### 4.1.1Searchbl模块

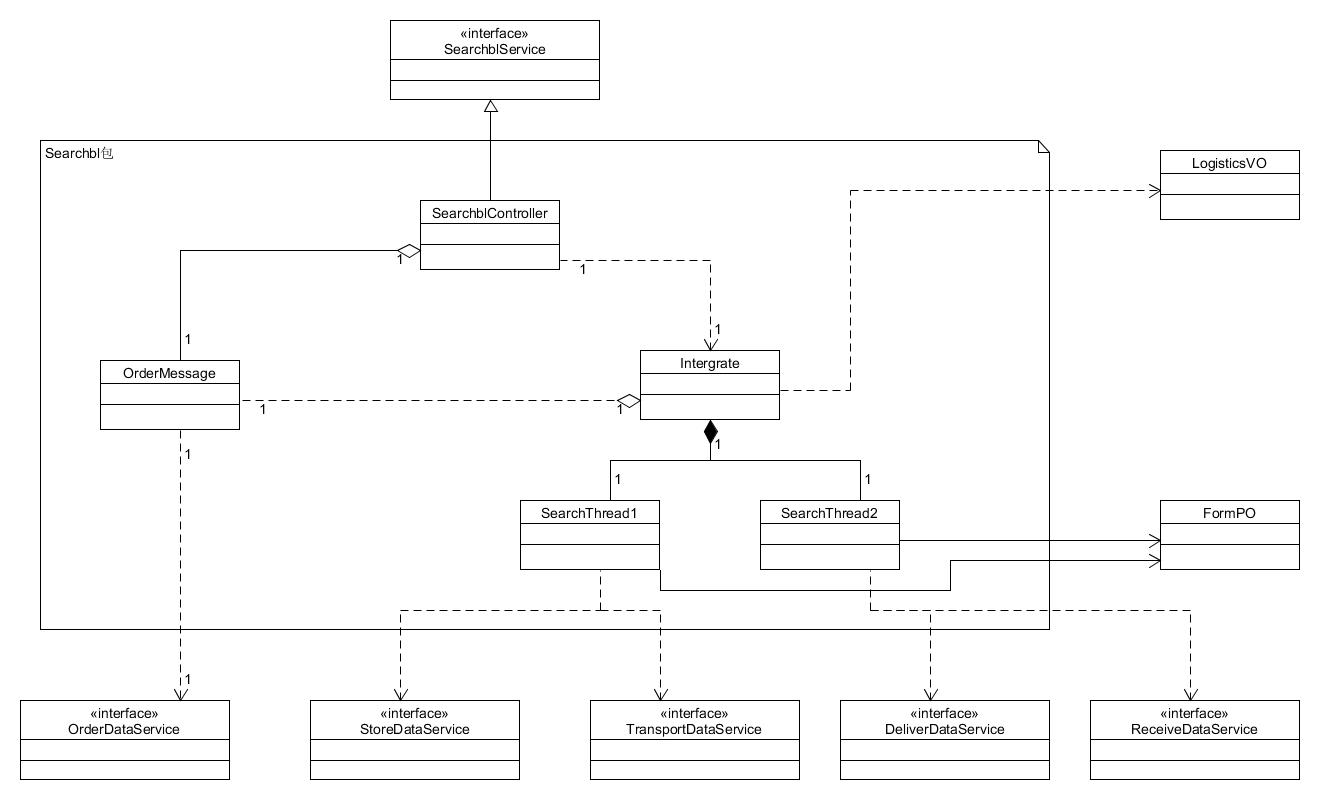
1. 模块概述

模块负责按照用户输入的订单号来查询订单状态，具体功能需求和非功能需求可参见需求规格说明文档和体系结构设计文档。

1. 整体结构

由于该模块逻辑简单，就采用了集中控制的风格，为了追求响应速度，采用了多线程的方式。该模块会首先确认订单是否存在，之后会爆发出多线程来搜索、整合信息，事实上可能存在冲突、死锁等问题

模块的设计如图



模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| SearchblController | 负责控制、安排搜索的进度与结果 |
| OrderMessage | 负责确认、获取订单信息（我认为用户输入错订单的可能性较大，所以特地增加审核订单号的过程） |
| Intergrate | 负责搜素、整合信息，并构建LogisticsVO |

1. 模块内部类的接口规范

**SearchblController的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SearchblController.  searchOrder | 语法 | Public LogisticsVO SearchOrder (String OrderID) |
| 前置条件 | 订单号格式正确 |
| 后置条件 | 对订单信息搜索；若找到，返回整合后的LogisticsVO，否则返回null |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderMessage.checkOrderID(String OrderID) | 查询ID是否存在 | |
| Intergrate.getLogistics(String OrderID) | 获取订单的状态信息 | |

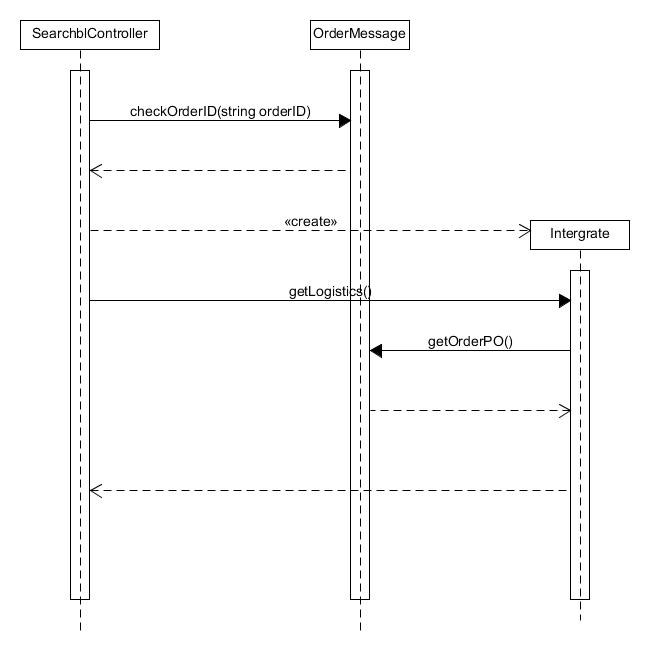
**OrderMessage的接口规范**

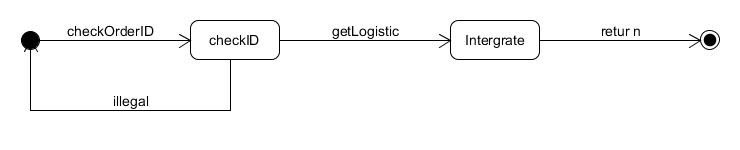
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderMessage.  checkOrderID | 语法 | public boolean checkOrderID(String OrderID) |
| 前置条件 | orderID为订单号 |
| 后置条件 | 若订单号存在，返回true，否则返回false |
| OrderMessage.  getOrderPO | 语法 | public OrderPO getOrderPO() |
| 前置条件 | 订单号存在 |
| 后置条件 | 返回订单信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDataService.  getOrderPO | 根据ID获得订单信息 | |

**Intergrate的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Intergrate.  getLogistics | 语法 | public boolean checkOrderID(String OrderID) |
| 前置条件 | orderID为订单号 |
| 后置条件 | 若订单号存在，返回true，否则返回false |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| TransportDataService.  getTransportPO(String OrderID) | 获取中转/装车单的信息 | |
| ReceiveDataService.  getReceivePO(String OrderID) | 获取到达单的信息 | |
| DeliverDataService.  getDeliverPO(String OrderID) | 获得派件单信息 | |
| StoreDataService.  getStoreInPO | 获取入库单信息 | |
| StoreDataService.  getStoreOutPO | 获取出库单信息 | |
|  |  | |
| OrderMessage.  getOrderPO | 获取订单信息 | |

1. 业务逻辑层的动态模型





1. 业务逻辑层的设计原理

采用集中式控制风格，由controller决定ordermessage和integrate的存亡。