Biên soạn: Nhóm Nghiên Cứu Blockchain Khoa HTTT

---

1. Thư viện trong Solidity

- Trong Solidity không trợ một số hàm có sẵn bắt buộc phải có những thư viện tự viết để giải quyết vấn đề.
- Ví dụ: Hàm lấy Index từ một giá trị trong Solidity không có sẵn để giải quyết chúng ta sử dụng thư viện **`library`** trong solidity
- Library đặt TRƯỚC các contract.
- Khai báo thư viện trong Contract

```
Library <LibraryName> {
//Implement
}
```

Ví dụ về **Library Search:** Lấy Index của một giá trị cho trước.

```
library Search {
   function indexOf(uint[] storage self, uint value) internal view returns (uint) {
      for (uint i = 0; i < self.length; i++)
      {
        if (self[i] == value)
        return i;
      }

        return uint(int(-1));
   }
}
```

- Rule: Function trong library phải dạng internal.
- Sử dụng Library:
  - using <LibraryName> for <DataType>;
  - Ví dụ về cách sử dụng Library

```solidity
/**
 * Using Test Library
 */
contract Test {
   using Search for uint[];
   uint[] public data;

   constructor() {
      data.push(1);
      data.push(2);
      data.push(3);
      data.push(4);
      data.push(5);
   }

   function isValuePresent() external view returns(uint){
      uint value = 4;

      //Now data is representing the Library
      uint index = data.indexOf(value);
      return index;
   }
}
```

2. Sự kiện (event) trong Solidity

- Sự kiện được khai báo bằng từ khoá: **event**
- Sử dụng dùng để giao tiếp giữa blockchain và ứng dụng front-end (Phần tương tác với smart contract).
- Khai báo sự kiện và sử dụng sự kiện:
  - `event <EventName> (<ListParam>) //Khai báo sự kiện`
  - `emit <EventName> (<ListParam>) //Gọi Event`
- EventName của event và emit phải CÙNG TÊN và ListParam có số lượng bằng nhau và cùng kiểu dữ liệu.
- Ví dụ về một contract chuyển tokenBalance

```solidity
contract EventSolidity {

    // Mapping Address to Balance
    mapping(address => uint) public tokenBalance;

    /**
     * Constructor Token Balance
     */
    constructor() public {
        tokenBalance[msg.sender] = 100;
    }

    /**
     * Send Token From To
     */
    function sendToken(address _to, uint _amount) public returns(bool) {
        require(tokenBalance[msg.sender] > _amount, "Not Enough Money");
        assert(tokenBalance[_to] + _amount >= tokenBalance[_to]);
        assert(tokenBalance[msg.sender] - _amount <= tokenBalance[msg.sender]);
        tokenBalance[msg.sender]-=_amount;
        tokenBalance[_to]+=_amount;
        return true;
    }
```

- Thực hiện viết Event cho contract EventSolidity.
    - Định nghĩa Event Soldity

```
// Mapping Address to Balance
mapping(address => uint) public tokenBalance;

event TokenSent(address _from, address _to, uint _amount);
/**
 * Constructor Token Balance
 */
constructor() public {
    tokenBalance[msg.sender] = 100;
}
```

    - Sử dụng event:

```
/**
 * Send Token From To
 */
function sendToken(address _to, uint _amount) public returns(bool) {
    require(tokenBalance[msg.sender] > _amount, "Not Enough Money");
    assert(tokenBalance[_to] + _amount >= tokenBalance[_to]);
    assert(tokenBalance[msg.sender] - _amount <= tokenBalance[msg.sender]);

    tokenBalance[msg.sender]-=_amount;
    tokenBalance[_to]+=_amount;

    emit TokenSent(msg.sender, _to, _amount);

    return true;
}
```

    - Mục tiêu để viết Event để thực hiện giao tiếp ứng dụng front end sau này.

3. **Error Handling**

- **require(bool condition, “message”):** Nếu điều kiện đúng thực hiện lệnh tiếp theo và nếu điều kiện sai trả về trạng thái ban đầu. Thường dùng để check những input đầu vào từ ứng dụng. Tham số “message” là tham số được thêm vô nhằm thông báo lỗi đang xảy ra. (Kiểm tra lỗi external)
- **assert(bool condition):** Dùng để kiểm tra các lỗi bên trong (internal)
- **revert(<message>):** Trả về tất cả state trước đó đã thực hiện. Nếu có message thì thông báo trả về message.
    - Thao khảo ví dụ: 09_RevertExample.sol

4. Tài liệu tham khảo

[1] https://www.tutorialspoint.com/solidity/solidity_libraries.htm, [Online] [Thời gian truy cập: 23/06/2022]

[2] https://www.tutorialspoint.com/solidity/solidity_events.htm, [Online] [Thời gian truy cập: 23/06/2022]

[3] https://www.tutorialspoint.com/solidity/solidity_error_handling.htm, [Online] [Thời gian truy cập: 23/06/2022]