1 后台

注意：

前端参数校验

工具：

DBUtil工具使用 new BeanListHandler<>(Category.class)，

连接池技术，使用 DruidUtils类编写

BeanUtils

技术：

LoadOnServlet

分页实现，

批量删除，

filter

文件上传

* 1. 系统用户管理

优化

servlet采用循环deleteOne的方式，

update功能的回显

1.2 目录管理

1.3 商品管理

LoadOnServlet

修改商品，上传图片没有修改的情况

把搜索条件存入session域，下次打开时记住

难点

多条件+分页 查询

1.4 注册用户管理

免邮箱验证

1. 前台
   1. 主页

列出top和hot商品（根据价格）

模糊搜索，支持搜索商品分类或商品名称

查询购物车getCart()操作ResultSetHandler,因为要封装List<CartItem>对象，里面有Product对象，对象套对象

* 1. 登录

验证码

2.3 注册

Ajax验证用户名是否重复

邮箱验证（未验证邮箱不可加购物车）

* 1. 购物车

前端计算总件数、总价格

方便的在购物车中增减数量，而且只有一件的时候是删除该商品

* 1. 下单

操作

1 验证库存

引入事务2~5

@Override

public int insertOrder(Order order) {

try {

QueryRunner runnerSameConnection = new QueryRunner();

// int i = 1 / 0;

runnerSameConnection.update(DruidUtils.getConnection(true),"insert into t\_order values(null, ?, ?, ?, ?, ?, ?, ?, ?, ?)",

order.getOrderNum(), order.getTotalPrice(), order.getReceiverName(), order.getReceiverMobile(),

order.getReceiverAddress(), order.getPayStatus(), order.getOrderTime(),

order.getUid(), order.getUsername());

} catch (Exception e) {

e.printStackTrace();

return 0;

}

return 1;

}

2新建Order

3 拷贝OrderItem

4 更改totalStockCount

5 删除勾选的selectedCartItems

6 查询orderList

bug

可以在任意步骤出现异常时回滚，和修改库存不被锁住，有冲突