ERC20 Exercise 1

Intro

In this exercise, you get an ERC20 token contract that uses the ERC20 component by OpenZeppelin. Your goal is to implement minting and burning functionality for the token and to write tests to ensure that all the token functionality works as expected.

Accounts

- 123 - Deployer & Owner
- 1 - Alice
- 2 - Bob

Tasks

Contract Implementation

In the contract `src/exercises/erc20/erc20.cairo`, implement the following functionality:

1. Create mint() and burn() functions inside the interface `IERC20MintingAndBurning`
2. Implement that interface in the `ERC20` contract
    - Implement an external `mint()` function
    - Implement an external `burn()` function
3. Enforce that only the deployer of the token can call the `mint()` function
    - Implement an internal function to check that the caller is the owner
4. Enforce that only the user can burn his own tokens

Tests

For every test, add assertions to make sure the token amounts are correct, use the ERC20 `balance_of` function

1. Implement a `mint_tokens()` helper function that mints tokens to a user, it receives:

    - Token contract address
    - Token contract dispatcher
    - Token deployer address
    - Account to mint the token to
    - Amount of tokens to mint

2. Complete the `test_mint_and_burn` test:

    - Mint 100 tokens to Alice
    - Mint 200 tokens to Bob
    - Burn 50 tokens from Alice

3. Complete the `test_transfer` test:

    - Transfer 50 tokens from Alice to Bob

4. Complete the `test_approve_and_transfer_from` test:

    - Approve Bob to spend 50 tokens on behalf of Alice
    - Use the transfer_from function to transfer 50 tokens from Alice to Bob, "spending the allowance"

Useful links

Foundry testing

Openzeppelin ERC20 implementation in Cairo