面试问答中英文16-20

面试问答

（中英文，十六至二十题）

基础知识类Basics

16：**What is the difference between view and pure?**

view 和 pure 之间有什么区别?

答：**Both view and pure are modifiers of functions. View can access state variables in a contract and cannot modify them; pure cannot access or modify state variables.**

view和pure都是函数的修饰符，view可以访问合约中的状态变量，不能修改；pure不能访问也不能修改状态变量。

17：**What is the difference between transfer From and safe Transfer From in ERC721?**

ERC721 中的 transfer From 和 safe Transfer From 之间有什么区别?

答：**Both transfer From and safe Transfer From are functions in the ERC721 contract that are used to transfer tokens from one address to another. The difference between them is that the safe Transfer From function checks if the receiving contract implements the on ERC721 Received function and if the value returned by the function is a specific magic number. If the receiving contract does**

not implement the on ERC721 Received function or returns a value that is not a specific magic number, the safe Transfer From function throws an exception and rolls back all changes to ensure that the tokens are not permanently lost. The transfer From function, on the other hand, does not perform this check and therefore may result in the permanent loss of tokens.

Transfer From和safe Transfer From都是ERC721合约中的函数，用于将代币从一个地址转移到另一个地址。它们之间的区别在于，safe Transfer From函数会检查接收合约是否实现了on ERC721 Received函数，并且该函数返回的值是否为特定的魔数。如果接收合约没有实现on ERC721 Received函数或者返回的值不是特定的魔数，safe Transfer From函数将抛出异常并回滚所有更改，以确保代币不会永久丢失。而transfer From函数则不会进行这种检查，因此可能会导致代币永久丢失。

18：**How to convert ERC1155 tokens to non-homogenized tokens?**

如何将 ERC1155 代币转换为非同质化代币？

答：**ERC1155 tokens are fungible and non-fungible tokens that can be used to manage multiple assets in the same contract. ERC721 tokens are non-fungible tokens and each token is unique. Therefore, converting ERC1155 tokens to ERC721 tokens requires converting each ERC1155 token to a unique ERC721 token.**

To convert ERC1155 tokens to ERC721 tokens, you need to perform the following steps:

1. Create a new ERC721 contract.

2. Create a new ERC721 token for each ERC1155 token.

3. Transfer ownership of the ERC1155 tokens to the new ERC721 tokens.

4. Destroy the original ERC1155 tokens.

ERC1155代币是一种可替代和不可替代的代币，可以在同一合约中管理多个资产。ERC721代币是一种非可替代代币，每个代币都是唯一的。因此，将ERC1155代币转换为ERC721代币需要将每个ERC1155代币转换为一个唯一的ERC721代币。

要将ERC1155代币转换为ERC721代币，您需要执行以下步骤：

1.创建一个新的ERC721合约。

2.为每个ERC1155代币创建一个新的ERC721代币。

3.将ERC1155代币的所有权转移到新的ERC721代币。

4.销毁原始ERC1155代币。

19：**What is the purpose of a modifier?**

修饰符（modifier）的作用是什么？

答：**A modifier is a function modifier used to declare a function modifier. The role of function modifiers is very similar to that of Spring's faceted functionality. When it is applied to a function, the logic in the modifier can be pre-executed before or after the function is executed (depending on the specific implementation) to enhance its functionality. The use of modifier can be extracted out of some common operations to improve coding efficiency and reduce code coupling.**

Modifiers can be used to control the access rights and behavior of functions. For example, if you want only the owner of a contract to have access to a function, you can mark the function as private and use a modifier to check if the caller is the owner of the contract. If the caller is not the owner of the contract, the function will stop executing and

roll back all changes. This can help protect the contract from unauthorized access and attacks.

modifier是一种函数修饰符，用于声明一个函数修改器。函数修改器的作用与Spring中的切面功能很相似，当它作用于一个函数上，可以在函数执行前或后（依赖于具体实现）预先执行modifier中的逻辑，以增强其功能。使用modifier可以将一些通用的操作提取出来，提高编码效率，降低代码耦合性。

modifier可以用于控制函数的访问权限和行为。例如，如果您希望只有合约的所有者才能访问某个函数，您可以将该函数标记为private，并使用modifier来检查调用者是否为合约的所有者。如果调用者不是合约的所有者，则函数将停止执行并回滚所有更改。这可以帮助保护合约免受未经授权的访问和攻击

20：**What is the maximum value that uint256 can store?**

uint256 可以存储的最大值是多少?

答：**Maximum 2^256-1. Why minus 1? 0 takes up a space, and the 256-bit representation of the number is from 000……00 to 111……11. For example, if we have uint2, then 2^2 can store 00 = 0, 01 = 1, 10 = 2, 11 = 3,**

最大2^256-1 。为什么减1？0占用了一个空间，256位的表示的数是从000……00到111……11 。例如：如果有uint2 , 那么2^2可存储的数可以是 00 = 0 ， 01 = 1, 10 = 2, 11 = 3，这里最大是3，也就是2^-1,而不是4。