基础知识类26-32

面试问答题（中英文）

基础知识类（26-32）

26、冷读（cold read）和热读（warm read）之间有什么区别?

What is the difference between cold read and warm read?

答:

冷读（cold read）和热读（warm read）是两种不同的访问存储变量的方式。冷读是指第一次读取存储变量时，需要从存储中读取变量的值，这需要较高的gas费用。而热读是指在第一次读取存储变量之后，再次读取存储变量时，可以从缓存中读取变量的值，这需要较低的gas费用。热读和冷读是由Ethereum虚拟机（EVM）自动处理的。

Cold read and warm read are two different ways of accessing stored variables. Cold read means that the first time you read a stored variable, you need to read the value of the variable from the storage, which requires a higher gas cost. Whereas, warm read means that after reading the storage variable for the first time, when reading the storage variable again, the value of the variable can be read from the cache, which requires a lower gas cost. Warm and cold reads are handled automatically by the Ethereum Virtual Machine (EVM).

27、AMM 如何定价资产?

How does AMM price assets?

答:

通过恒定乘积算法，a * b = k ,兑换 m个a 需要的b数量算法 = k/(a+m) - b ,这里没计算手续费。

Through the constant product algorithm, a * b = k ,the number of b's needed to convert m a's algorithmically = k/(a+m) - b ,no commission is calculated here.

28、接口中有效的函数修饰符有哪些?

What are the valid function modifiers in an interface?

答：

在Solidity中，接口是一种抽象合约，它定义了合约应该实现的函数。接口中的函数没有实现，只有函数签名。函数签名包括函数名称、参数类型和返回类型。接口中的函数可以使用以下修饰符：

external：指定函数只能从合约外部调用。

view：指定函数不会修改合约状态。

pure：指定函数既不会修改合约状态，也不会读取合约状态。

payable：指定函数可以接受以太币作为支付。

In Solidity, an interface is an abstract contract that defines the functions that the contract should implement. Functions in an interface have no implementation, only a function signature. The function signature consists of the function name, parameter type and return type. Functions in interfaces can use the following modifiers:

external: specifies that the function can only be called from outside the contract.

view: Specifies that the function does not modify the state of the contract.

pure: specifies that the function will neither modify nor read the contract state.

payable: Specifies that the function can accept Ether as payment.

29、根据 Solidity 风格指南，函数应该如何排序?

How should functions be sorted according to the Solidity style guide?

答：

函数应根据其可见性和顺序进行分组：

构造函数

receive 函数（如果存在）

fallback 函数（如果存在）

外部函数(external)

公共函数(public)

内部函数(internal)

私有函数(private)

在一个分组中，把 view 和 pure 函数放在最后。

Functions should be grouped according to their visibility and order:

constructor

receive function (if it exists)

fallback function (if present)

external function (if present)

public function (public)

internal function (internal)

private function (private)

In a grouping, put the view and pure functions last.

30、根据 Solidity 风格指南，函数修饰符应该如何排序?

How should function modifiers be ordered according to the Solidity style guide?

答:

函数修改器的顺序应该是:

可见性（Visibility）

可变性（Mutability）

虚拟（Virtual）

重载（Override）

自定义修改器（Custom modifiers）

The order of function modifiers should be.

Visibility

Mutability

Virtual

Override

Custom modifiers (Custom modifiers)

31、Solidity 提供哪些关键字来测量时间?

What keywords does Solidity provide to measure time?

答:

now：返回当前区块的时间戳（以秒为单位）。

时间单位：Solidity提供了几个时间单位，包括seconds、minutes、hours、days、weeks和years。这些单位可以与数字一起使用，例如5 minutes或1 hours。

block.timestamp：与now关键字类似，返回当前区块的时间戳。

block.number：返回当前区块的块号。

now: returns the timestamp (in seconds) of the current block.

time units: Solidity provides several units of time, including seconds, minutes, hours, days, weeks and years. These units can be used with numbers, such as 5 minutes or 1 hour.

block.timestamp: similar to the now keyword, returns the timestamp of the current block.

block.number: Returns the block number of the current block.

32、多大 uint 可以与一个地址在一个槽中?

How big a uint can be in a slot with an address?

答：

一个地址是20个字节（160位），而一个uint256是32个字节（256位）, 256-160 = 96。uint96及以下的都可以。

An address is 20 bytes (160 bits) and a uint256 is 32 bytes (256 bits), 256-160 = 96. uint96 and below are fine.

户用私钥签名信息证明身份，应用于Sovrin、uPort。

2.数字证书和区块链

原理是认证机构颁发数字证书，哈希值记录在区块链上。通过区块链确认证书真实性，应用于Blockcerts。

3.零知识证明（ZKP）

原理是用户生成零知识证明，证明身份属性。验证者验证证明有效性而不获取具体身份信息，应用于Zcash、zk-SNARKs。

4.代币和访问控制

原理是使用代币或智能合约控制资源访问。用户通过身份验证后获得代币，使用代币访问资源，应用于ERC-725、ERC-735。

1.Decentralized Identity (DID)

The principle is that the user generates a public-private key pair to create a unique DID, which is registered on blockchain. Users use the private key to sign information to prove their identity, applied to Sovrin, uPort.

2.Digital certificate and blockchain

The principle is that the certification authority issues a digital certificate and the hash value is recorded on blockchain. The authenticity of the certificate is confirmed through the blockchain, which is used in Blockcerts.

3.Zero Knowledge Proof (ZKP)

The principle is that the user generates zero-knowledge proof to prove identity attributes. The verifier verifies the validity of the proof without obtaining specific identity information, applied to Zcash, zk-SNARKs.

4.Token and Access Control

The principle is to use tokens or smart contracts to control access to resources. Users obtain tokens after authentication and use them to access resources, applied in ERC-725, ERC-735.