哈尔滨医科大学附属第二医院病房内泌五患者住院诊疗业务

目录

[工作目标 3](#_Toc471225592)

[业务背景 3](#_Toc471225593)

[词汇说明 3](#_Toc471225594)

[业务用例模型 4](#_Toc471225595)

[业务分析模型 4](#_Toc471225596)

[业务用例实现 4](#_Toc471225597)

[主流程： 4](#_Toc471225598)

[指令执行-取药流程： 6](#_Toc471225599)

[指令执行-配液流程： 7](#_Toc471225600)

[指令执行-输液流程： 8](#_Toc471225601)

[指令执行-检查流程： 8](#_Toc471225602)

[业务系统 9](#_Toc471225603)

[业务系统结构图 9](#_Toc471225604)

[业务实体 10](#_Toc471225605)

[场景设计 16](#_Toc471225606)

[描述责任医生下一个药品的临时医嘱，用法为口服的执行场景： 16](#_Toc471225607)

[描述责任医生下一个检查医嘱的执行场景： 16](#_Toc471225608)

[描述责任医生作废一个已收费、未完成，并由它科执行的临时医嘱场景： 17](#_Toc471225609)

[描述责任医生作废一个已预约、并由它科执行的检查类临时医嘱场景： 17](#_Toc471225610)

[描述责任医生停止一个长期医嘱场景： 17](#_Toc471225611)

[描述责任医生下一个手术医嘱场景： 17](#_Toc471225612)

[描述责任医生下一个出院医嘱场景： 18](#_Toc471225613)

[系统用例模型 19](#_Toc471225614)

[参与者 19](#_Toc471225615)

[用例 20](#_Toc471225616)

[分析模型 21](#_Toc471225617)

[用例实现 21](#_Toc471225618)

[送诊用例实现 21](#_Toc471225619)

[预存住院费用例实现 21](#_Toc471225620)

[接诊用例实现 22](#_Toc471225621)

[下医嘱用例实现 22](#_Toc471225622)

[核对医嘱用例实现 23](#_Toc471225623)

[分解长期医嘱用例实现 24](#_Toc471225624)

[发送执行条目用例实现 24](#_Toc471225625)

[定时启动执行条目用例实现 25](#_Toc471225626)

[确认完成执行条目用例实现 25](#_Toc471225627)

[取消医嘱用例实现 25](#_Toc471225628)

[同意退费用例实现 26](#_Toc471225629)

[摆药用例实现 27](#_Toc471225630)

[取消摆药用例实现 27](#_Toc471225631)

[出院登记用例实现 28](#_Toc471225632)

[出院结算用例实现 28](#_Toc471225633)

[分析元素 29](#_Toc471225634)

[分析包结构图 29](#_Toc471225635)

[实体类关系图 30](#_Toc471225636)

[设计模型 31](#_Toc471225637)

[主结构图 31](#_Toc471225638)

[实体结构图 32](#_Toc471225639)

[服务依赖关系图 33](#_Toc471225640)

[用例实现 33](#_Toc471225641)

[送诊用例实现 33](#_Toc471225642)

[预存住院费用用例实现 34](#_Toc471225643)

[接诊用例实现 34](#_Toc471225644)

[下医嘱用例实现 35](#_Toc471225645)

[核对医嘱用例实现 36](#_Toc471225646)

[发送执行条目用例实现 37](#_Toc471225647)

[定时启动执行条目用例实现 37](#_Toc471225648)

[确认完成执行条目用例实现 37](#_Toc471225649)

[取消医嘱用例实现 39](#_Toc471225650)

[同意退费用例实现 40](#_Toc471225651)

[分解长期医嘱用例实现 40](#_Toc471225652)

[摆药用例实现 41](#_Toc471225653)

[取消摆药用例实现 41](#_Toc471225654)

[出院登记用例实现 41](#_Toc471225655)

[出院结算用例实现 42](#_Toc471225656)

[可变性设计 43](#_Toc471225657)

[医嘱条目的分解取决于医嘱类型 43](#_Toc471225658)

[医嘱执行条目的完成和作废可以根据不同的条目类型嵌入业务逻辑 44](#_Toc471225659)

[医嘱执行条目到收费条目的映射 45](#_Toc471225660)

[测试设计 45](#_Toc471225661)

[入院阶段 46](#_Toc471225662)

[治疗阶段 47](#_Toc471225663)

[出院阶段 48](#_Toc471225664)

[实现模型 50](#_Toc471225665)

[选型和决策 50](#_Toc471225666)

[附录 50](#_Toc471225667)

[建模过程中偏离哈医大二院实际情况的地方 50](#_Toc471225668)

[有关方法的说明 51](#_Toc471225669)

[疑问 51](#_Toc471225670)

# 工作目标

采用新的方式做一个概念清晰、结构简单、能够演进的支撑医院业务的内核。期望这个内核是延续型的，并能够带动以此内核成长起的软件系统的高质量，并对人员分工产生实质性的影响。

高质量的内涵：

1. 通过简化业务和赋予软件系统更多的职责提高医生和护士的工作效率。
2. 软件系统可以采用较低的成本跟上人思考业务的变化。

关于新的方式的说明：

采用了经典RUP相关的工程方法。该方法可以在不同的层面上实现：

* **分离关注**，可以独立思考又实现相互联系
* **可视化**的表达对业务、模型、软件结构的理解和设计
* 能够从方法上实现对理解和设计进行**验证**

关于简化业务和赋予软件系统更多的职责的思路有：

* 医嘱开立、核对人工干预，分解、发送（基于规则）、收费自动完成，将更多的业务（如：出院登记、结算等）塑造为医嘱执行，并依靠软件系统完成更多的职责
* 将部分病历书写视为医嘱执行，通过简单信息的录入自动生成符合病案室归档的格式
* 将与医嘱执行有关的角色塑造为指令执行者，采用统一的待办项列表的形式办理业务，在正常流程下减少对业务和软件复杂度的感知

# 业务背景

内泌五以糖尿病患者为主，平均住院人数在50-70人间，每天入院患者约5-10人。患者一般住院时间为10天。该科室经常向眼科科室发起会诊请求。

# 词汇说明

医嘱分解：将医生开立的医嘱条目按着执行步骤进行的分解，分解后的执行条目由责任人（指令执行者）执行。这里的医嘱分解不仅包含了长期医嘱按着频次进行的分解内涵，也包括了临嘱的分解（如：检查医嘱分解为预约检查和检查两个执行条目）。医嘱的分解与类型有关，与不同的医院有关、甚至与具体的科室有关。

# 业务用例模型



住院业务（目标业务组织）包含了住院处、住院科室、药房、医技、质控、病案室等为住院业务服务的主要部门。

患者以门诊患者身份住院，以住院患者或死亡患者身份出院。

问题：病案室是否从住院业务中分离出来成为业务外内容需要交流。

# 业务分析模型

## 业务用例实现

### 主流程：



黄色步骤是与医嘱执行有关的业务步骤，一条医嘱的执行伴随着医嘱指令的分解、具体（多条）执行指令的执行，在指令执行过程中伴随着记录成本、扣费，以及更新医嘱的执行状态等。

其中指令执行为抽象的业务步骤，指令执行者为抽象岗位：



支撑医嘱指令执行的科室、岗位、软件系统等都需要按着对应的执行指令的要求，更新医嘱执行状态、记录成本、扣费等。

责任医生是医嘱的创建者，同时也是医嘱执行的执行者（处置、写出院小结等）

以下将以取药、配液和检查来说明具体的指令执行。

### 指令执行-取药流程：

以下是以取药流程为例描述该子流程：



### 指令执行-配液流程：

配液工作是批量操作，每天由配液中心统一配液。



### 指令执行-输液流程：



### 指令执行-检查流程：



护士分成了接诊护士和责任护士。



## 业务系统

### 业务系统结构图

在以内泌五作为主要分析科室的分析过程中，识别了与内泌五入院治病业务有关的业务科室。



* 内泌五依赖住院处为患者登记基本信息、分配科室，确定责任医生等。
* 依赖收费处收费、药房取药，配液中心配液，医技科室进行检查检验、手术室手术、其他住院科室进行会诊等。
* 当内泌五的责任医生下医嘱时，找不到对应的医嘱术语类型时需要联系医务科进行维护新的医嘱术语类型。
* 收费处的收费受物价科的规范。
* 当一次诊疗完成后，质控将对内泌五的诊疗过程进行评价。

### 业务实体

在诊疗过程中，涉及到与患者一次住院医嘱有关的业务实体有医嘱单、执行单，以及收费单。其包含的医嘱条目、执行条目和费用条目的关系都是一对多关系。**医嘱条目通过执行条目与费用条目关联。**



* 医嘱条目是责任医生根据患者病情下的医学指令，汇聚成的医嘱单表现了计划治疗过程
* 执行条目是根据医嘱条目分解得到的多个执行步骤，汇聚成的执行单体现了实际治疗过程
* 收费条目是在医嘱执行的过程中针对一条执行条目而形成的收费（也有非医嘱对应的收费条目），汇聚成的收费单表现了患者的费用花销
* 成本条目是在医嘱执行的过程中针对一条执行条目而形成的成本，当患者有义务承担这个成本的时候，成本条目将转化为收费条目，在因院方原因作废医嘱时，将通过新增负的收费条目来抵消收费，但如果实际发生了成本将不抵消成本条目
* 当医院为事前收费时，收费条目早于成本条目创建，当事后收费时，收费条目将晚于成本条目创建
* 医嘱类型是有层次关系的，如：嘱托长嘱属于长期医嘱



* 医嘱条目的名称往往是关联的医嘱类型的名称，如一条药品医嘱
* 医嘱条目可以分为单条医嘱和组合医嘱



* 一条医嘱条目可以对应多种医嘱类型（如：长期/临时，药品/非药品，检验/检查等）
* 不同的医嘱类型对应的医嘱条目间存在前后约束关系（如：皮试用/需皮试）
* 没有对应执行条目的医嘱条目为处于计划状态，可以直接删除
* 没有对应费用条目的执行条目为未执行的条目、不收费的条目或还没有收费的条目，可以直接作废或停止
* 一条执行中的医嘱条目对应一条正处于执行中的执行条目（执行条目的执行是串行的）
* 执行条目的类型与关联的医嘱条目的类型有关
* 部分执行条目会关联申请单和执行结果
* 费用条目有费用类型，它与执行条目的类型有关



* 费用条目当其不关联执行条目时，为非医嘱类的费用条目（如：床位费）
* 一次作废医嘱业务将在收费单上至少出现两条收费条目（金额一正一负）
* 每个抽象的执行条目都会根据执行的过程改变自己的状态，当指令执行者为本科室时，状态图：



* 医嘱作废和停止都由创建医嘱的责任医生来做。
* 一条医嘱条目对应的多条执行条目中不能有两条都处于执行中。
* 当指令执行者为他科室时，状态图：



* 该状态图增加了【已发送】状态，用以人为确定是否让他科执行。当医嘱条目被分解为多条执行条目时，只在首个执行条目有发送动作。
* 根据执行条目与费用条目的关联情况，可以有以下独立状态，是否发生成本（与成本条目建立关联），是否收费（与收费条目建立关联）

 

* 当停止和作废医嘱不依赖于费用处理时（后处理费用），将增加待退费状态
* 在本科执行的执行条目的退费由本科室来做，由他科室执行的执行条目的退费由他科室来做。
* 医嘱条目状态图：



* 除了责任医生手动停止指定长期医嘱条目外，创建出院医嘱或死亡医嘱以及转科医嘱时会停止该患者所有执行中的长期医嘱条目
* 停止/作废医嘱后，对应的执行条目处于（已收费/执行中、待执行）状态组合时需要自动退费
* 对于因手术等原因已停止的长期医嘱可以通过恢复操作继续执行（需创建新的执行周期）

在诊疗过程中，涉及到与患者一次住院病历有关的业务实体有病历夹、病历、病历项目等。



* 检查结果和检验结果除了是执行条目关联的执行结果外，也是一种住院病历类型。
* 住院病历有状态，当被签名后将不能在修改。

### 场景设计

#### 描述责任医生下一个药品的临时医嘱，用法为口服的执行场景：

1. 责任医生创建一个临时的药品医嘱。
2. （系统）根据药品医嘱对应的药品进行预扣操作。
3. 责任护士审核医嘱条目，（系统）创建医嘱条目对应的两条执行条目，一条为摆药执行条目，执行者为发药岗位。一条为取药执行条目，执行者为责任护士。
4. 责任护士发送摆药执行条目。
5. 发药岗位确定该患者是否欠费，不欠费摆药并通知扣费岗位扣费。如欠费，（系统）设置对应的医嘱条目执行状态描述为【欠费，未摆药】。
6. 扣费岗位（由系统承担）记录成本、扣费，创建成本条目和收费条目，设置执行条目的收费状态变为已收费。（系统）将摆药执行条目记录为已完成，并将取药执行条目设置为执行中。修改医嘱条目的执行状态描述为【已摆药，待取药】
7. 责任护士领药，发给患者。
8. 责任护士确认完成取药执行条目，设置取药执行条目的主状态变为已完成。由于取药执行条目是对应的医嘱条目的最后一条执行条目，医嘱条目也变为已完成，医嘱条目的执行状态描述为已完成。

#### 描述责任医生下一个检查医嘱的执行场景：

1. 责任医生创建一个检查医嘱（说明检查项目），并设置执行科室。
2. 责任护士审核医嘱条目，（系统）创建医嘱条目对应的两条执行条目，一条为预约检查执行条目，执行者为执行科室预约岗位。一条为检查执行条目，执行者为执行科室检查岗位。
3. 责任护士发送预约检查执行条目（含申请单）。
4. 执行科室预约岗位进行预约检查，并反馈预约检查结果，（系统）将该执行条目置为已完成。
5. 责任护士将预约检查结果反馈给住院患者。
6. 住院患者在检查的启动时间前到执行科室等待检查。
7. 执行科室检查岗位确定该患者是否欠费，不欠费进行检查，生成检查结果，并通知扣费岗位扣费。如欠费，（系统）设置对应的医嘱条目执行状态描述为【欠费，待检查】。
8. 扣费岗位（由系统承担）记录成本、扣费，创建成本条目和收费条目，设置执行条目的收费状态变为已收费。（系统）将检查执行条目记录为已完成。修改医嘱条目的执行状态描述为【已完成】
9. （系统）将检查结果发送给责任医生。

#### 描述责任医生作废一个已收费、未完成，并由它科执行的临时医嘱场景：

1. 责任医生作废对应的医嘱条目，修改医嘱条目和对应的执行条目的主状态为已作废，并修改执行条目的收费状态为待退费。
2. （系统）根据作废的原因确定是否在医嘱单上进行删除。
3. 由指令执行者对相应的执行条目（可能多条）进行退费操作。
4. 扣费岗位（由系统承担）退费，创建收费条目（金额为负），并修改响应的执行条目的收费状态为已退费。
5. 由指令执行者判断成本条目是否可以删除，删除或保留对应的成本条目。

#### 描述责任医生作废一个已预约、并由它科执行的检查类临时医嘱场景：

1. 责任医生作废对应的医嘱条目，修改医嘱条目和对应的执行条目的主状态为已作废。
2. （系统）根据作废的原因确定是否在医嘱单上进行删除。
3. （系统）删除执行条目对应的预约记录。
4. （可选）指令执行者根据新的预约情况将之后产生的预约申请补在该时间段内。

#### 描述责任医生停止一个长期医嘱场景：

1. 责任医生停止对应的医嘱条目，修改医嘱条目为已停止。
2. （系统）检索医嘱条目对应的执行条目中处于执行中已收费的执行条目，并将这些执行条目的收费状态置为待退费状态。
3. （系统）将执行条目置为已停止状态。
4. 由指令执行者对相应的执行条目（可能多条）进行退费操作。
5. 扣费岗位（由系统承担）退费，创建收费条目（金额为负），并修改响应的执行条目的收费状态为已退费。
6. 由指令执行者判断成本条目是否可以删除，删除或保留对应的成本条目。

#### 描述责任医生下一个手术医嘱场景：

1. 责任医生创建一个手术医嘱（确定手术医生、拟手术室）。
2. （系统）根据手术医嘱的情况发给相应的角色进行审批。
3. 手术医嘱审批同意后，（系统）创建手术医嘱条目和六条执行条目，分别是排班执行条目、术前访视执行条目、接收手术患者执行条目、手术执行条目、返回病房或ICU执行条目、编写病历执行条目。
4. 责任护士创建手术申请、发送排班执行条目到手术室进行排班。
5. 手术室排班岗位根据手术申请进行排班，确定手术间、手术时间、手术护士和麻醉师，并反馈。
6. （系统）将排班执行条目设置为已完成，更新患者一次住院的状态（准备手术）。修改术前访视执行条目为执行中。
7. 麻醉师和手术医生分别进行术前访视，确定手术方案，并将前访视执行条目置为已完成。
8. 责任医生停止手术患者在手术期间的长期医嘱。
9. 手术护士准备手术环境、接收手术患者，完成接收手术患者执行条目，记录手术开始时间。
10. （系统）更新患者一次住院的状态（手术中）在，将手术执行条目置为执行中。
11. 麻醉师进行麻醉。
12. 手术医生进行手术。
13. 手术护士协助手术医生进行手术。
14. 麻醉师观察手术患者生命体征，并记录手术过程数据。
15. （可选）手术医生或麻醉师创建新的医嘱获取手术期间需要但未准备的药品或用品。（系统）自动将其关联到手术医嘱上。
16. 手术结束，麻醉师唤醒手术患者，将手术执行条目设置为已完成。
17. 根据手术结果，手术医生判断手术患者回病房或进ICU等，并将返回病房或ICU执行条目置为已完成。
18. （系统）更新患者一次住院的状态（病房/ICU）。
19. 手术护士和麻醉师整理手术数据，在手术执行条目上登记材料费和其他费用【后收费】。
20. 扣费岗位记录成本，创建（多条）成本条目和收费条目，并进行扣费。
21. 手术护士整理剩余术间病历，归档到患者的病历夹中。
22. 手术医生编写手术记录，将编写病历执行条目设置为已完成。
23. 责任医生根据手术结果恢复患者的长期医嘱。

#### 描述责任医生下一个出院医嘱场景：

1. 责任医生创建一个临时的出院医嘱，确定执行时间。
2. 责任护士审核医嘱条目，（系统）创建医嘱条目对应的两条执行条目，一条为出院登记执行条目，执行者为责任护士岗位。一条为出院结算执行条目，执行者为扣费岗位。
3. 责任护士发送出院结算执行条目到扣费岗位。
4. （系统）根据执行时间修改出院登记执行条目的主状态为执行中。
5. 责任护士检查可以进行出院登记的患者状态是否满足（不满足通知相关岗位进行处理），做出院登记，（系统）设置对应的执行条目为已完成，设置对应的医嘱条目执行状态描述为【已出院登记，待结算】，将出院结算执行条目主状态设为执行中。
6. 扣费岗位对患者进行出院结算（补费或退费），（系统）置对应的执行条目为已完成，设置对应的医嘱条目执行状态描述为【已结算】，状态为【已完成】，修改患者一次就诊的状态为【已出院】。

# 系统用例模型

## 参与者



## 用例



* 指令执行者即有可能是现实中的岗位，也可能是其他软件系统。
* 发送执行条目只发生在有他科执行的医嘱中。
* 核对医嘱伴随着医嘱条目的分解，并通知第一个指令执行者。
* 下医嘱时有可能伴随着创建申请单。
* 确认完成执行条目会通知下一条同组执行条目对应的指令执行者。
* 确认完成执行条目伴随着回传检查或检验结果。
* 确认完成执行条目伴随着扣费。
* 取消医嘱包含了作废和停止两项动作。
* 同意退费是当取消医嘱后的一个操作。

# 分析模型

## 用例实现

### 送诊用例实现



* 在该场景下是否存在失败的情况还需探讨。

### 预存住院费用例实现

* 第一次预存住院费



* 收费单用于保存患者一次住院的费用账户额度

### 接诊用例实现



* 设置床位后，调用费用控制器创建计费项目。费用控制器定期根据计费项目创建收费条目，自动更新患者一次就诊关联的收费单余额。

### 下医嘱用例实现

* 下一个药品临时医嘱



* 下一个检查临时医嘱



### 核对医嘱用例实现



* 长期医嘱只分解近2天的执行条目，后续定时自动分解。

### 分解长期医嘱用例实现



### 发送执行条目用例实现



### 定时启动执行条目用例实现



### 确认完成执行条目用例实现



### 取消医嘱用例实现

* 作废一个已收费的临时药品医嘱



### 同意退费用例实现



### 摆药用例实现



### 取消摆药用例实现



### 出院登记用例实现



### 出院结算用例实现



## 分析元素

### 分析包结构图



* 尽量保持领域层分析包的独立性
* 医嘱分析包包含了医嘱的创建和执行业务中涉及的领域层元素
* 医嘱执行分析包为涉及医嘱执行业务的应用层分析包提供统一入口

### 实体类关系图



* 医嘱条目上的【分解执行条目】职责还须进一步分析。
* 收费条目除了对应于执行条目外，属于一个计费项目（如床位费）。
* 【护理级别价格】和【商品价格】都是计费项目，在设计模型中可以只保留计费项目
* 通过患者一次就诊直接与计费项目关联来计非医嘱产生的费用。

# 设计模型

## 主结构图



* orderexecute为交互组件，但稳定且有通用性，引入到了domain中。

## 实体结构图



* 药房（Pharmacy）、住院科室（InPatientDept）都是部门（Dept）
* 在部门（Dept）上关联医嘱执行（OrderExecute）和收费记录（ChargeRecord）
* 住院科室（InPatientDept）上关联医生（Doctor）、护士（Nurse）、患者（Visit）
* 药房（Pharmacy）上关联药品类型（DrugType）
* 医生（Doctor）、护士（Nurse）、一般员工（Staff）都是用户（AbstractUser）
* 医嘱执行主要扩展类：医嘱条目类型（OrderType）、医嘱执行条目（OrderExecute）
* 药品医嘱类型（DrugOrderType）的分解依赖于医嘱类型、用法（DrugTypeOrderResolver）

## 服务依赖关系图



## 用例实现

### 送诊用例实现



### 预存住院费用用例实现



### 接诊用例实现

* 主流程：



* 初始化收费项目：



### 下医嘱用例实现

* 下一个药品临时医嘱：



### 核对医嘱用例实现

* 核对药品临时医嘱：



### 发送执行条目用例实现



### 定时启动执行条目用例实现



### 确认完成执行条目用例实现

* 修改状态



* 处理费用（扣费、记录成本）



### 取消医嘱用例实现

* 作废一个已收费的临时药品医嘱



### 同意退费用例实现



### 分解长期医嘱用例实现



### 摆药用例实现



### 取消摆药用例实现



### 出院登记用例实现



### 出院结算用例实现



## 可变性设计

### 医嘱条目的分解取决于医嘱类型





* 医嘱条目分解委托给医嘱条目类型实现，医嘱条目类型是一个抽象类，不同的医嘱条目类型可以有不同的分解方式
* 每一个具体的医嘱条目类型可以根据自己的情况设计自己的属性集合（如DrugOrderType中的DrugType），这些属性集合可以在创建医嘱执行条目时使用到
* 每一次分解都会创建医嘱执行条目组（OrderExecuteTeam），所包含的所有医嘱执行条目共享一个teamId，并且OrderExecuteTeam可以建立医嘱执行条目的前后关系

### 医嘱执行条目的完成和作废可以根据不同的条目类型嵌入业务逻辑



* 通过继承OrderExecute可以创建不同业务逻辑的医嘱执行条目，覆盖掉doFinish和doCancel方法编写不同的业务逻辑

### 医嘱执行条目到收费条目的映射



* 收费业务逻辑的触发通过Listenter实现
* 可以通过覆盖掉自己的医嘱执行条目的createChargeRecords方法实现与收费条目的映射

## 测试设计

2016-12-28 10:30 患者001到内泌五办理住院，【患者001状态为在病房】

2016-12-28 10:10 患者001到收费处存2000元预存住院费, 【系统产生+2000元的费用条目】，【患者001状态为待接诊】

2016-12-28 09:50 患者001到住院处办理住院，【患者001状态为待预存费用】

2016-12-28 11:15药房摆药岗位333完成摆药，【系统产生-60元的药品001的收费条目，并创建相应的成本条目】

2016-12-28 13:45 责任护士003退费，【系统产生+60元的药品001的退费条目，并收回成本】

2016-12-28 11:45责任医生002作废了药品001的临嘱

2016-12-28 11:30责任医生002完成当前全部执行条目，两条执行条目（二级护理和取药）【系统产生-8元的二级护理的收费条目】

2016-12-28 11:05责任护士003发送摆药执行条目到药房

2016-12-28 11:00 患者001责任护士003核对全部医嘱

2016-12-28 10:50 患者001责任医生002开药品001临嘱2盒

2016-12-28 10:50 患者001责任医生002开二级护理长嘱

### 入院阶段

2017-01-04 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-06 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-05 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-03 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-02 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-01 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2016-12-30 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2016-12-29 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

### 治疗阶段

2017-01-09 10:30 收费处岗位222对患者001完成了一条执行条目（出院结算），【系统产生-1896元的费用条目】，【患者001状态为已出院】

2017-01-09 09:30责任护士002完成了两条执行条目（出院登记和二级护理），【系统产生-8元的二级护理的收费条目】，【患者001状态为待出院结算】

2017-01-08 09:10 责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-07 10:30责任护士003完成二级护理执行条目，【系统产生-8元的二级护理的收费条目】

2017-01-07 10:10 责任医生002对患者001开出院医嘱，出院时间为2017-01-09

2017-01-07 10:30 责任护士核对了对患者001的出院医嘱

### 出院阶段

# 实现模型

## 选型和决策

1、采用eclipse\jdk8\maven\springboot构建；

2、将应用划分为platform、domain、application、listener、portal、mobile、rest、test八个工程；

3、包结构以com.neusoft.hospital开始；

4、view采用springboot推荐的thymeleaf或swing构建；

5、持久化采用springdata；

6、主键生成采用uuid；

7、核心对象关系采用manytoone和onetomany构建（采用lazyload策略）；

8、数据库采用mysql；

9、采用Repository模式，通过在IdEntity中调用平台中的ApplicationContextUtil获取Repo；

10、采用可捕获异常来定义业务异常，并建立异常树来分类业务异常；

11、通过实体关系获取实体将不再通过Repo来获取，只通过实体上的引用获取；

12、当通过定制条件获取实体时使用Repo；

13、除了创建实体外，其他操作尽量在DomainService后通过Repo.find实例化实体；

14、领域逻辑尽量附着在实体上，其次是DomainService；

15、非领域逻辑不允许被DomainService和Entity依赖；

16、领域事件由DomainService抛出；

17、通过@Async可实现异步监听；

# 附录

## 建模过程中偏离哈医大二院实际情况的地方

* 作废临时医嘱不依赖于是否退费。
* 对长期医嘱的已停止状态增加了恢复操作。
* 收费过程同时记录成本，当出现退费时可以保留成本的发生。
* 作废和停止医嘱不经过责任护士核对，直接有效。
* 发送仅针对他科执行的医嘱。
* 医嘱的分解自动进行。

## 有关方法的说明

* 业务用例模型中的流程描述是不含在业务分析模型中新创建的业务实体的，而在业务分析模型中的场景设计是结合了新创建的业务实体及其状态变迁的
* 业务用例模型、业务分析模型，以及用例模型、分析模型，属系统分析师工作，分析模型是系统分析师和软件架构师交接点
* 一个阶段的工作做到一个程度就要将进度推进到下一个阶段，然后通过反馈来修正上一阶段的工作

## 疑问

* 发送医嘱涉及临时医嘱吗？（涉及）
* 长期医嘱的发送包含了分解和发送，如果执行科室是本科室，是否只需要分解，不需要发送？（是）
* 护士对长期医嘱生成的执行条目维护是否全部都由医生去做？而不是下一个描述医嘱，由护士去做？（护士做）
* 退费操作应该与医生作废和停止医嘱有关，为什么护士需要直接进行退费操作呢？（处理例外，如：多开留置针）
* 摆药和检查结果的录入是在HIS系统中做的吗？（摆药是，检查结果不是）