Essay – Lab 2 – Florin-George Bărcan

The second lesson introduces 2 very important data types: *mapping* and *address*.

Regarding *addresses*, a blockchain, in our case, a Ethereum blockchain is composed of accounts with each account have a balance of Ether, i.e., a currency. Each account has a unique identifier and that is the address.

Mappings are a key-value data structure for storing and accessing data.

We also get to know *msg.sender*, which is a global variable that points to the address of the person who calls the current function. With this variable, together with the *require* keyword, offers us some degree of security in our smart-contracts.

In this lesson we also found out we can create contracts that inherit other contracts, thus splitting our code into a much more organized code. A good example would be:

contract Animal {

function feed() public returns (string memory) {

return “animal finished eating”;

}

}

contract Cat is Animal {

function feedCat() public returns (string memory) {

return “cat finished eating”;

}

}

Here we have 2 contracts instead of one big contract. The contract Cat inherits the contract Animal and can have a different behavior from the contract that it inherits.

Solidity has 2 ways in which it can store variables. From the first lesson we learned about *memory* which stores the variables temporary. This time we learn about *storage*. This keyword indicates that the variable is stored permanently inside the blockchain.

Finally, we get to learn how to interact with other blockchains that we don’t own, like the CryptoKitties blockchain. The communication takes place using an interface, i.e., defining the contract without the content between the curly braces.