**SYNTAX CƠ BẢN TRONG NGÔN NGỮ LẬP TRÌNH SOLIDITY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Biên soạn:** Nhóm Nghiên Cứu Blockchain Khoa HTTT

--------------------------------------------------------------------------------------------------------------

1. **Kiến trúc của một source file solidity[1]**

* Kiến trúc một đoạn chương trình đơn giản như sau:

|  |
| --- |
| pragma solidity >=0.4.0 <0.6.0;  contract SimpleStorage {  uint storedData;  function set(uint x) public {  storedData = x;  }  function get() public view returns (uint) {  return storedData;  }  } |

* Cấu trúc một source file solidity luôn bắt đầu bằng khai báo pragma để khai báo phiên bản thực thi.
  + Lấy ví dụ 1 pragma solidity >=0.4.0 <0.6.0; //Phiên bản đang sử dụng từ 0.4.0 đến <0.6.0
  + Lấy ví dụ 2 pragma solidity ^0.4.0; //Phiên bản sử dụng cao nhất là 0.4.0 và 0.5.0 trở lên thì không hoạt động được.
  + Bài tập ví dụ về pragma như sau:
    - Cho pragma solidity ^0.5.0 chương trình compiler 0.5.2 có chạy được không? \_\_\_ [Y/N]
    - Cho pragma solidity >0.5.0 <0.6.0 chương trình compiler 0.5.0 có chạy được hay không? \_\_\_ [Y/N]
* Phần tiếp theo là phần contract giống class trong ngôn ngữ lập trình hướng đối tượng. Một contract là tập hợp đoạn mã như là hàm và dữ liệu và nó có một địa chỉ cụ thể trong Ethereumblockchain.
  + Khai báo Contract: contract <Tên\_contract> {}
* Phần tiếp theo uint storedData; là phần khai báo biến toàn cục trong contract về kiểu dữ liệu sẽ được bàn kỹ trong phần Type Variables
* Phần function set(uint x) public {...} là hàm một hàm không có kiểu dữ liệu trả về.
* Phần function get() public view returns (uint) {... return ...} là phần hàm có kiểu dữ liệu trả về là kiểu uint.
* Định nghĩa một hàm trong Solidity như sau:

function <tên\_hàm>(tham số 1, tham số 2, ...) [các hậu tố phía sau] //Sẽ được nói rõ trong phần hàm .

* Trong Solidity có rất nhiều keywords qua từng tài liệu sẽ cập nhật các keywords và giải thích về nó qua ký hiệu ❂

1. **Import Contract[2]**

* Trong ngôn ngữ Solidity cũng có thể import một file giống như trong ngôn ngữ JavaScript.
* Cách import một file theo cấu trúc như sau: import “filename”;
* Một số ví dụ import thông dụng:
  + Import từ npm:

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

hay

import "@openzeppelin/contracts@4.2.0/token/ERC20/ERC20.sol";

* + Import từ github:

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v2.5.0/contracts/math/SafeMath.sol";

* + Import Local File:

import "./myLovelyLovelyLib.sol";

❂ Keyword: **import** 🡪 Dùng để import file source từ một địa chỉ vào source hiện hành.

* Thực hành import một contract bằng 3 phương pháp nêu trên.

1. **Chương trình ứng dụng đầu tiên[3]**

* Cho đoạn chương trình sau đây

|  |
| --- |
| **File Name:** 01\_SolidityTest.sol |
| pragma solidity ^0.5;  contract SolidityTest {  function getResult() public view returns(uint){  uint a = 1;  uint b = 2;  uint result = a + b;  return result;  }  } |

* Kết quả mong muốn chạy của chương trình: 3

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

1. **Cách comment trong ngôn ngữ Solidity[4]**

* Trong ngôn ngữ lập trình Solidity có 2 cách comment:
  + **Cách 1:** Comment trên 1 dòng sử dụng **//đây là comment**
  + **Cách 2:** Comment nhiều hơn 1 dòng sử dụng

**/\***

**\* đây là comment**

**\*/**

1. **Tài liệu tham khảo**

[1] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity - Basic Syntax”. [Online] [Thời gian truy cập: 28/04/2022].

[2] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity - Basic Syntax”. [Online] [Thời gian truy cập: 28/04/2022].

[3] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity - First Application”. [Online] [Thời gian truy cập: 28/04/2022].

[4] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity - Comments”. [Online] [Thời gian truy cập: 28/04/2022].