Solidity

Sanson Sebastiano

2022

Contents

1 Layout del file sorgente **2**

2 Struttura di un contratto **2**

3 Units and Globally Available Variables **2**

4 Expressions and Control Structures **3**

1 Layout del file sorgente

- **SPDX-License-Identifier:** necessario per indicare la licenza del codice sorgente, in quanto per questioni di fiducia meglio se pubblico, del smart contract. Se non si vuole specificare la licenza, usare il valore speciale *UNLICENSED*.
- **pragma solidity >= x.y.z:** indica la versione di Solidity che si sta utilizzando.
  pragma è una keyword che abilita certe features del compilatore.
- **import "filename":** importa un file esterno.

2 Struttura di un contratto

- **State variables:** sono le variabili che vengono salvate nella blockchain. Qui ci sono tutti i tipi di variabili di stato.
- **Functions:** sono le funzioni che vengono eseguite dal contratto. Possono essere definite come esterne o interne al contratto e possono avere differenti livelli di visibilità.
- **Function modifiers:** ?
- **Events:** sono interfacce convenienti con le funzionalità di registrazione EVM.
- **Errors:** permette di definire errori personalizzati, attraverso un nome e parametri.
  Revert statements ?
- **Struct type**
- **Enum type**

3 Units and Globally Available Variables

- **Ether unit:**
  - *wei* = 1
  - *gwei* = 10e9
  - *ether* = 10e18
- **Time units:** secondi, minuti, ore, giorni, settimane, anni.

- **Special Variables and Functions:**
  - **block:** contiene informazioni su un blocco.
  - **msg:** contiene informazioni su un messaggio.
  - **tx:** contiene informazioni su una transazione.
  - **abi:** contiene funzioni per la codifica e la decodifica di valori ABI.
  - **type:** contiene informazioni sul tipo di un contratto.
  - **blockhash:** restituisce il blocco hash di un blocco precedente.
  - **gasleft:** restituisce la quantità di gas rimanente.
  - **now:** restituisce il timestamp corrente (secondi dal 1970).
  - **addmod:** restituisce (x + y)
  - **mulmod:** restituisce (x * y)
  - **keccak256:** restituisce il risultato di una funzione di hash di 256 bit.
  - **sha256:** restituisce il risultato di una funzione di hash di 256 bit.
  - **ripemd160:** restituisce il risultato di una funzione di hash di 160 bit.
  - **ecrecover:** recupera l'indirizzo dell'account che ha firmato un messaggio.

4 Expressions and Control Structures

- **Control Structures:** supporta *if, else, while, do-while, for, break, continue, return, throw, revert, require, assert, try/catch.*
- **Function calls:**
  - **Internal function calls:** sono le chiamate di funzioni interne al contratto.
    Solo le funzioni di uno stessa istanza di un contratto possono essere chiamate internamente.
    Sconsigliata la ricorsione.
  - **External function calls:** sono le chiamate di funzioni esterne al contratto.
    Sono chiamate del tipo *otherContract.functionName()* e *this.functionName()*.
    Nota: una chiamata esterna non crea una propria transazione, ma è parte di una transazione complessiva.

    È possibile specificare un ammontare di *Wei* o *gas* con la chiamata.
    Nota: è consigliato non specificare il valore del gas esplicitamente in quanto i costi del gas potrebbero cambiare nel tempo.

- **Function Calls with Named Parameters:** il nome dei parametri possono essere specificati in qualsiasi ordine se sono racchiusi tra {}.
  Una chiamata a questa funzione, la lista dei parametri può essere in ordine differente ma i nomi devono coincidere con quelli specificati nella dichiarazione di tale funzione.
- **Omitted Names in Function Definitions:** è possibile omettere i nomi dei parametri e del valore di ritorno in una dichiarazione di funzione.
  Saranno comunque presenti nello *stack* ma saranno inacessibili dai nomi.
- **Creating Contracts via *new*:** un contratto può creare un altro contratto con la keyword *new*.