Libera Università di Bolzano– Facoltà di Ingegneria

Esame Financial Trading and Algorithms

# Regole

* + Non è permessa alcuna comunicazione con altre persone o tra studenti. I telefoni e ogni altro mezzo di comunicazione devono essere tenuti spenti. Aprire un programma di comunicazione non è permesso ed è considerato truffa.
  + Sei responsabile per il salvataggio corretto dei tuoi files.

## Esercizio Smart Contract

## Idea

Il tuo compito è creare un contratto per gestire le certificazioni. Lo studente deve presentare una richiesta di iscrizione e pagare una commissione. Dopo aver superato un esame, l'autorità di certificazione (che è il deployer del contratto) inserisce i dati della certificazione: nome, livello, data. Chiunque può fornire un indirizzo, nome e livello e ottenere le informazioni se quella persona ha almeno quel livello di certificazione e quando è stata presa. L'utente ha sempre il diritto di cancellare la propria certificazione, pagando la commissione.

## Compito

Scrivi il contratto Solidity **certification**.

## Strutture dati suggerite

    mapping(address=>uint) private application\_time;

    mapping(address=>string) private certification\_name;

    mapping(address=>uint) private certification\_time;

    mapping(address=>uint) private certification\_level;

    address payable public authority;

    uint public fee;

## Passo 1

Scrivi un costruttore che imposti l'autorità e la commissione.

## Passo 2

Aggiungi la funzione **enrol**. L'utente deve pagare almeno la commissione. La funzione imposta il tempo attuale come tempo di richiesta. Non è necessario specificare il proprio nome né il livello richiesto.

**GIRA PAGINA 🡪**

## Passo 3

Aggiungi una funzione **insert**, utilizzabile solo dall'autorità. L'autorità passa come parametro un indirizzo, un nome e un livello. Se quell'indirizzo ha già una certificazione più alta o se il tempo di richiesta è più vecchio di 90 giorni, la funzione restituisce **falso**. In caso contrario, la funzione aggiorna i dati, reimposta il tempo di richiesta e restituisce **vero**.

## Passo 4

Aggiungi una view **check** che prende come parametri un indirizzo, un nome e un livello. Se quell'indirizzo corrisponde al nome fornito e ha un livello di certificazione almeno pari a quel livello, restituisce il momento della certificazione. In caso contrario restituisce 0.

## Passo 5

Aggiungi la funzione **cancel** tramite la quale un utente cancella tutti i suoi dati (le variabili stringa vengono impostate a ""). La commissione deve essere pagata affinché la funzione funzioni.

## Passo 6

Carica il contratto sulla blockchain di Sepolia e verifica tutte le funzioni utilizzando il tuo Account 1 per l'autorità e Account 2 per lo studente.

Se non sei riuscito a farlo completamente, dopo aver copiato tutto in un file di testo, rimuovi le funzioni problematiche, carica e verifica ciò che hai fatto correttamente

## Salva e restituisci

* un file di testo **certification.txt** contenente il codice sorgente Solidity che hai costruito
* l'indirizzo del tuo contratto sulla blockchain Sepolia.