**数据结构与算法设计实践**

**课程设计任务书**

东北大学软件学院

2019年1月

# 

一 课程基本信息 3

二 教学目的和基本要求 3

三 课程设计的内容和安排 3

四 课程设计的题目 4

五 考核的方法和成绩的评定 4

六 时间的安排 4

七 课程设计报告的格式 4

八 课程设计结果的提交 5

九 课程设计题目说明 5

十 评分标准 20

# 一 课程基本信息

* 1. 课程设计名称：数据结构与算法课程设计
  2. 课程设计编号：
  3. 课程设计学分：2学分
  4. 课程设计周数（学时）：2周（32学时）
  5. 课程设计授课单位：软件工程系
  6. 指导方式：集体辅导和个别辅导相结合
  7. 适用专业：软件工程，软日, 软英，软留
  8. 课程设计教材及主要参考资料：

[1] 严蔚敏、吴伟民 编著.《数据结构》（C语言版）.北京：清华大学出版社. 2018

[2] M. AllenWeiss. Data Structures and Algorithm Analysis in Java Third Edition. 机械工业出版社, 2015.4.2

# 二 教学目的和基本要求

1. 了解并掌握数据结构和算法设计的方法，具备初步的独立分析和设计能力
2. 初步掌握软件开发过程的问题分析分析、系统设计、程序编码和测试等基本方法和技能。
3. 提高综合运用所学的理论知识和方法，独立分析和解决问题的能力。
4. 训练用系统的观点和软件开发一般规范进行软件的开发，培养软件工作者所具备的科学工作方法和作风。

# 三 课程设计的内容和安排

1. 问题分析和任务定义：根据设计题目的要求，充分地分析和理解问题，明确问题要求做什么，限制的条件是什么。
2. 逻辑设计：对问题中涉及到的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序和各抽象数据类型，逻辑设计的结果应写出每个抽象数据类型的定义（包括数据结构的描述和每个基本操作的功能说明），各个主要模块的算法，并画出模块之间的调用关系图。
3. 物理设计：定义相应的存储结构并写出各函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。详细设计的结果是对数据结构和基本操作做出进一步的求精，写出数据结构存储结构的类型定义，写出函数形式的算法框架。
4. 程序编码：把详细设计的结果进一步求精为程序设计语言。同时加入一些注解和断言，使程序中逻辑概；念清晰。
5. 程序调试和测试：采用自底向上，分模块进行，即先调试低层函数。能够熟练掌握调试工具的各种功能，设计测试数据确定疑点，通过修改程序来证实它或绕过它。调试正确后，认真整理源程序和注释，形成格式和风格良好的源程序清单和结果。
6. 结果分析：程序运行结果包括正确的输入及其输出结果、含有错误的输入和输出结果、算法时间复杂度和空间复杂度分析。
7. 撰写课程设计报告：总结和提升上述过程和步骤，写出结构严谨、表述清楚、符合设计规范的报告。

实践工作量满足课上学时32学时 + 课外32学时，对于课上学时分配如下：

* 1. 4学时：需求理解。
  2. 6学时：数据结构设计；
  3. 8学时：算法设计；
  4. 14学时：系统实现及测试；

# 四 课程设计的题目

本次实践的应用系统是《东软智能云工厂》的部分功能模块，详见《需求规约-东软智能云工厂》，要求完成用户管理功能，设备管理功能，数据字典功能，订单管理，产品管理，可接订单，生产管理功能七部分。最后需要对所做系统书写实验报告一份，格式见附页

# 五 考核的方法和成绩的评定

参见数据结构评分标准.xlsx

# 六 时间的安排

共2周，根据老师要求完成系统设计和实现。

# 七 课程设计报告的格式

* 1. 问题的概述、分析和研究意义
  2. 数据结构的逻辑设计和物理存储设计
  3. 重要算法的设计、流程描述和代码描述
  4. 数据结构的时间、空间复杂性分析和重要算法的复杂度分析。
  5. 程序的最终实现结果（包括重点界面的抓取，能说明问题的重要实验结果数据的打印或其可视化结果）。
  6. 参考文献
  7. 附录：关键数据结构的定义及关键算法的源代码。

# 八 课程设计结果的提交

1. 课程设计报告（一份、A4纸打印，同时包括一份电子文档）
2. 完整的程序系统（电子方式提交），能够对输入产生相应的输出，同时尽量完成可视化演示。
3. 源程序文档（电子方式提交），源程序代码要求适当缩进编排、结构清晰、可读性好，对源程序的类说明（如采用面向对象方法设计）、函数说明、接口说明、关键变量说明等要进行注释。
4. 答辩报告（制作PPT报告，电子方式提交），要求重点突出，思路清晰，同时就此报告准备答辩。

所有电子方式提交的文件全部存放在一个目录中，并对其进行压缩，压缩后的文件按规定格式进行命名，命名格式为：**班级号＋学号＋姓名**。

# 九 课程设计题目说明

本课程设计主要是完成智能云制造平台功能，包括用户管理功能，设备管理功能，数据字典功能，订单管理，产品管理，可接订单，生产管理功能七部分。主要考查线性结构(数组，链表，队列)、树、查找结构以及相关算法的设计与实现。

**整体要求**

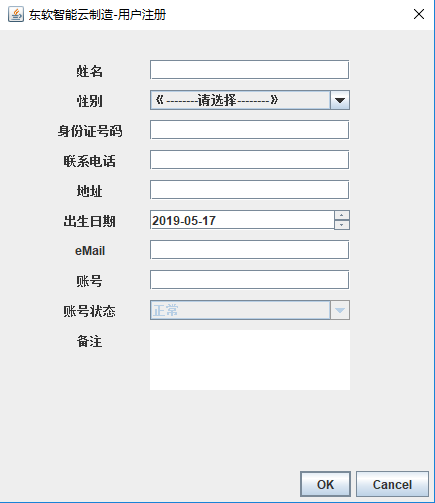
1. 所有数据以文件格式保存，文件存储在工程目录中。
2. 文件数据格式可以是JSON格式，也可以是以对象序列化的方式存储。
3. 系统分为用户管理功能，设备管理功能，数据字典功能，订单管理，产品管理，可接订单，生产管理功能七部分。

**系统功能**

**本部分介绍采用图形界面进行功能介绍，使用c语言实现可以将图形界面改为终端控制台输入。**

1. 用户登录





1.1完成登陆注册功能

1.2共有三种角色权限为：超级管理员，工厂管理和工人

权限说明：

a.超级管理员系统自带，不能被注册，可以发布订单，管理数据字典，查看所有系统用户等，界面如下



b.工厂管理通过用户注册，拥有用户管理，接收订单等功能，界面如下：



c.工人是通过工厂管理注册的角色，除用户管理功能外，其余功能跟工厂管理一致

2. 用户管理（超级管理员）

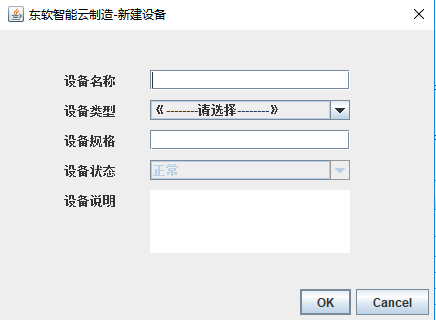


2.1可以查看所有系统用户

2.2基于姓名、用户编码、所属工厂等查询，可以单独用一个字段进行简单查询，也可以用多个字段进行精确查询

1. 设备管理（超级管理员）





* 1. 设备信息的增删改查
  2. 基于设备名称、设备编码、设备类型等查询，可以单独用一个字段进行简单查询，也可以用多个字段进行精确查询
  3. 新增设备租用为空闲，当设备未被租用时，租用信息为空

4. 数据字典界面



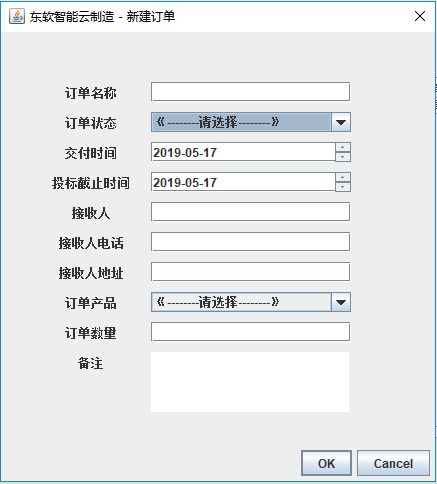


4.1数据字典的增删改查

4.2一个主项包含多个子项，用于下拉框和字段状态等。

5. 订单管理





5.1订单信息的增删改查

5.2新增订单有6种状态，分别为

1. 竞标发布：创建订单时默认状态，此状态可以被各个云工厂投标
2. 中标确定：当已过投标截止时间且有至少一个工厂中标时
3. 订单生产：当至少有一家工厂投入设备生产时
4. 工厂发货：当至少有一家工厂发货时
5. 订单结束：当超级管理员收货时
6. 无竞标：当已过投标截止时间且没有工厂投标

5.3新增订单的订单产品数据来自于产品管理

5.4当订单状态为工厂发货时点击确定按钮收货并结束订单流程

5.5在投标截止日之前，可以查看所有投标工厂的投标信息并可以决定工厂中标，

界面如下：



5.6竞标规则说明：

1. 多家工厂竞标时，承接数量均达到订单需求时，单个报价低者中标
2. 多家工厂竞标时，单个报价低者中标但其承接数量未达到订单需求时将在其余竞标者中选择报价最低者，直至满足订单需求
3. 可自行选择中标单位

6. 产品管理



6.1产品信息的增删改查

6.2基于产品名称和产品编码的查询

7. 用户管理（工厂管理）



7.1查看本工厂的所有用户信息，进行增删改查

7.2只能增加本工厂的工作人员

7.3基于多种字段查询

**8. 设备管理（工厂管理/工人） 【选做模块】**



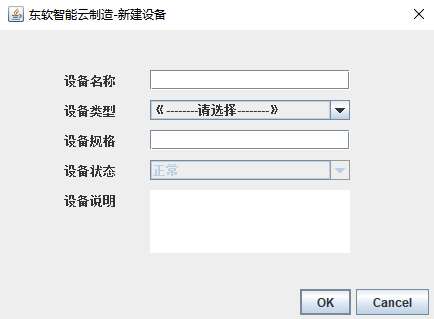
8.1设备信息的增删改查

8.2基于多种字段查询

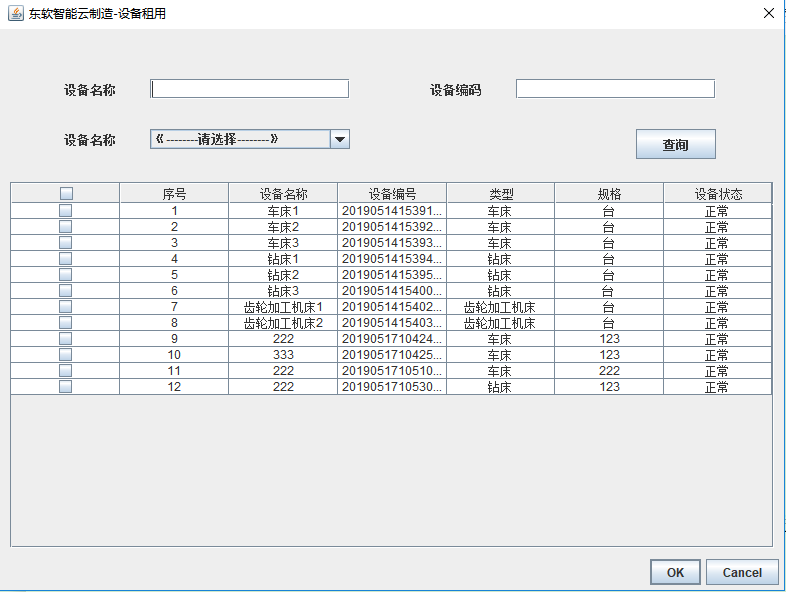
8.3设备所用人为设备拥有者

8.4新增设备时设备类型信息基于数据字典界面，设备类型强制为正常，但设备状态为

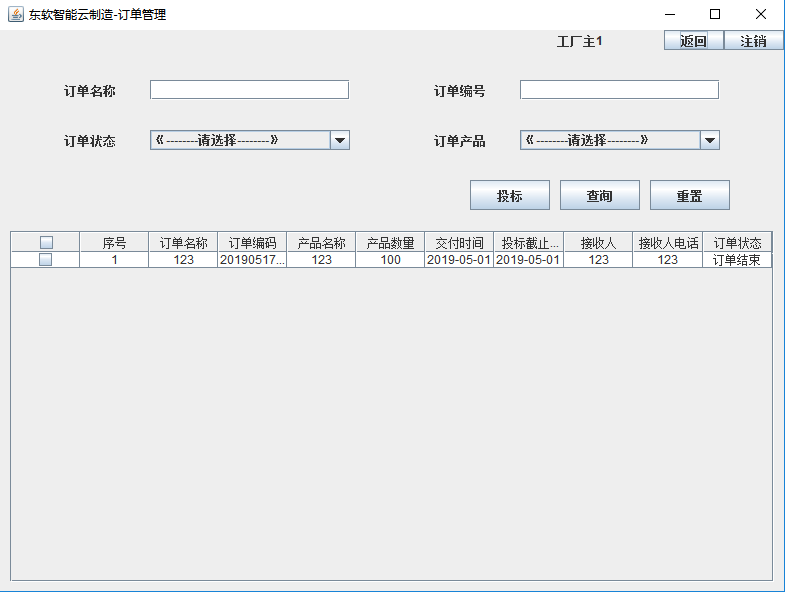
除正常状态外时，该设备不能投入生产，界面如下：



8.5租用设备界面显示所用超级管理员空闲状态的设备，可以进行名称及编码进行查询，界面如下：



**9. 订单管理（工厂管理/工人） 【选做模块】**

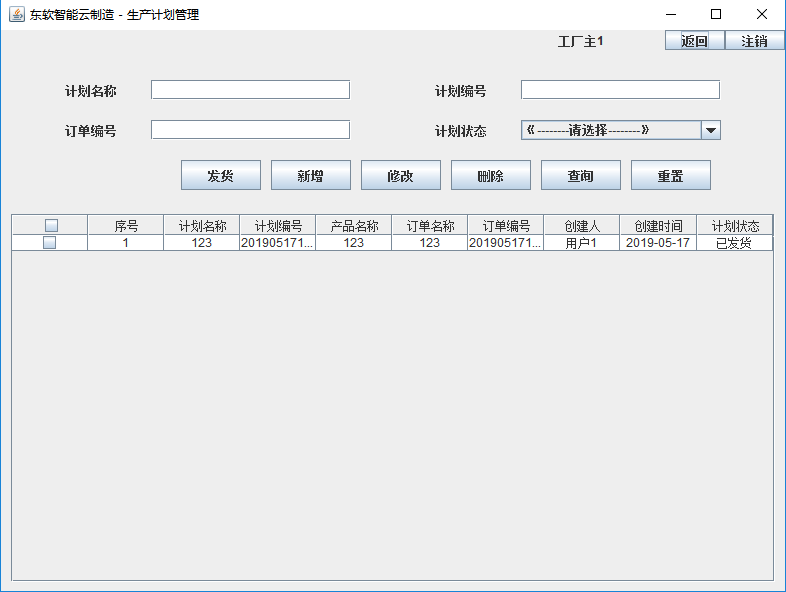
****

9.1显示所有超级管理员发布状态为竞标发布的订单信息

9.2进行投标，一个工厂只能进行一次投标，工人和管理人员均可操作

9.3多字段查询

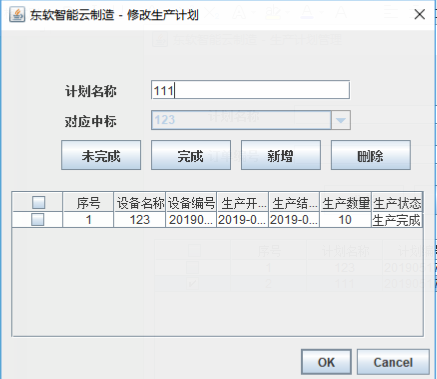
**10. 生产计划管理（工厂管理/工人） 【选做模块】**



10.1当工厂中标后，点击新增选择对应中标即可发布计划，此时订单状态变为生产中

10.2为计划投入设备生产，设备数据来自于设备管理界面

10.3在修改界面选择设备选择完成，此时计划状态更改为已结束，修改界面如下：



10.4当计划状态为生产结束时，选择发货，计划状态更改为已发货

**整体设计**

这里主要涉及的结构体包括：**竞标**（线性），**设备**(线性)，**工厂**（线性），**订单**（线性或哈希），**产品**（线性）。

本文给出参考类结构设计，可自行分析设计自己的类结构。

基本结构（这里只写了必须具备的类成员参考，可以添加其他成员或者完全自己设计类）：

|  |
| --- |
| struct bidEntity //竞标结构  char \* beId ; // 主键  char \* beCode; // 编号  char \* beOrderCode; // 订单编号  char \* bePrice; // 单价  char \* beCount; // 承接数量  char \* beProCount; // 中标数量  char \* beState; // 竞标状态 0 已投标 1 中标 2 未中标  char \* beCreaterCode ;// 竞标人  **Date** beCreate; // 竞标时间 |

|  |
| --- |
| struct DeviceEntity //设施类  char \* deId; // 主键  char \* deName; // 名称  char \* deCode; // 设备编码  char \* deType; // 类型  char \* deSpecifications; // 规格  boolean isRentDevice; // 是否是租用设备 ： 通过创建者类型， 可滤掉此类设备    char \* deState; // 设备状态  char \* deRemark; // 设备说明    char \* deRentState; // 租用状态  char \* deRentUserCode; // 租用人编码  **Date** deRentDate; // 租用日期    char \* deCreater; // 设备创建者  **Date** deCreate; // 设备创建时 间  char \* deUpdater; // 设备更新者  **Date** deUpdate; // 设备更新时间 |

|  |
| --- |
| struct FactoryEntity //工厂类  char \* feId; // 工厂主键  char \* feName ; // 工厂名称  char \* feCode ; // 工厂编码  char \* feRemark; // 工厂说明  char \* feContacts ; // 工厂联系人  char \* feContactsTel ; // 工厂联系人电话  char \* feTel ; // 工厂电话  char \* feAddress; // 工厂地址    char \* feState; // 工厂状态 0 关闭， 1 正常  char \* feCreater ; // 工厂创建者  **Date**  feCreate; // 创建时间  char \* feUpdater; // 更新者  **Date** feUpdate; // 更新时间class |
| struct OrderEntity//订单类  char \* oeId ; // 主键  char \* oeCode; // 订单编号  char \* oeName; // 订单名称    char \* oeState; // 订单状态 1 竞标发布 ：2 中标确认： 3 生产中 ：4 已发货： 5 结束 6 无投标    Date oeDliverDate; // 交付时间  Date oeCutOffDate; // 投标截止时间    char \* oeReceiver; // 接收人  char \* oeReceiverTel; // 接收人电话  char \* oeReceiverAddress; // 接受地址    char \* odeProductCode; // 产品编码  char \* odeCount; // 订单数量  char \* oeRemark;    char \* oeCreater;  Date oeCreate;  char \* oeUpdater;  Date oeUpdate;    ProductEntity product; // 订单产品.  UserEntity creater ;  UserEntity updater;  DataDictionarySubEntity state; |

|  |
| --- |
| class Records //产品类  char \* peId ; // 主键  char \* peCode; // 产品类型编号  char \* peName; // 产品名称  char \* Specifications; // 产品规格  char \* peType; // 产品类 型  char \* peRemark; // 备注 |

# 十 评分标准

|  |  |  |  |
| --- | --- | --- | --- |
| **评价内容** | **评分依据** | **评分标准** | **得分** |
| **分析和设计问题的能力**  **20分** | 1. 能够理解案例需求，能够运用电路图、流程图、类图、ER图等软件工程方法描述问题。  2. 能够设计出合理的存储结构。   3. 能够设计出合理的算法来实现系统要求。   4. 算法具有较好的时间复杂度和空间复杂度。 | A. 能满足第1--4条；  B. 能满足第1--3条；  C. 能满足第1--2条；  D. 能满足第1条；  E. 以上均不满足。 |  |
| **解决问题能力**  **40分** | 1. 能够按照任务书的要求，采用合理的数据结构和算法，实现系统功能。  2. 程序逻辑清晰，代码可读性好。  3. 代码符合编码规范，变量，方法命名统一，设计合理的函数封装，保证函数功能高内聚低耦合。  4. 程序考虑边界情况，测试充分，健壮性好。 | A. 能满足第1--4条；  B. 能满足第1--3条；  C. 能满足第1--2条；  D. 能满足第1条；  E. 以上均不满足。 |  |
| **学习能力**  **20分** | 1. 能够通过网络、课堂、书籍等渠道，使用高级程序设计语言实现数组，链表，栈和队列等线性结构。  2. 能够通过网络、课堂、书籍等渠道，使用高级程序设计语言实现树结构，图结构等非线性结构。  3. 能够通过网络、课堂、书籍等渠道，学习常用的，经典的算法，用于系统功能的实现。  4. 在解决问题的过程中有自己独到的见解，并能够有所创新。 | A. 能满足第1--4条；  B. 能满足第1--3条；  C. 能满足第1--2条；  D. 能满足第1条；  E. 以上均不满足。 |  |
| **报告质量**  **20分** | 1．报告内容完整，正确，充实，叙述清晰准确。  2．报告格式统一，字体大小段落一致，图表编号规范。  3．报告逻辑清晰，可准确反映系统的设计和实现过程。  4．报告用词准确，符合科技文档写作要求。 | A. 能满足第1--4条；  B. 能满足第1--3条；  C. 能满足第1--2条；  D. 能满足第1条；  E. 以上均不满足。 |  |
| **总分** | | |  |