# 茅台酒溯源系统

# 信息安全基础编程作业

#### 姓名：李若冰

#### **学号：2251657**

#### **指导教师：尹常青**

#### 专业：软件学院\软件工程专业

### 项目简介

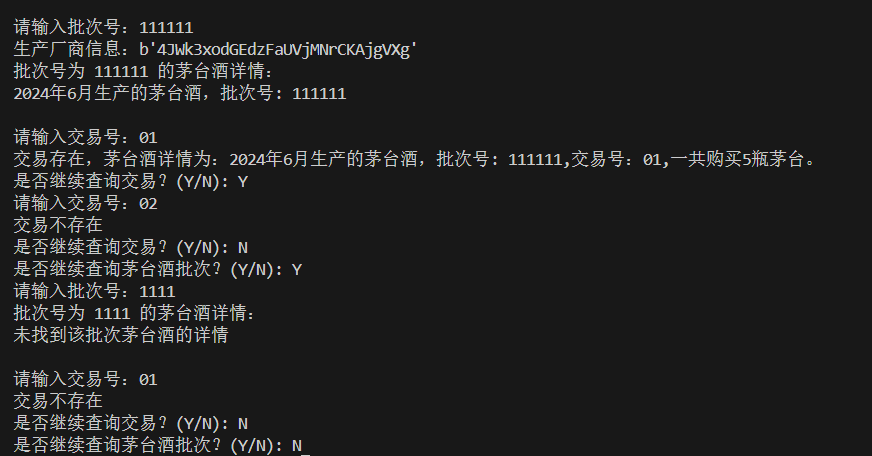
##### 1.1项目要求

使用区块链技术完成一个茅台酒溯源程序的小型demo。

##### 1.2实现思路

1. 定义区块和区块链：创建一个类来表示区块，其中包含时间戳、数据、上一个区块的哈希和当前区块的哈希。创建一个类来表示整个区块链，其中包含一个区块链列表和一些操作区块链的方法。
2. 创建茅台酒溯源的数据结构：定义一个类来表示茅台酒的溯源数据，包含生产者、批次号等信息。
3. 实现茅台酒溯源的交易：创建一个类来表示茅台酒溯源的交易，包含发送者、接收者、交易件数和溯源数据等信息。在交易中添加验证密钥的步骤，确保交易的有效性。
4. 实现茅台酒溯源的功能：确保每个茅台酒溯源交易都被验证并添加到区块链中,如创建茅台酒溯源交易、添加区块等。
5. 提供用户接口：实现一个简单的用户接口，让用户能够通过命令行或其他方式与茅台酒溯源程序进行交互。用户可以查询特定批次号的茅台酒溯源信息和购买的交易信息。

**1.3项目运行示例**



### **区块链技术简介**

区块链技术在茅台酒溯源中能够提供去中心化、不可篡改性、透明性、可追溯性、安全性和去信任化等特性，为茅台酒溯源提供了一种可靠且高效的解决方案。

* **去中心化**：区块链是一种去中心化的技术，数据存储在网络中的多个节点上，而不是集中存储在单个中心服务器上。在茅台酒溯源中，多个参与方（如生产商、经销商、消费者等）可以成为网络的节点，每个节点都有一份完整的茅台酒溯源数据副本。这种去中心化的结构消除了单点故障，提高了数据的可靠性和安全性。
* **不可篡改性**：区块链中的每个区块都包含了前一个区块的哈希值，形成了一个由链接的区块组成的链。这种链式结构使得区块链中的数据具有不可篡改性，一旦数据被写入区块链，就很难篡改或删除。在茅台酒溯源中，一旦茅台酒的生产信息被写入区块链，就能够确保溯源数据的真实性和可信度，防止数据被篡改或伪造。
* **透明性和可追溯性**：区块链技术提供了高度的透明性和可追溯性。每个参与方都可以查看和验证区块链中的数据，并且无法对数据进行擅自修改。茅台酒溯源中的每一笔交易和相关信息都被记录在区块链中，可以追溯到茅台酒的生产环节，确保了生产过程的透明性和可追溯性。
* **安全性**：区块链使用密码学技术确保数据的安全性。每个交易都使用数字签名进行验证，确保交易的真实性和完整性。茅台酒溯源中的数据可以通过密码学算法进行加密，保护数据的隐私性和机密性。
* **去信任化**：区块链技术通过共识机制和智能合约等机制实现了去信任化。茅台酒溯源中的参与方可以通过区块链网络自动执行约定的规则和逻辑，无需相互信任。这种去信任化的特性可以减少人为的错误和欺诈行为，提高数据的可信度和可靠性。

### **项目具体实现**

##### 3.1功能类介绍

- `BlockChain`：表示区块链，包含一个区块列表。

- `Block`：表示区块，包含一个交易列表和前一个区块的哈希值。

- `Transaction`：表示交易，包含发送者、接收者、数量、批次号和详情。

- `Wallet`：表示一个账户，包含地址和余额。

**3.2功能类具体实现**

###### 3.2.1Block



**功能介绍：**

**表示区块，并提供了计算区块哈希值的功能**。它的构造函数初始化了区块的属性，并在初始化过程中计算了区块的哈希值。\_\_repr\_\_方法用于方便地显示区块的信息。

属性：

* prev\_hash：父区块的哈希值。
* transactions：交易对，表示该区块中的交易信息。
* timestamp：区块创建的时间戳。
* hash：区块的哈希值。
* nonce：随机数。

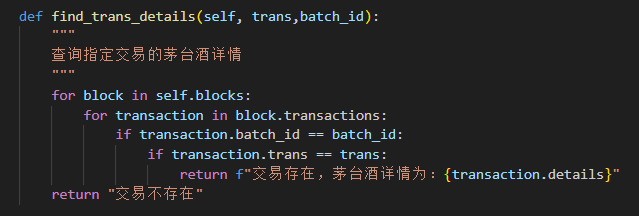
方法：

* \_\_init\_\_(self, transactions, prev\_hash)：类的构造函数，用于初始化区块的属性。它接受交易对和父区块的哈希值作为参数，并将它们保存到类的属性中。然后，它获取当前时间作为区块的创建时间，并将nonce的初始值设置为None。接下来，它计算区块的哈希值，使用SHA-256哈希算法对父哈希值、交易对和时间戳进行哈希运算，并将结果保存到hash属性中。
* \_\_repr\_\_(self)：类的字符串表示方法，用于打印或显示实例化对象的信息。它返回一个包含父哈希值、区块内容和区块哈希值的字符串。

**功能输出：**



###### 3.2.2BLockChain



**功能介绍：**

**表示区块链，并提供了添加区块、打印区块链、查询批次号和交易号对应的茅台酒详情**等功能。通过add\_block方法可以将区块添加到区块链中，print\_list方法可以打印区块链中的区块信息。find\_batch\_details方法用于查询指定批次号的茅台酒详情，find\_trans\_details方法用于查询指定交易的茅台酒详情。

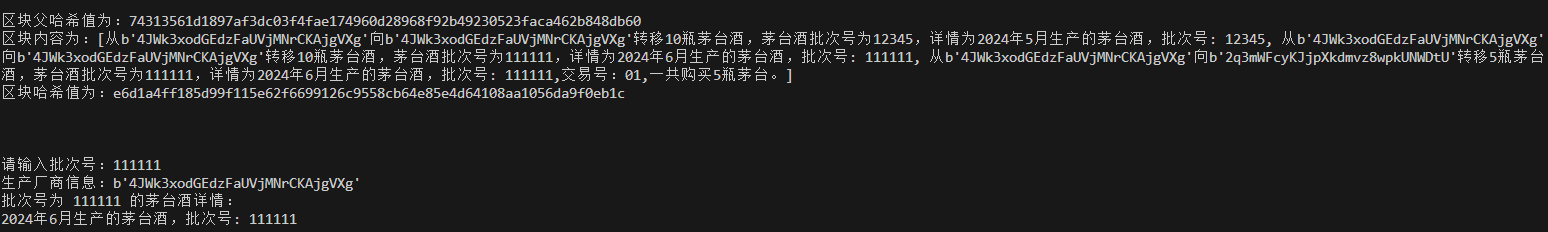
属性：

* blocks：包含的区块列表。

方法：

* \_\_init\_\_(self)：类的构造函数，用于初始化区块链的属性。它创建一个空的区块列表blocks。
* add\_block(self, block)：添加区块到区块链中。它接受一个Block对象作为参数，并将其添加到blocks列表中。
* print\_list(self)：打印区块链中的区块信息。它遍历blocks列表，并打印每个区块的父哈希值、内容和哈希值。
* find\_batch\_details(self, batch\_id)：查询指定批次号的茅台酒详情。它遍历区块链中的每个区块和交易，检查交易的批次号是否与给定的批次号匹配。如果找到匹配的批次号，它将返回该交易的茅台酒详情；否则，返回未找到该批次茅台酒的提示信息。
* find\_trans\_details(self, trans, batch\_id)：查询指定交易的茅台酒详情。它遍历区块链中的每个区块和交易，检查交易的批次号和交易号是否与给定的批次号和交易号匹配。如果找到匹配的交易，它将返回该交易的茅台酒详情；否则，返回交易不存在的提示信息。

**功能输出（部分示例）：**



**3.2.3Transaction**



**功能介绍：**

表示交易，并提供了设置签名和公钥的功能。构造函数用于初始化交易的属性，set\_sign方法用于设置交易的签名和公钥。\_\_repr\_\_方法用于方便地显示交易的信息。交易包含发送方、接收方、交易数量以及茅台酒的批次号和详情。

属性：

* pubkey：交易发送者的公钥。
* signature：交易的数字签名。
* sender：发送方的标识。
* recipient：接收方的标识。
* amount：交易数量。
* batch\_id：茅台酒的批次号。
* details：茅台酒的详情。
* trans：茅台酒的详情。

方法：

* \_\_init\_\_(self, sender, recipient, amount, batch\_id=None, details=None, trans=None)：类的构造函数，用于初始化交易的属性。它接受发送方、接收方、交易数量以及茅台酒的批次号和详情作为参数，并将它们保存到类的属性中。pubkey和signature的初始值设置为None。
* set\_sign(self, signature, pubkey)：设置交易的签名和公钥。它接受数字签名和公钥作为参数，并将它们设置到相应的属性中。
* \_\_repr\_\_(self)：类的字符串表示方法，用于打印或显示交易的信息。它返回一个包含发送方、接收方、交易数量、茅台酒批次号和详情的字符串。

**功能输出（部分示例）：**



**3.2.4Wallet**



**功能介绍:**

属性：

* \_private\_key：钱包的私钥，基于椭圆曲线生成的一个唯一秘钥。
* \_public\_key：钱包的公钥，基于私钥生成的公钥。
* balance：钱包的余额。

方法：

* \_\_init\_\_(self)：类的构造函数，用于初始化钱包的属性。在初始化时，它会生成一个唯一的私钥，并基于私钥生成对应的公钥。同时，将初始余额设置为10000。
* address(self)：返回钱包的地址。它通过对公钥进行哈希和Base58编码生成唯一的地址。
* pubkey(self)：返回钱包的公钥字符串。
* sign(self, message)：生成给定消息的数字签名。它将消息进行哈希，并使用私钥对哈希结果进行签名。
* verify\_sign(pubkey, message, signature)：验证给定公钥、消息和签名的有效性。它将加载公钥，并尝试使用公钥验证签名是否与消息匹配。如果验证成功，返回True；否则，返回False。
* 这个类的作用是表示钱包，并提供了生成地址、生成数字签名、验证签名等功能。在初始化时，钱包会生成一个唯一的私钥和对应的公钥。通过address方法可以获取钱包的地址，通过pubkey方法可以获取钱包的公钥字符串。sign方法用于生成给定消息的数字签名，verify\_sign方法用于验证给定公钥、消息和签名的有效性。除此之外，钱包还具有余额属性balance，可以用于存储和管理钱包的余额信息。

**3.3demo示例**

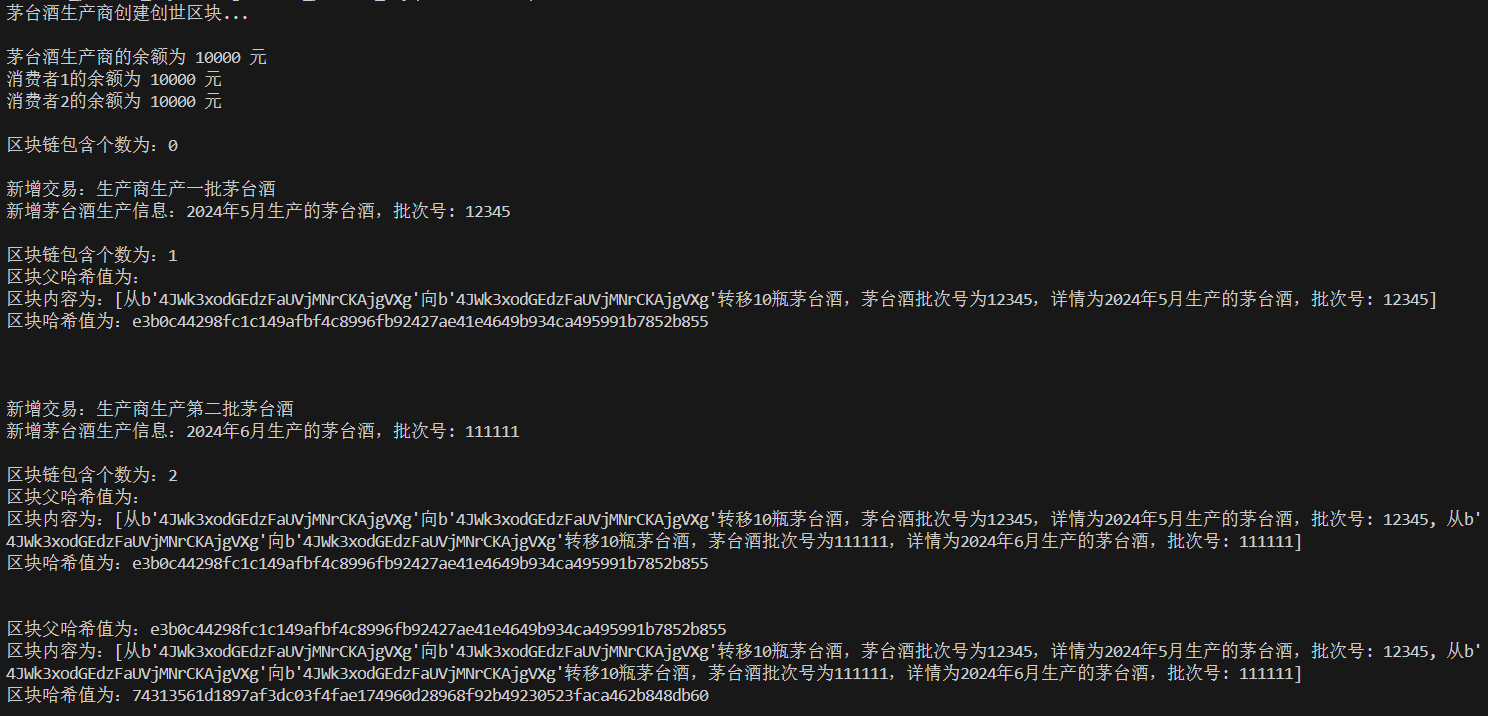
1.本项目添加了三个交易信息放在区块链里面用于模拟查询。

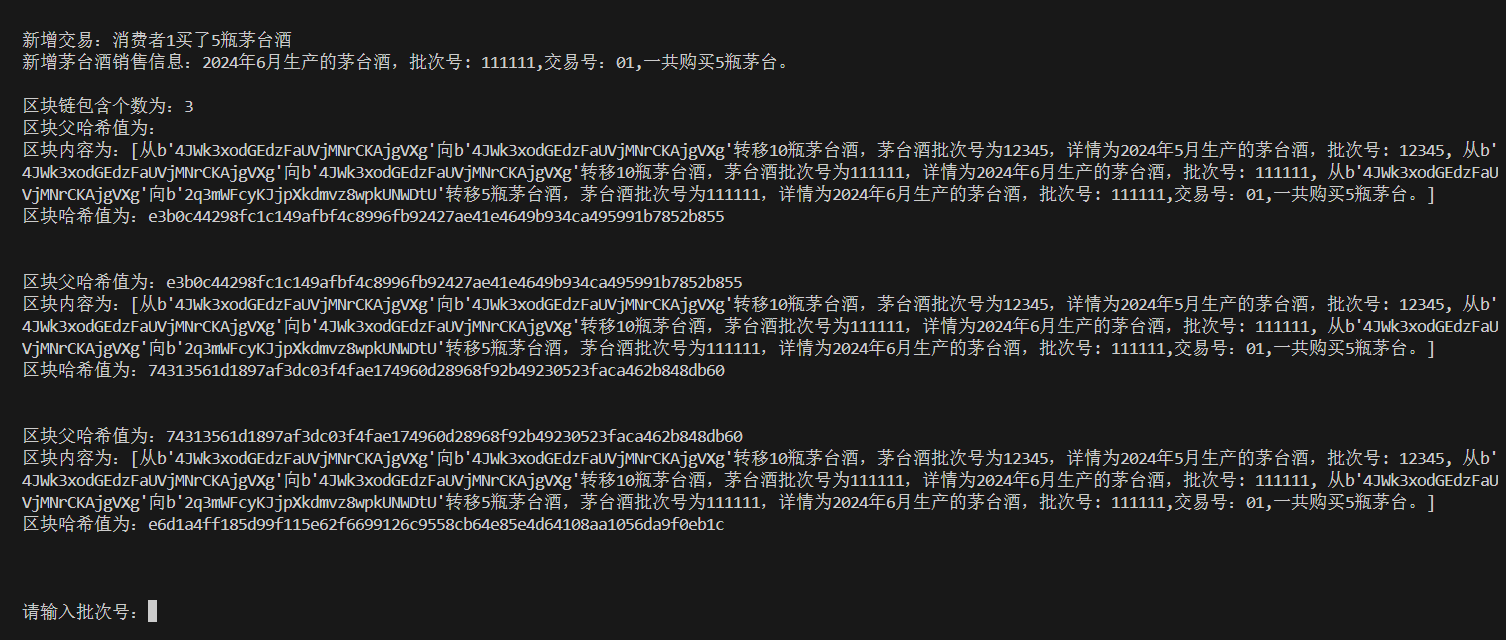
第一个信息：生产商生产一批茅台酒，批次号是12345

第二个信息：生产商生产第二批茅台酒，批次号是111111

第三个信息：消费者1购买了五瓶茅台，批次号是111111，交易号是01

构建区块链如下图所示：





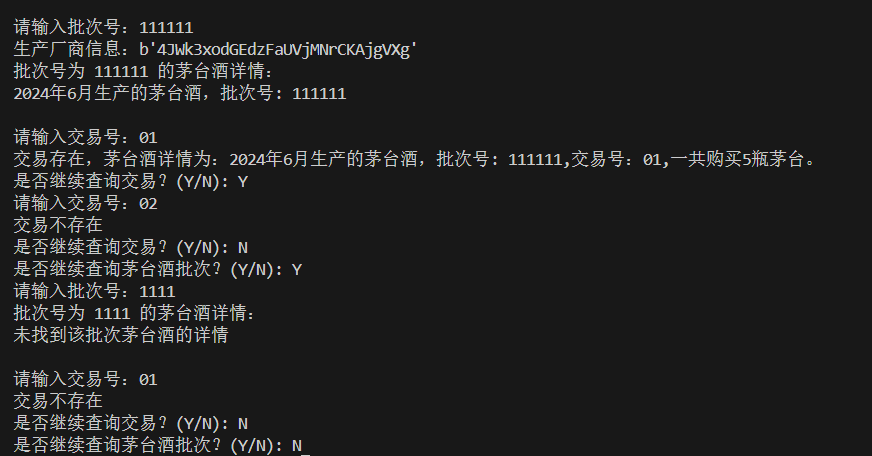
1. 进行用户交互查询

查询流程：

输入批次号：查看茅台酒的基本信息和生产厂商

输入交易号：查看茅台酒的购买信息是否存在

输入后进行是否继续查询的询问



### **项目心得**

在整个项目的开发过程当中，我首先去了解了为什么老师希望用区块链技术完成整个溯源的功能，学习到了区块链的使用知识。然后在选择如何开发整个项目的语言的过程中产生了一些疑问，比如说最开始去查找，希望使用智能合约系统，然后尝试了ganacha，remix等等进行信息构建，但是发现智能合约部署无法部署成功，之后选择了本次上交的实验方法。

在本个项目中，我首先学习了比特币的交易，以及区块链相关的挖矿等等内容，然后再在比特币的交易基础上进行我自己的溯源系统的开发和修改，在整个开发过程当中，区块链的构造是最重要的工作，每一个事件形成自己的哈希值，并且放在区块链里面，形成一个不可以篡改的系统，便于真伪确认。在构造过程中，有茅台的生产信息和买卖信息，在这个过程当中，一旦买卖成功，就会进入区块链，从而可以查询到相应交易存在与否，从而进行真伪确认。