智能充电桩调度计费系统

面向对象设计

（文档封面及目录格式与以前作业相同）

# 软件体系结构设计 (10分)

//请设计系统的软件体系结构。如下例所示：(1)画出体系结构图；(2)说明体系结构各个部分的组成和功能。

例：

本系统采用“分层”的体系结构，其软件分层结构如下所示：



用户界面层：该层完成XXX功能。

控制器层：该层完成XXX功能。

……

# 用例实现方案设计—创建动态结构(80分)

//系统操作是由外部参与者通过发送系统消息产生的。请根据不同用例场景中包含的系统操作, 如表1所示，为表中每一个系统操作创建“协作图/时序图(二者选其一)”。

表1 系统操作表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例** | **系统操作**  **注:每个系统操作包含的参数仅作参考，可根据小组具体情况进行修改，建议不少于以下给出的参数。** | | **返回信息**  **注:每个返回信息包含的参数仅作参考，可根据小组具体情况进行修改，建议不少于以下给出的参数。** |
| 充电申请 | 1、提交充电申请 | E\_chargingRequest(car\_Id, Request\_Amount, Request\_Mode) | Return(car\_position, car\_state, queue\_Num, request\_time) |
| 2、修改充电量 | Modify\_Amount(car\_Id, Amount) | Return(0/1) |
| 3、修改充电模式 | Modify\_Mode(car\_Id, Mode) | Return(0/1) |
| 4、查看队列状态 | Query\_Car\_State(car\_id) | Return(car\_Number\_before\_position, car\_state, queue\_Num, request\_time) |
| 5、开始充电 | Start\_Charging(car\_id, ChargePileNum) | Return(0/1) |
| 6、查看充电状态 | Query\_Charging\_State(car\_id) | Return(详单信息，具体参数详见需求说明） |
| 7、结束充电 | End\_Charging(car\_id, ChargingPileNum) | Return(0/1) |
| 查看账单 | 1、查看账单申请 | Request\_Bill(carId, date) | Return(carId, date, Bill\_Id, ChargePileNum, ChargeAmount, ChargeDuration, StartTime, EndTime, TotalChargeFee, TotalServiceFee, TotalFee) |
| 查看详单 | 1、查看详单申请 | Request\_DetailedList(Bill\_Id) | Return(carId, date, Bill\_Id, ChargePileNum, ChargeAmount, ChargeDuration, StartTime, EndTime, ChargeFee, ServiceFee, subtotalFee) |
| 运行充电桩 | 1、启动充电桩 | powerOn(pile\_Id) | Return(0/1) |
| 2、设置参数 | setParameters(计费规则，三个时段的电价数据等） | Return(0/1) |
| 3、运行充电桩 | Start\_ChargingPile(pile\_Id) | Return(0/1) |
| 4、关闭充电桩 | powerOff(pile\_Id) | Return(0/1) |
| 查看充电桩状态 | 1、查看充电桩状态。要求定时刷新所有充电桩的状态并在客户端显示 | Query\_PileState(pile\_Id) | Return(workingState, TotalChargeNum, TotalChargeTime, TotalCapacity) |
| 查看队列状态 | 1、查看队列状态 | Query\_QueueState(queuelist) | Return(car\_Id, car\_Capacity, Request\_Amount, waitTime) |

## 用例1(参照表1给出具体用例名称，如“注册”)

* + 1. logon(role:string,Id:string,pwd:string) （参照表1给出系统操作名称，带参数）

实现该操作的协作图或时序图如下所示：



* + 1. XXX (系统操作名称，带参数)

实现该操作的协作图或时序图如下所示：

… …

## 用例2(具体用例名称)

// 内部格式同2.1

… …

## 2.n系统内部场景

// 系统内部操作通常由系统内部事件触发，不需要通过外部参与者。请根据系统内部场景中包含的操作，如表2所示，为表中每一个系统内部操作构造“协作图/时序图(二者选其一)”。

表2系统内部操作表

|  |  |  |
| --- | --- | --- |
| **场景** | **内部操作** | **操作说明** |
| 充电桩故障调度 | 1、优先级调度 | 单一充电桩故障且正好该充电桩有车排队的情况下，开始重新为故障队列中的车辆进行调度。1和2两种调度策略实现一种即可，具体调度策略参见《系统详细需求》 |
| 2、时间顺序调度 |
| 3、故障恢复 | 当充电桩故障恢复，若其它同类型充电桩中尚有车辆排队，则重新调整调度，具体调度策略参见《系统详细需求》 |
| 扩展调度请求 | 1、基于“单次调度总充电时长最短”的调度 | 具体调度策略参见《系统详细需求》。注：本部分为选做，可加分。 |
| 2、基于“b) 批量调度总充电时长最短”的调度 |

2.n.1优先级调度（参照表2给出内部操作名称，可带参数）

实现该操作的协作图或时序图如下所示：

… …

# 用例实现方案设计—创建静态结构(10分)

// 结合第2节中的交互图和《软件需求规格说明书(面向对象分析)》中的领域模型，完成设计类图。设计类图需要根据第1节描述的软件体系结构，分别列出体系结构中每个部分所包含的设计类，以及类之间的调用关系。例如对于分层体系结构，通常列出用户界面层、控制器层、业务领域层、持久化层，如果存在系统层的类，也应当列出。

// 每个设计类应当包含名称、属性、方法，具体格式可参考下例：

## 用户界面层设计类图

// 用户界面层设计类可以不必列出界面类的属性，但必要的系统操作方法需要列出；同时列出用户界面类与其它层次类之间调用关系。



**业务/领域层**

**控制器/处理层**

**用户界面层**

## 控制器/处理层设计类图

// 控制器类只需列出关键属性和关键方法，同时列出与其它层次类之间调用关系。

## 业务/领域层设计类图

// 业务领域类应当列出所有可能的属性和方法（不包含get/set方法和构造函数等），同时列出与其它层次类之间调用关系。

## 持久化层设计类图

// 持久化层类只需列出关键属性和关键方法，同时列出与其它层次类之间调用关系。

## 系统层设计类图

// 如果有，请列出