销售 业务逻辑层

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Sale.calculate | 语法 | public Order calculate() | |
| 前置条件 | 顾客进入购物车，对购物车中的图书进行结算 | |
| 后置条件 | 返回相应的订单 | |
| Sale.putInCart | 语法 | public ResultMessage putInCart(String id) | |
| 前置条件 | 顾客找到所需的图书，并加入购物车 | |
| 后置条件 | 将图书加入购物车 | |
| Sale.enterCart | 语法 | public void enterCart() | |
| 前置条件 | 顾客找到所有需要购买的图书，准备结算 | |
| 后置条件 | 系统允许顾客进入购物车进行操作 | |
| Sale.purchase | 语法 | public ResultMessage purchase(Order order) | |
| 前置条件 | 顾客已经进行过结算，系统已经生成订单 | |
| 后置条件 | 结束此次购买，持久化更新设计的领域对象的数据 | |
| Sale.chooseEquivalent | 语法 | public double chooseEquivalent (String id) | |
| 前置条件 | 顾客拥有等价券，并且已经进入购物车 | |
| 后置条件 | 重新计算图书总价 | |
| Sale.chooseCupon | 语法 | public double chooseCoupon (String id) | |
| 前置条件 | 顾客拥有折价券，并且已经进入购物车 | |
| 后置条件 | 重新计算图书总价 | |
| Sale.removeFromCart | 语法 | public void removeFromCart(String id) | |
| 前置条件 | 顾客已经进入购物车 | |
| 后置条件 | 从购物车删除所选图书 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| DataService.findBook(String id) | | | 根据ID进行查找图书 |
| DataService.findEquivalent(String id) | | | 根据ID进行查找等价券 |
| DataService.findCupon(String id) | | | 根据ID进行查找折价券 |
| DataService.update | | | 更新单一持久化对象 |
| DataService.insert | | | 插入单一持久化对象 |
| DataService.delete | | | 删除单一持久化对象 |
| DatabaseFactory.getMemberDatabase | | | 得到Member数据库的服务的引用 |
| DatabaseFactory.getBookDatabase | | | 得到Book数据库的服务的引用 |

会员Member 业务逻辑

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Member.find | 语法 | public Member find(String id) | |
| 前置条件 | 销售经理已经登录，且输入符合规则 | |
| 后置条件 | 返回相应的会员信息 | |
| Member.addEquivalent | 语法 | public ResultMessage addEquivalent(Equivalent e) | |
| 前置条件 | 已经制定了等价券赠送规则，并且输入符合规则 | |
| 后置条件 | 增加会员的等价券信息 | |
| Member.addCoupon | 语法 | public ResultMessage addCoupon(Coupon c) | |
| 前置条件 | 已经制定了折价券赠送规则，并且输入符合规则 | |
| 后置条件 | 增加会员的折价券信息 | |
| Member.addFavorities | 语法 | public ResultMessage addFavorities(Book book) | |
| 前置条件 | 顾客找到想收藏的图书 | |
| 后置条件 | 将图书加入收藏夹 | |
| Member.getOrderList | 语法 | public ArrayList<Order> getOrderList() | |
| 前置条件 | 销售经理已经登录 | |
| 后置条件 | 返回所有的订单列表 | |
| Member.changeOrder | 语法 | public ResultMessage changeOrder(State state) | |
| 前置条件 | 销售经理已经登录，并且输入符合规则 | |
| 后置条件 | 修改订单状态 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| DataService.findMember(String id) | | | 根据ID进行查找会员 |
| DataService.findEquivalent(String id) | | | 根据ID进行查找等价券 |
| DataService.findCupon(String id) | | | 根据ID进行查找折价券 |
| DataService.update | | | 更新单一持久化对象 |
| DataService.insert | | | 插入单一持久化对象 |
| DataSerivce.delete | | | 删除单一持久化对象 |
| DatabaseFactory.getBookDatabase | | | 得到Book数据库的服务的引用 |
| DatabaseFactory.getPromotionDatabase | | | 得到Promotion数据库的服务的引用 |

图书Book 业务逻辑层

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Book.findByKey | 语法 | public ArrayList<Book> find(String name) | |
| 前置条件 | 顾客已经登录，且输入符合规则 | |
| 后置条件 | 返回相应的图书列表 | |
| Book.findByType | 语法 | public ArrayList<Book> findByType(String type) | |
| 前置条件 | 顾客已经登录，且输入符合规则 | |
| 后置条件 | 返回相应的图书列表 | |
| Book.add | 语法 | public ResultMessage add(Book b) | |
| 前置条件 | 已经正确输入了新增图书的信息 | |
| 后置条件 | 添加新增图书 | |
| Book.delete | 语法 | public ResultMessage delete(String id) | |
| 前置条件 | 已经正确输入了待删除图书的信息 | |
| 后置条件 | 将图书删除 | |
| Book.modify | 语法 | public Book modify(String id) | |
| 前置条件 | 已经正确输入了待修改图书的信息 | |
| 后置条件 | 返回该图书的所有信息 | |
| Member.changeOrder | 语法 | public ResultMessage changeOrder(State state) | |
| 前置条件 | 销售经理已经登录，并且输入符合规则 | |
| 后置条件 | 修改订单状态 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| DataService.find (String id) | | | 根据ID进行查找图书 |
| DataService.findByType(String type) | | | 根据类别进行查找图书 |
| DataService.findByKey(String key) | | | 根据关键字进行查找图书 |
| DataService.update | | | 更新单一持久化对象 |
| DataService.insert | | | 插入单一持久化对象 |
| DataSerivce.delete | | | 删除单一持久化对象 |

数据层模块接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| DateService.find | 语法 | public PO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 根据ID进行查找返回相应的po结果 |
| DataService.insert | 语法 | public void insert(PO po) throws RemoteException |
| 前置条件 | 同样ID的PO在Mapper中不存在 |
| 后置条件 | 在Mapper增加一个记录 |
| DataService.delete | 语法 | public void delete(PO po) throws RemoteException |
| 前置条件 | 在Mapper中存在同样ID的po |
| 后置条件 | 删除一个po |
| DataService.update | 语法 | public void update(PO po) throws RemoteException |
| 前置条件 | 在Mapper中存在同样ID的po |
| 后置条件 | 更新一个po |
| DataService.init | 语法 | public void init()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| DataService.finish | 语法 | public void finish()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |