```
package com.example.springsecurityapplication.controllers;
com.example.springsecurityapplication.repositories.ProductRepository;
import com.example.springsecurityapplication.services.ProductService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
@Controller
@RequestMapping("/product")
public class MainController {
    private final ProductRepository productRepository;
    private final ProductService productService;
    @Autowired
    public MainController (ProductRepository productRepository,
ProductService productService) {
        this.productRepository = productRepository;
        this.productService = productService;
    // Данный метод предназначен для отображении товаров без прохождения
аутентификации и авторизации
    @GetMapping("")
    public String getAllProduct(Model model) {
        model.addAttribute("products", productService.getAllProduct());
        return "product/product";
    @GetMapping("/info/{id}")
    public String infoProduct(@PathVariable("id") int id, Model model) {
        model.addAttribute("product", productService.getProductId(id));
        return "product/infoProduct";
    }
    @PostMapping("/search")
    public String productSearch(@RequestParam("search") String search,
@RequestParam("ot") String ot, @RequestParam("do") String Do,
@RequestParam(value = "price", required = false, defaultValue = "")
String price, @RequestParam(value = "category", required = false,
defaultValue = "") String category, Model model){
        if(!ot.isEmpty() & !Do.isEmpty()) {
            // Если сортировка по цене выбрана
            if (!price.isEmpty()) {
                // Если в качестве сортировки выбрана сортировкам по
возрастанию
                if (price.equals("ascending price")) {
                    // Если категория товара не пустая
                    if (!category.isEmpty()) {
                        // Если категория равная мебели
                        if (category.equals("CPU")) {
                            model.addAttribute("search product",
productRepository.findByTitleAndCategoryOrderByPrice(search.toLowerCase()
, Float.parseFloat(ot), Float.parseFloat(Do), 1));
                            // Если категория равная бытовой техники
                        } else if (category.equals("GPU")) {
```

```
model.addAttribute("search product",
productRepository.findByTitleAndCategoryOrderByPrice(search.toLowerCase()
, Float.parseFloat(ot), Float.parseFloat(Do), 2));
                            // Если категория равная одежде
                        } else if (category.equals("motherboard")) {
                            model.addAttribute("search product",
productRepository.findByTitleAndCategoryOrderByPrice(search.toLowerCase()
, Float.parseFloat(ot), Float.parseFloat(Do), 3));
                        // Если категория не выбрана
                    } else {
                        model.addAttribute("search product",
productRepository.findByTitleOrderByPrice(search.toLowerCase(),
Float.parseFloat(ot), Float.parseFloat(Do)));
                    // Если в качестве сортировки выбрана сортировкам по
убыванию
                } else if (price.equals("descending price")) {
                    // Если категория не пустая
                    if (!category.isEmpty()) {
                        // Если категория равная мебели
                        if (category.equals("CPU")) {
                            model.addAttribute("search product",
productRepository.findByTitleAndCategoryOrderByPriceDesc(search.toLowerCa
se(), Float.parseFloat(ot), Float.parseFloat(Do), 1));
                            // Если категория равная бытовой техники
                        } else if (category.equals("GPU")) {
                            model.addAttribute("search_product",
productRepository.findByTitleAndCategoryOrderByPriceDesc(search.toLowerCa
se(), Float.parseFloat(ot), Float.parseFloat(Do), 2));
                            // Если категория равная одежде
                        } else if (category.equals("motherboard")) {
                            model.addAttribute("search product",
productRepository.findByTitleAndCategoryOrderByPriceDesc(search.toLowerCa
se(), Float.parseFloat(ot), Float.parseFloat(Do), 3));
                        // Если категория не выбрана
                    else {
                        model.addAttribute("search product",
productRepository.findByTitleOrderByPriceDesc(search.toLowerCase(),
Float.parseFloat(ot), Float.parseFloat(Do)));
            }
            else {
                model.addAttribute("search product",
productRepository.findByTitleAndPriceGreaterThanEqualAndPriceLessThan(sea
rch.toLowerCase(), Float.parseFloat(ot), Float.parseFloat(Do)));
            }
        } else {
model.addAttribute("search product",productRepository.findByTitleContaini
ngIgnoreCase(search));
        model.addAttribute("value search", search);
        model.addAttribute("value_price_ot", ot);
        model.addAttribute("value price do", Do);
        model.addAttribute("products", productService.getAllProduct());
```

```
return "/product/product";
}
```