при создании объекта класса
@PrePersist
private void init(){
    dateTimeOfCreated = LocalDateTime.now();
}

public void addImageProduct(Image image){
    image.setProduct(this);
    imageUrl.add(image);
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public float getPrice() {
    return price;
}

public void setPrice(float price) {
    this.price = price;
}

public String getWarehouse() {
    return warehouse;
}

public void setWarehouse(String warehouse) {
    this.warehouse = warehouse;
}

public String getSeller() {
    return seller;
}

public void setSeller(String seller) {
    this.seller = seller;
}
```

```
    public Product(String title, String description, float price, String
warehouse, String seller) {
        this.title = title;
        this.description = description;
        this.price = price;
        this.warehouse = warehouse;
        this.seller = seller;
    }

    public Product() {
    }

    public List<Image> getImageList() {
        return imageList;
    }

    public void setImageList(List<Image> imageList) {
        this.imageList = imageList;
    }

    public LocalDateTime getDateTimeOfCreated() {
        return dateTimeOfCreated;
    }

    public void setDateTimeOfCreated(LocalDateTime dateTimeOfCreated) {
        this.dateTimeOfCreated = dateTimeOfCreated;
    }

    public Category getCategory() {
        return category;
    }

    public void setCategory(Category category) {
        this.category = category;
    }
}
```