Homework 4

VolcanoCoin contract

At each point where you change your contract you should re deploy to the JavascriptVM and test your changes.

1. In Remix, create a new file called `VolcanoCoin.sol`.

2. Define the pragma compiler version to `^0.8.0`.

3. Before the pragma version, add a license identifer `// SPDX-License-Identifier: UNLICENSED`.

4. Create a contract called VolcanoCoin.

5. Create a variable to hold the total supply of 10000.

6. Make a public function that returns the total supply.

7. Make a public function that can increase the total supply. Inside the function, add 1000 to the current total supply.

8. We probably want users to be aware of functions in the contract for greater transparency, but to make them all public will create some security risks (e.g. we don't want users to be able to change the total supply).

   Declare an `address` variable called `owner`.

9. Next, create a `modifier` which only allows an owner to execute certain functions.

10. Make your change total supply function `public`, but add your modifier so that only the owner can execute it.

11. The contract owner's address should only be updateable in one place. Create a constructor and within the constructor, store the owner's address.

12. It would be useful to broadcast a change in the total supply. Create an event that emits the new value whenever the total supply changes. When the supply changes, emit this event.

13. In order to keep track of user balances, we need to associate a user's address with the balance that they have.
a) What is the best data structure to hold this association ?
b) Using your choice of data structure, set up a variable called `balances` to keep track of the number of volcano coins that a user has.

14. We want to allow the balances variable to be read from the contract, there are 2 ways to do this.
What are those ways ?
Use one of the ways to make your balances variable visible to users of the contract.

15. Now change the constructor, to give all of the total supply to the owner of the contract.

16. Now add a public function called transfer to allow a user to transfer their tokens to another address. This function should have 2 parameters :
    - the amount to transfer and
    - the recipient address.

Why do we not need the sender's address here ?
What would be the implication of having the sender's address as a parameter ?

17. Add an `event` to the transfer function to indicate that a transfer has taken place, it should log the amount and the recipient address.

18. We want to keep a record for each user's transfers. Create a `struct` called Payment that stores the transfer amount and the recipient's address.

19. We want to have a payments array for each user sending the payment. Create a `mapping` which returns an array of Payment structs when given this user's address.

Resources

Official Solidity Documentation
Globally Available Variables