

《程序设计与算法综合实习 I》 课程设计报告

题 目： 物品竞拍管理

班 级：_____

学生学号：_____

学生姓名：_____

指导老师：_____

提交时间：_____

成 绩：_____

华南农业大学 信息学院

目录

一、需求分析	1
1、系统概述	1
2、系统运行环境	2
(1) 硬件环境	2
(2) 软件环境	2
3、功能需求描述	2
二、系统设计	3
1、总体开发思想	3
(1) 面向对象设计思想	3
(2) 模块化开发方法	3
(3) 版本控制与版本管理	3
2、数据结构实现	3
3、数据存储方式	4
(1) 内存存储	4
(2) 持久化存储	4
(3) 使用的语言	5
4、系统模块结构图	6
5、模块设计	6
三、功能实现	8
1、主要函数	8
(1) 物品管理类	8
(2) 用户管理类	9
(3) 物品竞拍管理类	10
2、使用界面	11
四、AI 辅助编码	14
1、简述	14
2、AI 辅助编码展示	15
3、AI 辅助编码思考	15
五、总结	16
六、附录	17

一、需求分析

1、系统概述

在线拍卖作为一种新兴的电子商务模式，因其灵活性和互动性而受到广泛关注。随着互联网技术的不断进步，传统的拍卖方式正逐渐向电子化转型。然而，为了满足日益增长的用户需求和保证交易的公正性，需要一个高效、安全的物品拍卖管理系统。而设计并实现这样的一个系统，不仅需要综合运用多个领域的专业知识，这包括但不限于网络通信、数据库管理、用户界面设计以及算法优化等关键技术，而且需要充分了解竞拍的业务逻辑，将理论与实践相结合。在系统设计阶段，开发者必须深入分析拍卖流程的每一个环节，确保系统能够准确地模拟现实世界的拍卖活动。这包括但不限于拍卖物品的展示、竞拍过程的实时更新、出价的合法性验证、拍卖结束时的结算流程，以及交易后的物流协调等.....

在程序设计与算法综合实习 I 的课程中，物品拍卖系统作为一个实践课题，能够有效地评估学生对复杂软件系统架构设计的理解程度以及理论知识的掌握情况。根据自身现有的编程能力和课程要求，我将物品拍卖系统的核心功能进行了合理的抽象和简化，采用标准为 C11 的 C++ 语言实现，对于实际软件开发中所需技术栈的深入掌握以及可能遇到的技术难题，将在后续的学习过程中进一步探索和理解。基于上述原因，在本次算法实习中，我将归纳为以下几个主要模块：1) 物品管理；2) 用户管理；3) 拍卖会管理；4) 文件的输入与输出处理；5) 信息管理。其中功能见表 1。

表 1 主要功能描述

功能名称	功能描述
录入物品信息	与竞拍相关的物品信息，如物品名、新旧情况、物品描述、估价等
修改商品信息	给定物品的编号，修改该物品的信息
删除商品信息	给定物品的编号，删除该物品的信息
查询商品信息	根据输入的物品的编号或名称，显示该物品的信息
展示同类型商品	显示该类所有物品信息
多用户管理	用户自行管理登记的竞拍物品
模拟竞拍物品	用户对公开的竞拍物品进行出价，可以选择是否显示其他人的出价，最后价高者得
统计功能	一段时间内的竞拍物品数量、总成交价、竞拍者中拍数量排名等
...	详见后续内容

2、系统运行环境

（1）硬件环境

一台个人电脑或笔记本电脑即可，无需服务器或其他专用硬件。

（2）软件环境

操作系统：Windows 或 Linux 操作系统

3、功能需求描述

表 2 系统各功能需求描述

模块	功能	功能描述
物品管理模块	录入物品信息	提供交互界面，允许拍卖方输入物品名称、新旧情况、描述、估价等信息，将其作为新拍卖物品录入系统
	修改物品信息	拍卖方可通过输入物品编号，修改该物品的名称、描述、估价等信息
	删除物品信息	拍卖方可通过输入物品编号，将该物品从系统中移除
	查询物品信息	拍卖方/用户可通过输入物品编号或名称，查看该物品的详细信息
	展示同类物品	系统应提供功能，列出同一类别(如家电、书籍等)的所有拍卖物品
用户管理模块	用户添加	系统可以登记添加一个新的用户
	用户决策	用户可以进行决策，进行是否竞价等操作
	管理拍卖	用户可以查看自己的竞价物品
		用户可查看自己已经拍下的拍卖物品
	出价竞拍	对公开拍卖物品，用户可提交出价，可选择是否显示他人出价情况
统计功能模块	拍卖结束	结束当前拍卖轮次，最高出价者可获得该物品
	显示中标结果	系统应显示每一场拍卖的最终中标结果
	统计物品数量	系统应能统计给定时间段内新增的拍卖物品数量
	统计成交总价	系统应能统计给定时间段内所有拍卖物品的成交总价
文件读写模块	统计用户中标数	系统应能列出给定时间段内，每个用户中标物品数量排名
	导入物品数据	系统应能从文件中导入批量物品信息，初始化拍卖物品库存
	导出拍卖记录	系统应能将历史拍卖记录导出为文件，以备查询和分析

二、系统设计

1、总体开发思想

（1）面向对象设计思想

在本次课程设计实践中，我采用了面向对象的设计（OOD）原则来构建系统。这种方法允许我将系统分解为相互协作的独立对象，每个对象都封装了特定的数据和行为。为了实现这一目标，我创建了一系列的类，每个类都代表了系统中的一个关键概念或实体。例如，用户管理模块被封装为 `User` 类与 `UserQueue` 类，拍卖会管理则对应于 `Auction` 类。通过这种方式，我能够创建一个清晰定义的接口，以及一个易于理解和扩展的系统结构。

由于时间限制限制，我未能实现完整的访问修饰符（如 `private`、`public` 和 `protected`），但我已经确保了类的成员变量和成员函数遵循了最小权限原则，以提高系统的封装性。这为未来的工作奠定了基础，届时可以进一步细化类的封装和继承特性。由于篇幅的限制，在此我仅展示工具类 `Tool` 的类图（如图 1），物品管理类、用户管理类、模拟竞拍管理类下见[附件 1](#)、[附件 2](#)、[附件 3](#)

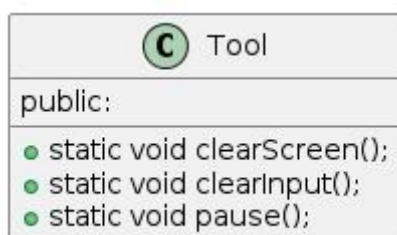


图 1 工具类类图

（2）模块化开发方法

在本次课程设计实践中，我还采用了模块化的开发方法。我利用 C 语言提供的头文件保护机制来避免重复包含问题，并通过精心设计的接口来实现模块之间的松耦合。这种模块化的开发方式不仅使得代码结构更加清晰，而且极大地提高了代码的可读性和可维护性。

（3）版本控制与版本管理

为了提高开发效率，我采用了 Git 作为代码版本管理工具。项目的源代码托管在 GitHub 上，这不仅便于版本控制。通过 Git，可以轻松地跟踪更改，合并分支，使得项目开发者无需通过复杂的方式维护/更新代码。

2、数据结构实现

在本次课设中，主要的数据结构为双向队列与一维链表，如图 2 所示，系统通过访问这两类数据结构中的类实例，调用实例方法进行操作。以单场竞拍为例，当竞拍 **ArtAuction** 需要将一件新的竞拍物品添加进竞拍列表中，只需要调用 **ArtAuction.items.addItem** 方法，即可将新的竞拍物品添加到 **itemList**，而 **itemList** 实质为存储 **item** 实例的链表。具体的类实例方法详见 [附件 1](#)、[附件 2](#) 与 [附件 3](#)。

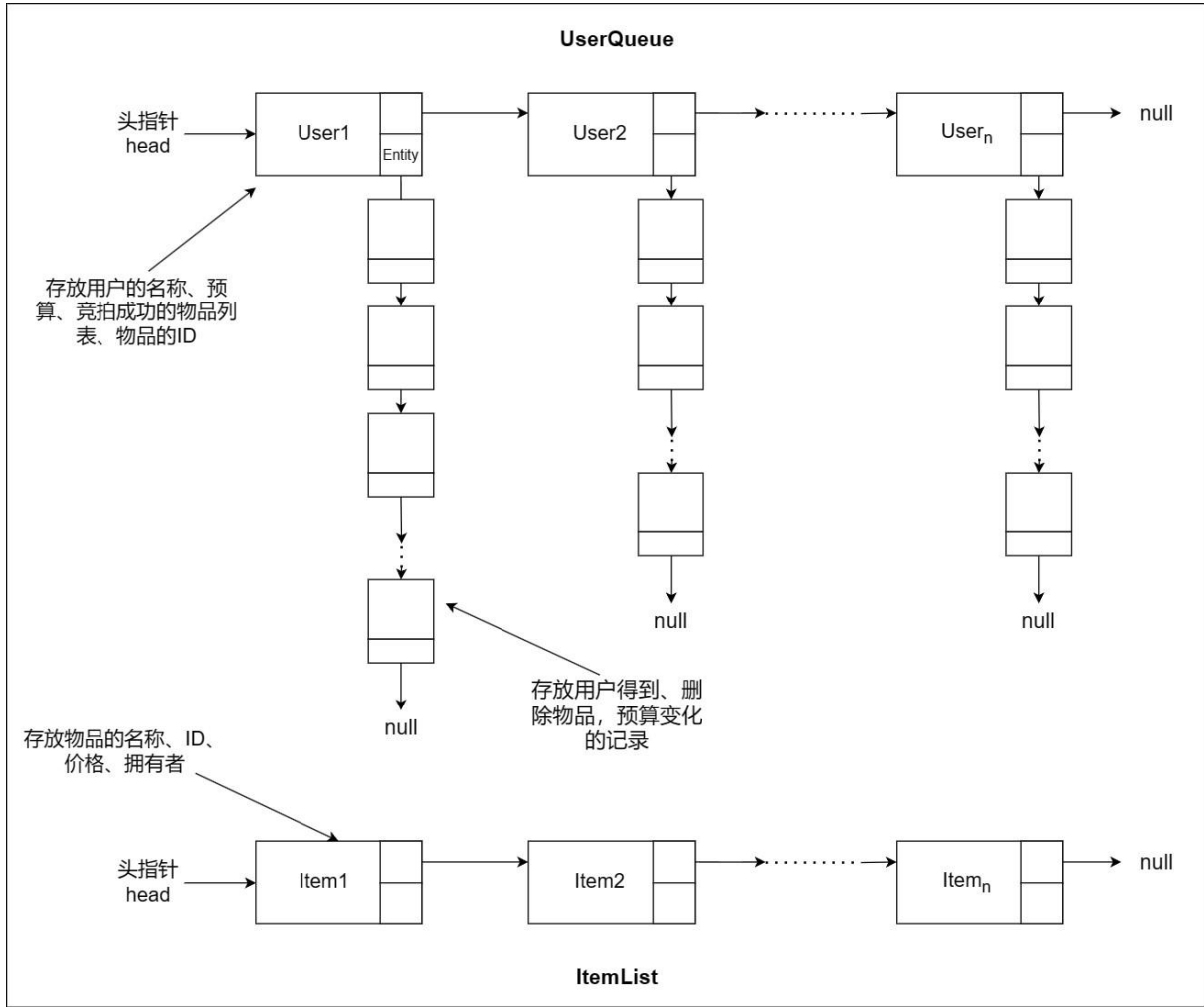


图 2 数据结构图

3、数据存储方式

(1) 内存存储

系统采用的是内存中的数据存储方式。系统运行时将所有物品、用户、拍卖会等数据保存在计算机的内存中，以链表或队列的形式进行管理和操作。这种做法虽然简单高效，但数据无法长期保存，程序关闭后所有数据将丢失。

(2) 持久化存储

为了解决上述程序关闭后会造成数据的丢失问题，该系统使用了 C++ 标准库中的文件流对象 `ifstream` 和 `ofstream` 库，采用了文件存储的方式来持久化数据。具体来说，当系统运行时，所有数据包括物品信息、用户信息、拍卖会信息等都存储在内存中，以链表或队列等数据结构进行管理和操作。当需要保存数据时，系统会调用封装类中的 `save{Class}` 方法，将当前封装类中的各项信息写入指定的文件中。当需要加载保存的数据时，系统会调用封装类的 `load{Class}` 方法，根据从指定文件中读取的内容，重建该封装类某实体的各项信息。因此，该系统实现了在内存和文件之间数据的转换，实现了数据的持久化存储。

（3）使用的语言

该项目使用的 C++11 标准及其标准库，不涉及任何第三方库。

4、系统模块结构图

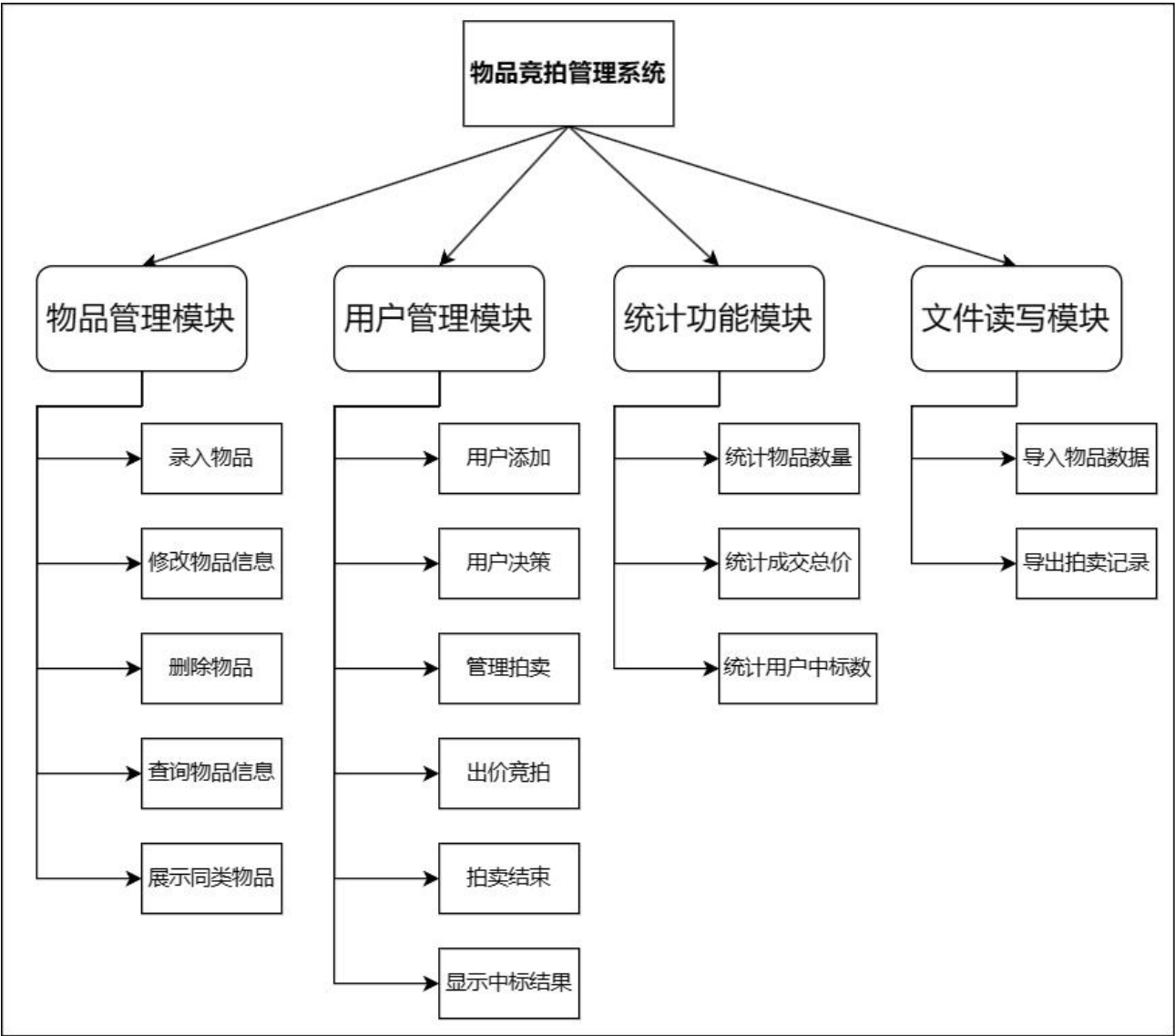


图 3 系统模块结构图

5、模块设计

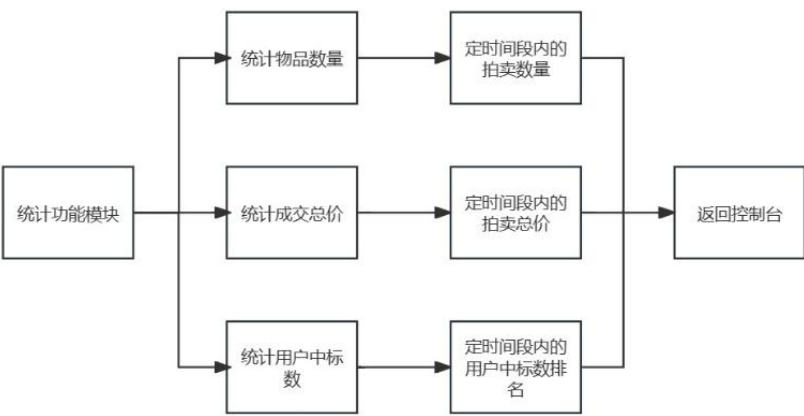


图 4 统计功能模块

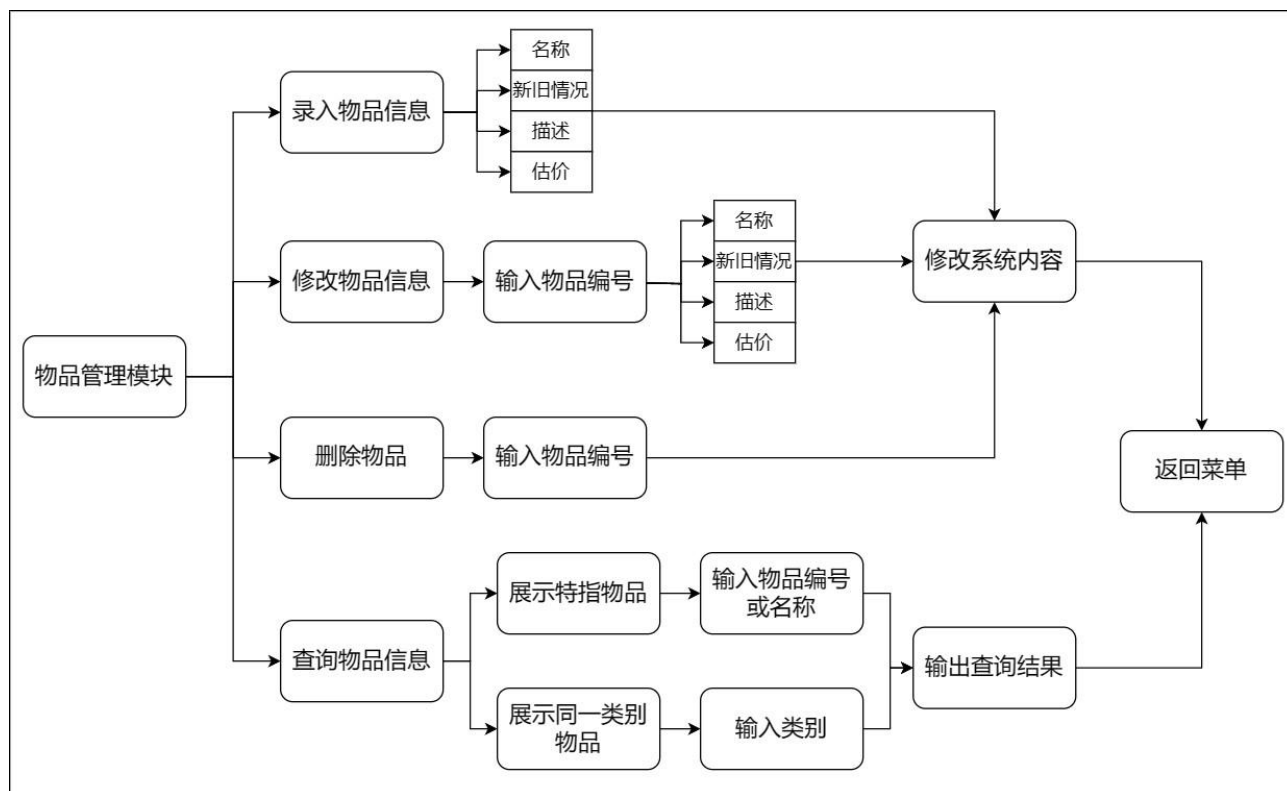


图 5 物品管理模块

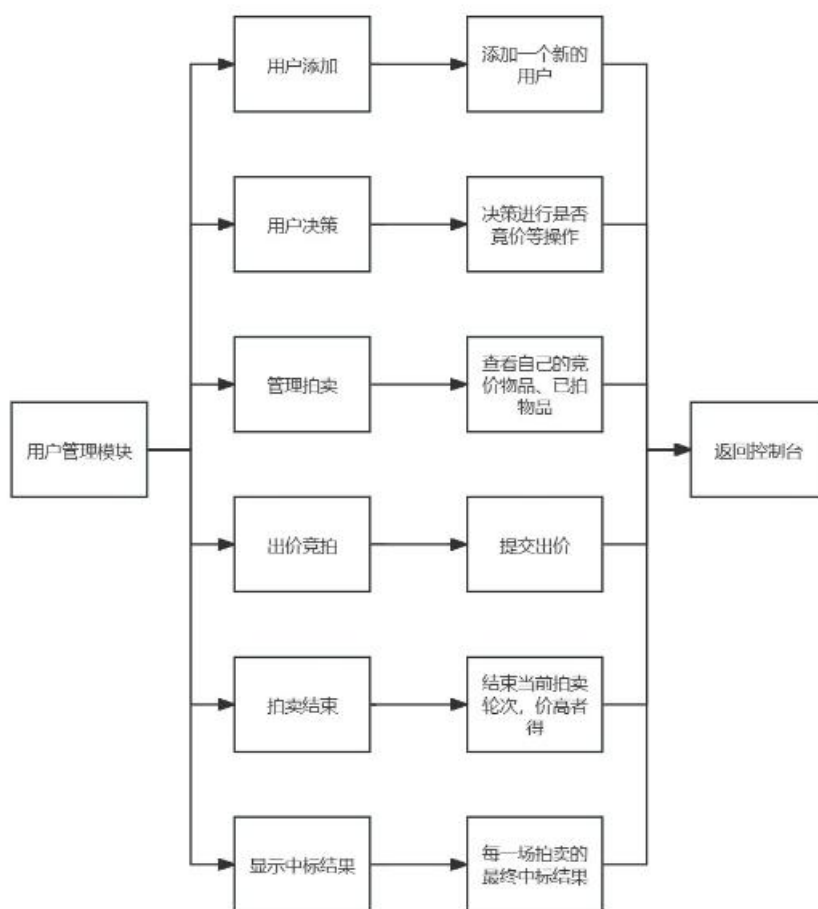


图 6 用户管理模块

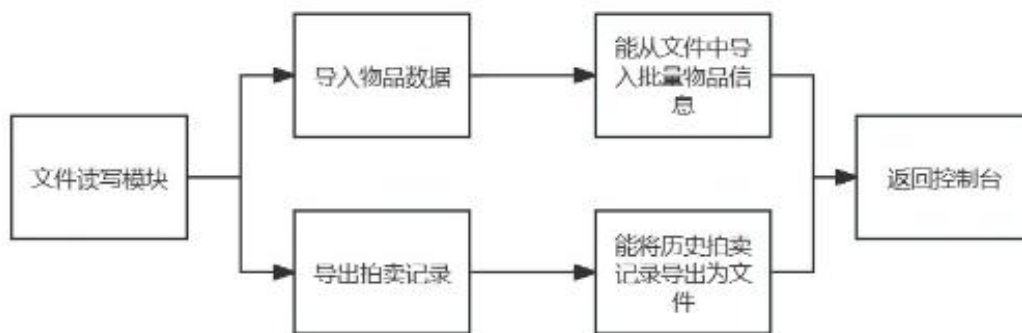


图 7 文件管理模块

三、功能实现

1、主要函数

(1) 物品管理类

表 3 Item 类方法解释

成员函数	描述
Item()	构造一个空的物品对象。
Item(const Item &item)	通过另一个物品对象初始化。
Item(const std::string& name, ...)	使用详细信息初始化物品。
initItem()	初始化物品。
modifyItem(const std::string& modification)	修改物品信息。
addBid(const std::string user, double price)	为物品添加出价。
showBid()	显示物品的所有出价。
showItem()	显示物品信息。
deleteItem()	删除物品。
operator==(const Item& other) const	比较两个物品是否相等。
operator!=(const Item& other) const	比较两个物品是否不等。
setNumber(int num)	设置物品的编号。
setName(std::string n)	设置物品的名称。
setCondition(std::string c)	设置物品的状态。
setDescription(std::string d)	设置物品的描述。
setEstimatedValue(double e)	设置物品的估价。
setType()	设置物品的类型。

表 4 ItemList 类方法解释

成员函数	描述
ItemList()	构造一个空的物品列表。
ItemList(std::initializer_list<Item> items)	使用物品的初始化列表创建物品列表。
ItemList(const std::vector<Item>& items)	使用物品的向量创建物品列表。
~ItemList()	析构函数。

<code>addItem(Item item)</code>	向列表中添加一个物品。
<code>addItem(const std::string& name, ...)</code>	向列表中添加一个用详细信息初始化的物品。
<code>modifyItem(int id, const std::string& modification)</code>	修改指定 ID 的物品信息。
<code>deleteItem(int id)</code>	删除指定 ID 的物品。
<code>showItems()</code>	显示所有物品。
<code>displayItem(int id)</code>	显示指定 ID 的物品。
<code>displayItem(const std::string& name)</code>	显示指定名称的物品。
<code>displayItemsByType(const std::string& type)</code>	显示指定类型的物品。
<code>findItem(int id)</code>	查找并返回指定 ID 的物品。
<code>begin()</code>	获取物品列表的迭代器，指向开始。
<code>end()</code>	获取物品列表的迭代器，指向结束。
<code>size() const</code>	获取物品列表中物品的数量。
<code>empty() const</code>	检查物品列表是否为空。
<code>saveToFile(const std::string& filename)</code>	将物品列表保存到文件。
<code>loadFromFile(const std::string& filename)</code>	从文件加载物品列表。
<code>printItem(const Item& item)</code>	打印指定物品的信息。

(2) 用户管理类

表 5 UserQueue 类方法解释

成员函数	描述
<code>UserQueue()</code>	构造一个空的用户队列。
<code>UserQueue(std::initializer_list<User> users)</code>	使用用户列表初始化用户队列。
<code>UserQueue(const std::deque<User>& users)</code>	使用用户双端队列初始化用户队列。
<code>findUser(const std::string& name)</code>	查找具有指定名称的用户。
<code>switchUser()</code>	切换当前用户。
<code>addUser(const User& user)</code>	向队列中添加一个用户对象。
<code>addUser()</code>	添加一个用户。
<code>showUsers() const</code>	显示队列中的所有用户。
<code>deleteUser(const std::string& name)</code>	删除队列中指定名称的用户。
<code>getUsers() const</code>	获取用户队列的一个副本。
<code>size() const</code>	返回队列中用户的数量。
<code>empty() const</code>	检查队列是否为空。
<code>front()</code>	返回队列中第一个用户的引用。
<code>back()</code>	返回队列中最后一个用户的引用。
<code>begin()</code>	返回指向队列开始的迭代器。
<code>end()</code>	返回指向队列结束的迭代器。
<code>push(const User& user)</code>	在队列尾部添加一个用户。
<code>pop()</code>	移除队列中的第一个用户。
<code>saveToFile(const std::string& filename)</code>	将用户队列保存到指定的文件。
<code>loadFromFile(const std::string& filename)</code>	从指定的文件加载用户队列。
<code>getItemOnAuction(Bid &HighestBid, Item &item, UserQueue &users)</code>	静态方法，将拍卖成功的物品加入 UserOwnItem

表 6 User 类方法解释

成员函数	描述
User()	构造一个空的用户对象。
User(const std::string& name, double budget)	使用名称和预算初始化用户。
InitUser()	初始化用户。
getName()	获取用户的名称。
getBudget()	获取用户的预算。
displayItem(int id)	显示指定 ID 的物品。
displayItem(const std::string& name)	显示指定名称的物品。
displayItemsByType(const std::string& type)	显示指定类型的物品。
deleteItem(int id)	删除指定 ID 的物品。
bidItem(int id, double price, ItemList &items)	用户对指定 ID 的物品进行出价。
bitItem(Item &item, double price)	用户对物品进行出价。
getItem(int id, ItemList &items)	获取指定 ID 的物品。
getItem(Bid HighestBid, Item& item)	获取最高出价的物品。
operator==(const User& other) const	比较两个用户是否相等。
operator=(const User& other)	赋值运算符，复制用户信息。
operator&()	地址运算符。

(3) 物品竞拍管理类

表 7 Auction 类方法解释

成员函数	描述
Auction()	构造一个没有名称和日期的拍卖。
Auction(const std::string& name, const std::string& date)	使用名称和日期初始化拍卖。
void InitItem()	初始化拍卖的物品。
void InitItem(ItemList& items)	使用提供的物品列表初始化拍卖的物品。
void InitUser()	初始化拍卖的用户。
void InitUser(UserQueue& users)	使用提供的用户队列初始化拍卖的用户。
void InitAuction()	初始化拍卖。
void InitAuction(const std::string& name, const std::string& date)	使用提供的名称和日期初始化拍卖。
bool checkAuction()	检查拍卖是否准备好进行。
void showAuction()	显示拍卖信息。
void start()	开始拍卖。
int getItemsCount()	获取拍卖中物品的数量。
int getBiddedCount()	获取已出价的物品数量。
double getTotalAmount()	获取拍卖的总金额。
void settleAuction(Item& item)	结算拍卖的物品。
void showUserOwnItems()	显示用户拥有的物品。
void showUserOwnItems(Auction auction)	显示拍卖中用户拥有的物品。
void auctionSelect()	拍卖选择菜单（具体作用未提供）。
void saveAuction(const std::string& filename)	将拍卖信息保存到文件。

<code>void loadAuction(const std::string& filename)</code>	从文件加载拍卖信息。
<code>int menu(User& user, Item& item)</code>	显示拍卖菜单并处理用户选择。
<code>std::string getName() const</code>	获取拍卖的名称。
<code>std::string getDate() const</code>	获取拍卖的日期。

2、使用界面

```

C:\Users\m1354\Desktop x + - □ ×
Id: 1
Name: item1
Condition: Great
Description: A test item
Estimated Value: 100
Types: A B C
Id: 2
Name: item2
Condition: Good
Description: Another test item
Estimated Value: 200
Types: D B F
Id: 3
Name: item3
Condition: Bad
Description: Yet another test item
Estimated Value: 300
Types: A H C
Press enter to continue...

```

(a) 初始化竞拍物品

```

C:\Users\m1354\Desktop x + - □ ×
User name: user1 Budget: 200
User name: user2 Budget: 300
Press enter to continue...

```

(b) 初始化用户

```

C:\Users\m1354\Desktop x + - □ ×
-----
| Auction name: Auction1 |
| Auction date: 2024-05-06 |
-----
| The list of items is as follows: |
Id: 1
Name: item1
Condition: Great
Description: A test item
Estimated Value: 100
Types: A B C
Id: 2
Name: item2
Condition: Good
Description: Another test item
Estimated Value: 200
Types: D B F
Id: 3
Name: item3
Condition: Bad
Description: Yet another test item
Estimated Value: 300
Types: A H C
-----
| The list of users is as follows: |
User name: user1 Budget: 200
User name: user2 Budget: 300
-----
Press enter to continue...

```

(c) 初始化拍卖会

```

C:\Users\m1354\Desktop x + - □ ×
-----
| User: user1 |
| Current Item: item1 |
-----
| 1. View the bids of others for the current
   auction item |
| 2. View detailed information of the curren
   t auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction ite
   ms |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to
   operate |
| Please choose the operation: |

```

(d) 拍卖会操作界面

图 8 调用 loadAuction 方法，读取已有拍卖会¹

¹ 操作界面中的 Press enter to continue 是 Tool 类中的 Pause 方法，Tool 类中其余方法详见图 1

```
-----
| User: user2 |
| Current Item: item1 |
-----
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
1
Bids for item item1:
Bid number: 1
User: user1
Price: 80
Press enter to continue...
```

(a) 查看他人出价

```
-----
| User: user1 |
| Current Item: item1 |
-----
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
2
Number: 1
Name: item1
Condition: Great
Description: A test item
Estimated value: 100
Types: A B C
Press enter to continue...
```

(b) 查看当前竞拍物品细节

```
-----
| User: user1 |
| Current Item: item1 |
-----
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
3
Enter your bid: 80
Bid added successfully.
Bid successful.
Press enter to continue...Press enter to continue...
```

(c) 竞价

```
-----
| User: user1 |
| Current Item: item2 |
-----
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
4
Id: 1
Name: item1
Condition: Great
Description: A test item
Estimated Value: 100
Types: A B C
Press enter to continue...
```

(d) 查看已经购买物品

```
-----
| User: user1 |
| Current Item: item1 |
-----
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
5
Your budget is: 200
Press enter to continue...
```

(e) 查看余额

```
-----
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
ms |
| 7. Search for an auction item |
| 8. Give up the operation, the next user operate |
| Please choose the operation: |
6
The list of subsequent auction items is as follows:
Number: 1
Name: item1
Condition: Great
Description: A test item
Estimated value: 100
Types: A B C
Number: 2
Name: item2
Condition: Good
Description: Another test item
Estimated value: 200
Types: D B F
Number: 3
Name: item3
Condition: Bad
Description: Yet another test item
Estimated value: 300
Types: A H C
Press enter to continue...
```

(6) 查看后续拍卖物品

图 9 用户基础操作（菜单 1-6 选项）

```
C:\Users\m1354\Desktop x + - □ x
| Current Item: item2 |
|-----|
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to operate |
| Please choose the operation: |
7
Please enter the search mode:
1 represents search by name
2 represents search by id
3 represents search by category: 1
Please enter the name of the item to be searched: item1
Id: 1
Name: item1
Condition: Great
Description: A test item
Estimated Value: 100
Types: A B C
Press enter to continue...Press enter to continue...
```

(a) 按照物品名搜索物品

```
C:\Users\m1354\Desktop x + - □ x
| Current Item: item2 |
|-----|
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to operate |
| Please choose the operation: |
7
Please enter the search mode:
1 represents search by name
2 represents search by id
3 represents search by category: 2
Please enter the id of the item to be searched: 2
Id: 2
Name: item2
Condition: Good
Description: Another test item
Estimated Value: 200
Types: D B F
Press enter to continue...Press enter to continue...
```

(b) 按照 Id 进行搜索²

```
C:\Users\m1354\Desktop x + - □ x
t auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to operate |
| Please choose the operation: |
Please enter the search mode:
1 represents search by name
2 represents search by id
3 represents search by category: 3
Please enter the category of the item to be searched: A
Id: 1
Name: item1
Condition: Great
Description: A test item
Estimated Value: 100
Types: A B C
Id: 3
Name: item3
Condition: Bad
Description: Yet another test item
Estimated Value: 300
Types: A H C
Press enter to continue...Press enter to continue...
```

(c) 搜索显示相同类别物品

```
C:\Users\m1354\Desktop x + - □ x
|-----|
| User: user1 |
| Current Item: item2 |
|-----|
| 1. View the bids of others for the current auction item |
| 2. View detailed information of the current auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction items |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to operate |
| Please choose the operation: |
8
The next user operates
Press enter to continue...
```

(d) 放弃操作

图 10 搜索操作与放弃（菜单 7、8 选项）

² 在运行程序时，会初始化 Item.cpp 中的静态变量 Id，后续所有物品都会 Id 基础上自增 1，以实现所有物品的 Id 值唯一


```
C:\Users\m1354\Desktop x + - □ ×

-----
| User: user2 |
| Current Item: item3 |
-----
| 1. View the bids of others for the current
| auction item |
| 2. View detailed information of the curren
t auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction ite
ms |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to
operate |
| Please choose the operation: |
8
The next user operates
Press enter to continue...
The auction for the item item3 has ended
The item item3 was purchased by user2 at a p
rice of 100
user2
User found
Item successfully purchased.
Press enter to continue...
```

(a) 搜索最高出价，该用户竞拍成功

```
C:\Users\m1354\Desktop x + - □ ×

-----
| User: user1 |
| Current Item: item2 |
-----
| 1. View the bids of others for the current
| auction item |
| 2. View detailed information of the curren
t auction item |
| 3. Bid |
| 4. View the goods you have purchased |
| 5. View your balance |
| 6. List the list of subsequent auction ite
ms |
| 7. Search for an auction item |
| 8. Give up the operation, the next user to
operate |
| Please choose the operation: |
8
The next user operates
Press enter to continue...
The auction for the item item2 has ended
No one bid for the item item2
Press enter to continue...
```

(b) 无人竞拍，该物品没有人获取

图 11 竞价物品操作（逻辑实现³）

四、AI 辅助编码

1、简述

AI 辅助编码是指利用人工智能技术来增强软件开发过程中的代码编写活动。随着计算能力的提高和机器学习算法的进步，AI 在自动生成代码、调试、测试以及优化软件性能方面发挥着越来越重要的作用。AI 辅助编码的核心优势在于其能够理解和预测开发者的意图，从而提供实时的代码建议、自动补全和错误检测。这不仅提高了编码的速度和质量，还有助于减少重复性工作，让开发者能够专注于更复杂和创造性的任务。

目前，采用 AI 辅助编码主要有三种方式：

①直接采用通用大模型，利用大模型的隐性知识进行辅助编码，譬如 ChatGPT、Claude 等大型语言模型；

³ 竞拍的方式采取了队列模拟，当一轮竞拍中，所有人放弃竞拍，则开始核算当轮竞拍。核算时，如果该物品有竞价，则最高出价用户竞拍成功，若无人竞拍，则开始竞拍下一个物品。

②利用 LangChain 等技术，搭建更加具有针对性的智能体 AI Agent 进行辅助编码，目前有不少老师、同学在研究这一方面，笔者在今天的大创答辩中就听到了以业务流程为导向的代码纠错大模型，该方式比①相比，更加面向实际业务，并且更加轻量级。

③训练编程垂类模型，譬如 CitHub Copilot，利用 GitHub 亿级别的开源代码进行训练，该方式训练出来的大模型性能更加优越，逻辑推理能力、辅助编程能力更强。

由于时间问题，笔者暂且按笔，假以时日再做三类方式的优缺点分析。在此主要提一个发现的 AI 辅助编码的思考。

2、AI 辅助编码展示

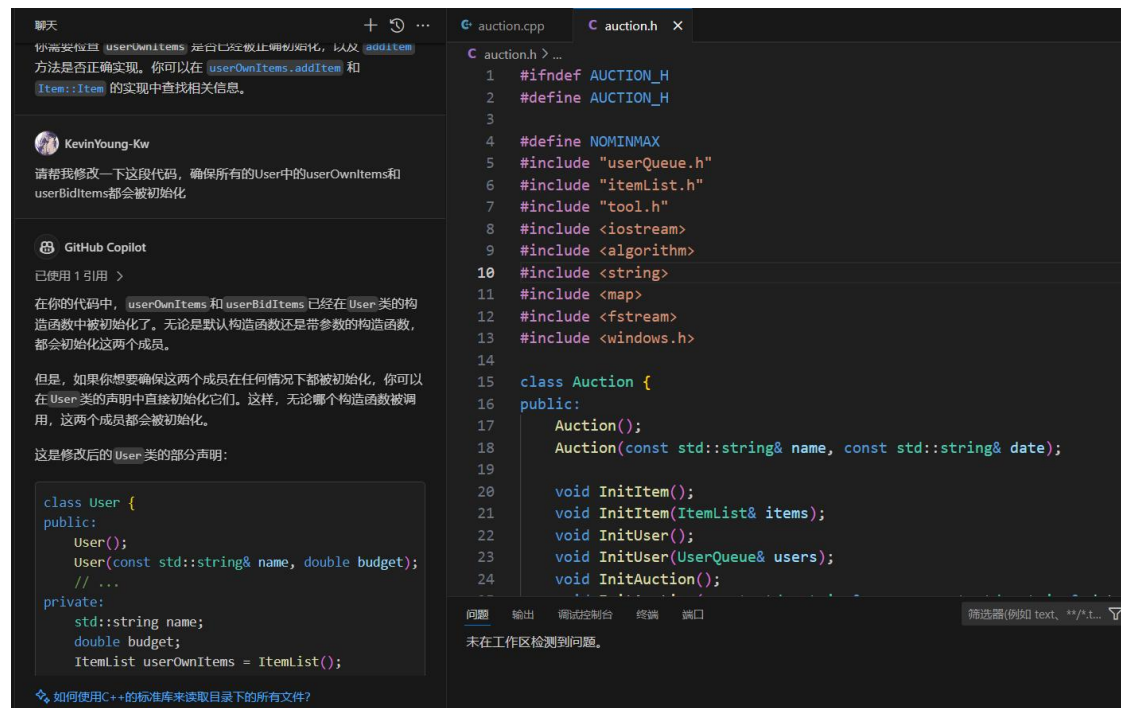


图 12 GitHub Copilot 辅助编译

3、AI 辅助编码思考

在大型软件开发的过程中，代码量往往非常庞大，动辄涉及数万甚至数十万行代码。在这样的开发规模下，AI 辅助编码技术的应用显得尤为重要。然而，现有的 AI 模型在处理如此庞大的数据时，可能会遇到输入上下文窗口的限制问题。即使模型能够处理如此大量的数据，其编码解码（Encoder, Decoder）机制以及注意力（Attention）机制可能导致输出结果与预期目标存在较大偏差。

笔者在模块化开发实践中发现，通过精心设计的模块化结构、接口以及类底层架构，可以有效地利用 AI 模型的抽象调用能力，实现更为复杂的功能。这种设计思想可以从宏观到微观逐步细化，也可以从微观到宏观逐步扩展。

具体来说，宏观到微观的方法涉及让 AI 模型首先提出一个高层次的抽象实现策略。在此基础上，逐步细化具体的实现细节，直至达到具体的代码实现。这种方法的优势在于能够利用 AI 的全局视角，为复杂问题提供一个高层次的解决方案框架。而微观到宏观的方法则是从构建基础的类和方法开始，让 AI 模型直接调用这些基础组件来实现更复杂的功能。这种方法的优势在于能够充分利用 AI 在模式识别和自动化方面的能力，减少重复性编码工作，提高开发效率。

综上所述，如果能设计一个大模型，专门进行模块化与接口设计的自动化，用编程大模型进行基本的代码撰写，用模块化与接口设计大模型进行封装，封装的结果会提高编程大模型的抽象调用能力，就可以实现软件开发的全过程自动化。

五、总结

本次课程设计的核心目标是开发物品拍卖管理系统。通过深入分析在线拍卖的业务流程和需求，设计并实现了一个基于 C++11 标准的物品竞拍管理系统。该系统涵盖了物品管理、用户管理、拍卖会管理等基本功能，通过模块化设计和面向对象编程原则来提高系统的可维护性和可扩展性。

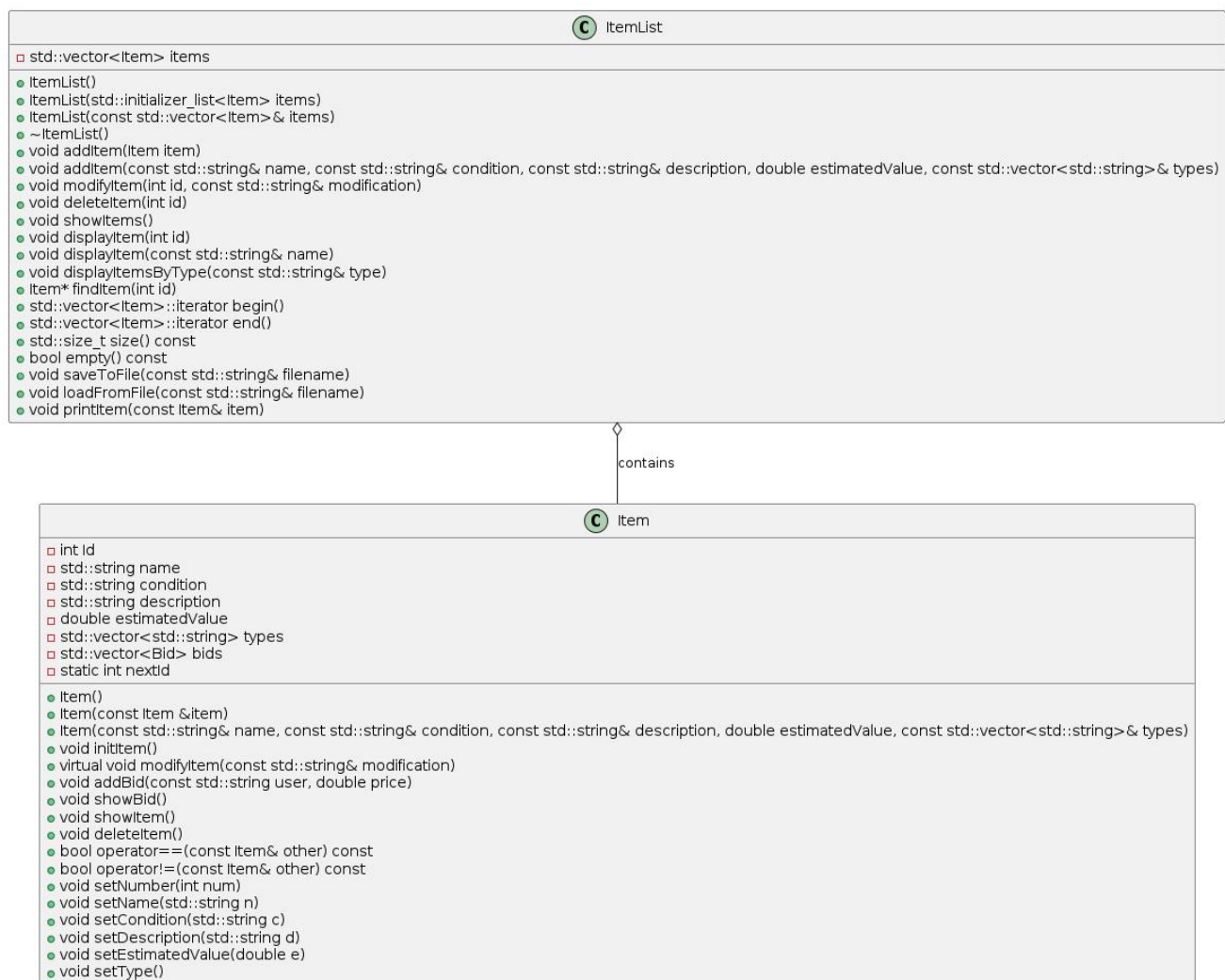
在技术实现方面，该系统采用了内存存储与文件存储相结合的方式，利用 C++ 标准库中的文件流对象在两者之间进行数据转换，确保了数据的即时性和持久性。系统设计遵循面向对象思想，通过类和对象的封装，实现了对复杂业务逻辑的简化和抽象。

功能方面，系统实现了物品信息的增删改查、用户管理以及拍卖流程模拟等核心需求，并提供了统计功能，能够对历史拍卖活动进行数据分析，为管理决策提供支持。

值得一提的是，本次设计中探讨了 AI 辅助编码的应用。分析了当前主要的 AI 辅助编码方式，并结合模块化设计思想，提出了一种适合该系统的 AI 辅助编码技巧，以期在未来实现更高层次的自动化开发。

由于时间的关系，既定的开发目标没能够实现，对于进程调度、多用户交互的代码实现问题，将在后续的学习过程中进一步探索和理解。

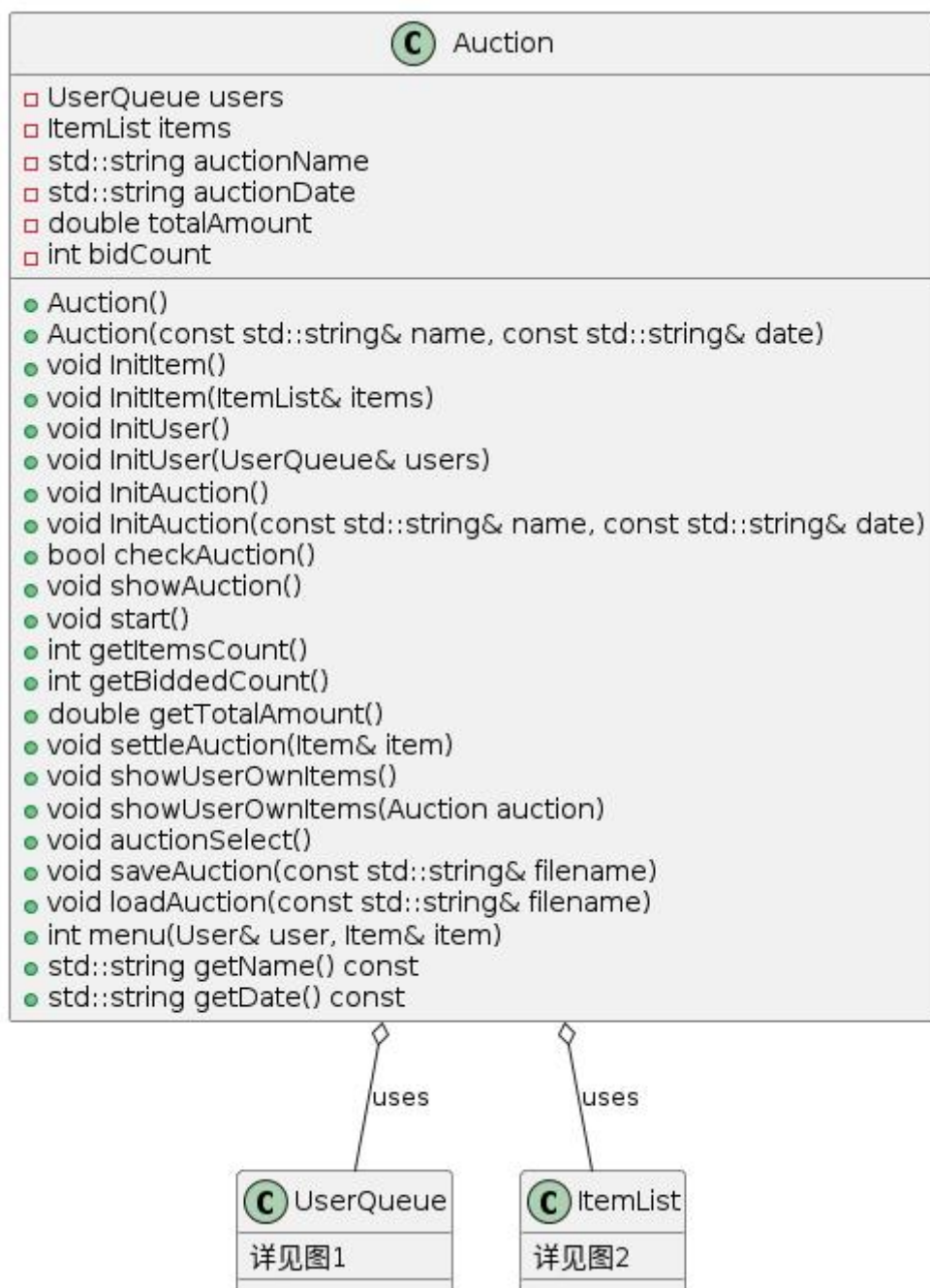
六、附录



附件 1 物品管理类图



附件 2 用户管理类图



附件 3 物品竞拍管理类图

评分表

题目：

学号:

姓名：

评分项目

分数

1. 功能是否满足要求（50 分）
2. 界面是否美观（10 分）
3. 所学知识在设计中的运用是否得当（10 分）
4. 对效率要求较高的部分是否有更深的思考与设计（10 分）
5. 答辩情况（20 分）

总成绩:

评语:

评阅老师：
日

评阅时间： 年 月

