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 i biglietti degli autobus. Il contratto consente a qualsiasi persona di acquistare un biglietto con una durata predefinita e tiene un credito nel caso in cui l'utente paghi di più. Qualsiasi utente può verificare quanti secondi di validità rimangono al suo biglietto e il controllore può verificare se il biglietto dell'utente è ancora valido o meno.

## Bus ticket

Scrivi il contratto Solidity **busticket**.

## Strutture dati suggerite

    mapping (address => uint) public trip\_start;

    mapping (address => uint) public credit;

    address public controller;

    uint public ticket\_price;

    uint public duration;

## Passo 1

Crea un **costruttore** che imposti chi è il controllore, il prezzo del biglietto e la durata del biglietto in secondi.

## Passo 2

Aggiungi una funzione **buy\_ticket** che controlla se è stato pagato abbastanza denaro e imposta l'inizio del viaggio per quell'indirizzo. Nel caso in cui l'utente abbia pagato più del necessario, il credito per quell'utente viene aumentato. La funzione restituisce l'importo del credito dell'utente.

## Passo 3

Aggiungi una view **remaining\_time**, che consente all'utente di sapere quanti secondi di validità ha il suo ultimo biglietto.

**GIRA PAGINA 🡪**

## Passo 4

Aggiungi una view **check**, che consente al controllore di verificare se un indirizzo ha un biglietto valido. La funzione restituisce un bool.

## Passo 5

Aggiungi una funzione **get\_credit**, che consente a qualsiasi utente di recuperare il suo credito. Non dimenticare di azzerare il credito.

## Passo 6

Fai deploy del contratto sulla blockchain di Sepolia e verifica il funzionamento di buy\_ticket e get\_credit.

## Passo 7

Aggiungi una funzione **buy\_ticket2**, simile a buy\_ticket ma che consente all'utente di pagare e contemporaneamente utilizzare il suo credito per il biglietto. Ad esempio, se il biglietto costa 10 wei e l'utente ha 3 wei di credito, può pagare solo 8 wei e ottenere il biglietto e 1 wei come credito.

**Salva e restituisci**

un file di testo **bus.txt** contenente il codice sorgente Solidity che hai creato e l'indirizzo del tuo contratto Sepolia.