

# | Dart语言（4）实验一：基于RooCode四角色协同开发“外贸发货看板”命令行应用

以下是“外贸发货看板”命令程序需求，专门针对**Dart语言开发**，并结合RooCode四个角色（战略协调者、技术助理、技术领导者、高级软件工程师）的实验讲义。

## | 一、需求说明

### | 1. 功能需求概述

- **中国贸易商**：登录系统，添加新运单，更新已有运单信息。
- **下游客户**：登录系统，输入客户代码，查看所有相关运单信息。

### | 2. 主要数据结构

#### | 下游客户（Customer）

字段	类型	示例	说明
客户代码	String	NTS-2304	唯一标识客户
客户姓名	String	Ruixin	
客户联系方式	String	+966 551234567	
客户国家	String	SA	
客户语言	String	ar	
客户时区	int	+3	日期偏移量（小时）

#### | 运单（Shipment）

字段	类型	示例	说明
发货港口时区	int	+8	日期偏移量（小时）
单号	String	NTS-2304_Elec_10_20250506123000	由客户代码 <b>货物简称</b> 包裹数_时间戳组成
客户代码	String	NTS-2304	关联客户
货物运输方式	String	Sea / Air	海运或空运
货物预计运输时间	int	30	天数
运单状态	String	未发货/已发货/已到达/已提货	
运单建立时间	DateTime	2025-05-06 12:30:00	自动生成
运单发出日期	Date	2025-05-07	
运单到达时间	Date	2025-06-06	
运单提货时间	Date	2025-06-07	

字段	类型	示例	说明
货物简称	String	Elec	不超过16字符
货物明细	String	电子元件，型号ABC123	
包裹数量	int	10	

### 3. 用例描述

- 用例1**：贸易商登录后创建新运单或更新已有运单。
- 用例2**：客户登录后输入客户代码，查询所有相关运单信息。

## 二、开发步骤与角色分工

### Step 1：需求理解与任务拆解（由**战略协调者**完成）

- 任务**：理解需求，拆解为“设计数据结构”、“设计命令行交互”、“实现数据存储”、“实现用户身份区分”等子任务。
- 提示词注意事项**：描述清晰，强调拆解的逻辑性和任务间依赖。
- 学生练习**：请编写提示词，让战略协调者帮你拆解“外贸发货看板”Dart命令行程序的开发任务。

### Step 2：制定详细开发计划（由**技术领导者**完成）

- 任务**：根据拆解任务，制定详细开发计划，包括模块划分、接口设计、数据存储方案（内存或文件）、命令行交互设计等。
- 提示词注意事项**：计划条理清晰，涵盖每个模块的输入输出和核心逻辑。
- 学生练习**：请编写提示词，让技术领导者帮你制定“外贸发货看板”第一阶段的开发计划。

### Step 3：技术问题答疑（由**技术助理**支持）

- 任务**：解答开发过程中的技术问题，如“Dart命令行参数解析推荐库”、“如何格式化时间戳”、“如何读写本地文件”等。
- 提示词注意事项**：问题具体，便于技术助理给出精准建议。
- 学生练习**：请编写提示词，向技术助理咨询一个你关心的Dart命令行开发技术问题。

### Step 4：编码实现（由**高级软件工程师**负责）

- 任务**：根据计划，编写代码实现各功能模块，包括数据结构定义、命令行交互、数据存储和查询功能。
- 提示词注意事项**：代码模块化，注重规范和可维护性。

- **学生练习**：请编写提示词，让软件工程师帮你实现“添加运单”功能的Dart代码。

## Step 5：整体协调与进度跟踪（由**战略协调者**持续管理）

- **任务**：协调各角色工作进度，处理简单问题，汇总成果，准备演示。
- **提示词注意事项**：强调任务状态和优先级管理。
- **学生练习**：请编写提示词，让战略协调者帮你汇总当前开发进度。

## 三、Dart命令行界面使用示例

假设程序文件名为 `shipment_board.dart`，以下是示例交互：

```
# 贸易商登录
$ dart run shipment_board.dart login --role trader
请输入用户名: trader1
登录成功!

# 添加新运单
$ dart run shipment_board.dart add_shipment
请输入客户代码: NTS-2304
请输入货物简称: Elec
请输入包裹数量: 10
请输入运输方式 (Sea/Air): Sea
请输入预计运输天数: 30
请输入货物明细: 电子元件, 型号ABC123
运单创建成功, 单号: NTS-2304_Elec_10_20250506123000

# 更新运单状态
$ dart run shipment_board.dart update_shipment --tracking NTS-2304_Elec_10_20250506123000
请输入新的运单状态 (未发货/已发货/已到达/已提货): 已发货
运单状态更新成功!

# 客户登录
$ dart run shipment_board.dart login --role customer
请输入客户代码: NTS-2304
登录成功!

# 查看所有运单
$ dart run shipment_board.dart list_shipments
运单列表:
1. 单号: NTS-2304_Elec_10_20250506123000, 状态: 已发货, 预计到达: 2025-06-06
...
```

## 四、补充说明

- Dart命令执行程序运行方式：
    - 创建项目：`dart create -t console-simple shipment_board`
    - 运行程序：`dart run bin/shipment_board.dart`
  - 相关Dart库推荐：
    - `dart:io` 用于读取输入、写入输出、文件操作
    - `args` 包用于解析命令行参数
    - `intl` 包用于时间格式化（可选）
- 

## 五、总结

- 本讲义通过“外贸发货看板”案例，结合RooCode四角色分工，演示了如何用Dart语言开发命令行应用。
  - 角色分工明确，流程清晰，便于学生理解协作开发。
  - 结合Dart语言特性，设计合理的数据结构和交互方式，保证程序可扩展和易维护。
- 

参考文献： [1] Write command-line apps - Dart <https://dart.dev/tutorials/server/cmdline> [2] Get started: Command-line and server apps - Dart <https://dart.dev/tutorials/server/get-started> [3] dart: The Dart command-line tool <https://dart.dev/tools/dart-tool> [4] Dart Beginners Course - Tutorial #05: Build a Command Line App <https://www.youtube.com/watch?v=XoZ01mY-cUg> [5] Dart CLI Foundations: CLI Tools to Streamline Your Workflow <https://pieces.app/blog/dart-cli-foundations> [6] Implementing and Using Data Structures in Dart - Finotes <https://www.blog.fnotes.com/post/implementing-and-using-data-structures-in-dart> [7] Learn Dart - Dart CLI Online Course | Fireship.io <https://fireship.io/courses/dart/basics-cli/> [8] Dart data structures - DhiWise <https://www.dhiwise.com/post/dart-data-structures-decoded-empowering-flutter-development> [9] Generate a Dart CLI with Very Good CLI <https://verygood.ventures/blog/generate-command-line-application-cli> [10] Dart and Flutter data structures: A comprehensive guide <https://blog.logrocket.com/dart-flutter-data-structures-comprehensive-guide/> [11] Tutorials: Command-line apps and servers - Dart <https://dart.dev/tutorials/server> [12] GitHub - kaljitism/Data-Structures-and-Algorithms-Dart <https://github.com/kaljitism/Data-Structures-and-Algorithms-Dart> [13] How to Create Command Line in Dart - YouTube <https://www.youtube.com/watch?v=FuK1VhrCPzg> [14] TheAlgorithms/Dart: All Algorithms implemented in Dart - GitHub <https://github.com/TheAlgorithms/Dart> [15] Data Structures & Algorithms in Dart, Chapter 4: Stacks | Kodeco <https://www.kodeco.com/books/data-structures-algorithms-in-dart/v1.0/chapters/4-stacks> [16] Queue (and Stack) [Data Structures, Dart, Flutter] - YouTube <https://www.youtube.com/watch?v=1dM0CjiBVGQ>