



```

        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 сортировка 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.toLowerCase()
, Float.parseFloat(ot), Float.parseFloat(Do), 1));
            }
            // Если категория равная бытовой техники
            } else if (category.equals("GPU")) {
                model.addAttribute("search_product",
productRepository.findByTitleAndCategoryOrderByPriceDesc(search.toLowerCase()
, Float.parseFloat(ot), Float.parseFloat(Do), 2));
            }
            // Если категория равная одежде
            } else if (category.equals("motherboard")) {
                model.addAttribute("search_product",
productRepository.findByTitleAndCategoryOrderByPriceDesc(search.toLowerCase()
, 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";  
    }  
}
```