

솔리디티 언어 실습(리믹스)

(helloworld, counter컨트랙트)

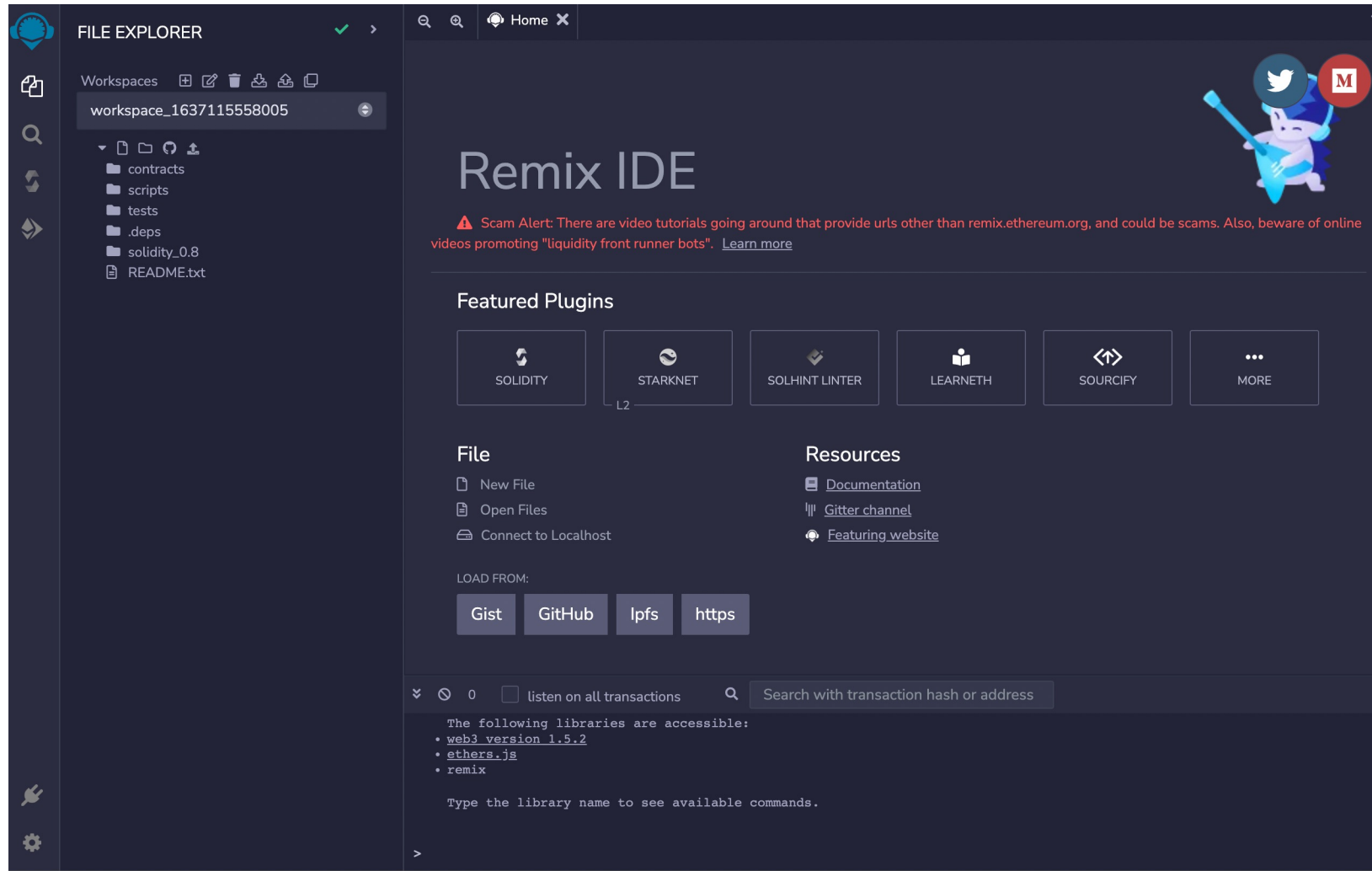
김동환

dannykim@kakao.com

<https://kimsfamily.kr/page/Profile>

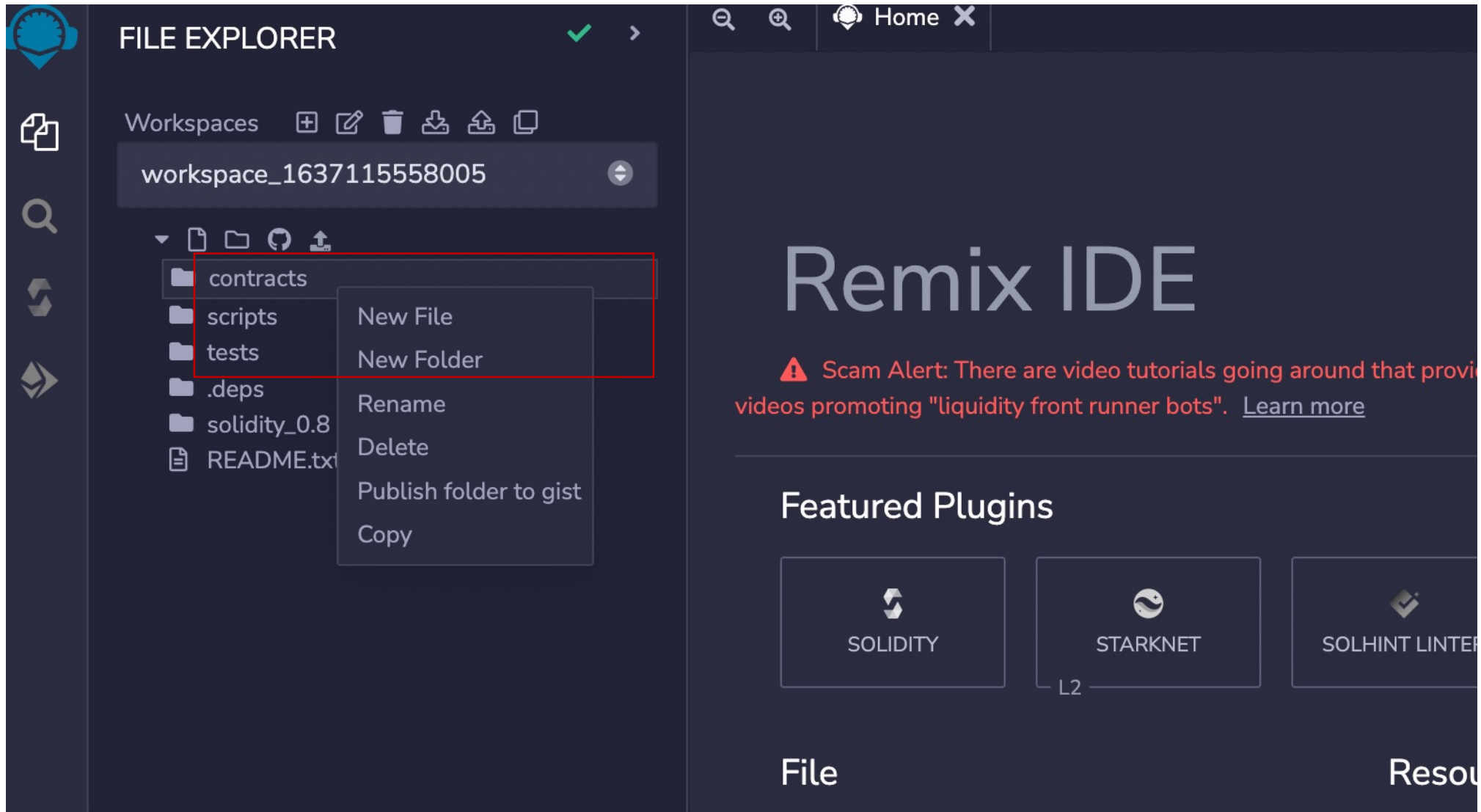
리믹스 IDE – 사이트 접속

<http://remix.ethereum.org/>



리믹스 IDE – 파일 생성 및 저장

폴더 선택 후 – 마우스 우클릭 – New File



Hello World

```
1 // SPDX-License-Identifier: MIT
2 // compiler version must be greater than or equal to 0.8.13 and less than 0.9.0
3 pragma solidity ^0.8.13;
4
5 contract HelloWorld {
6     string public greet = "Hello World!";
7 }
```

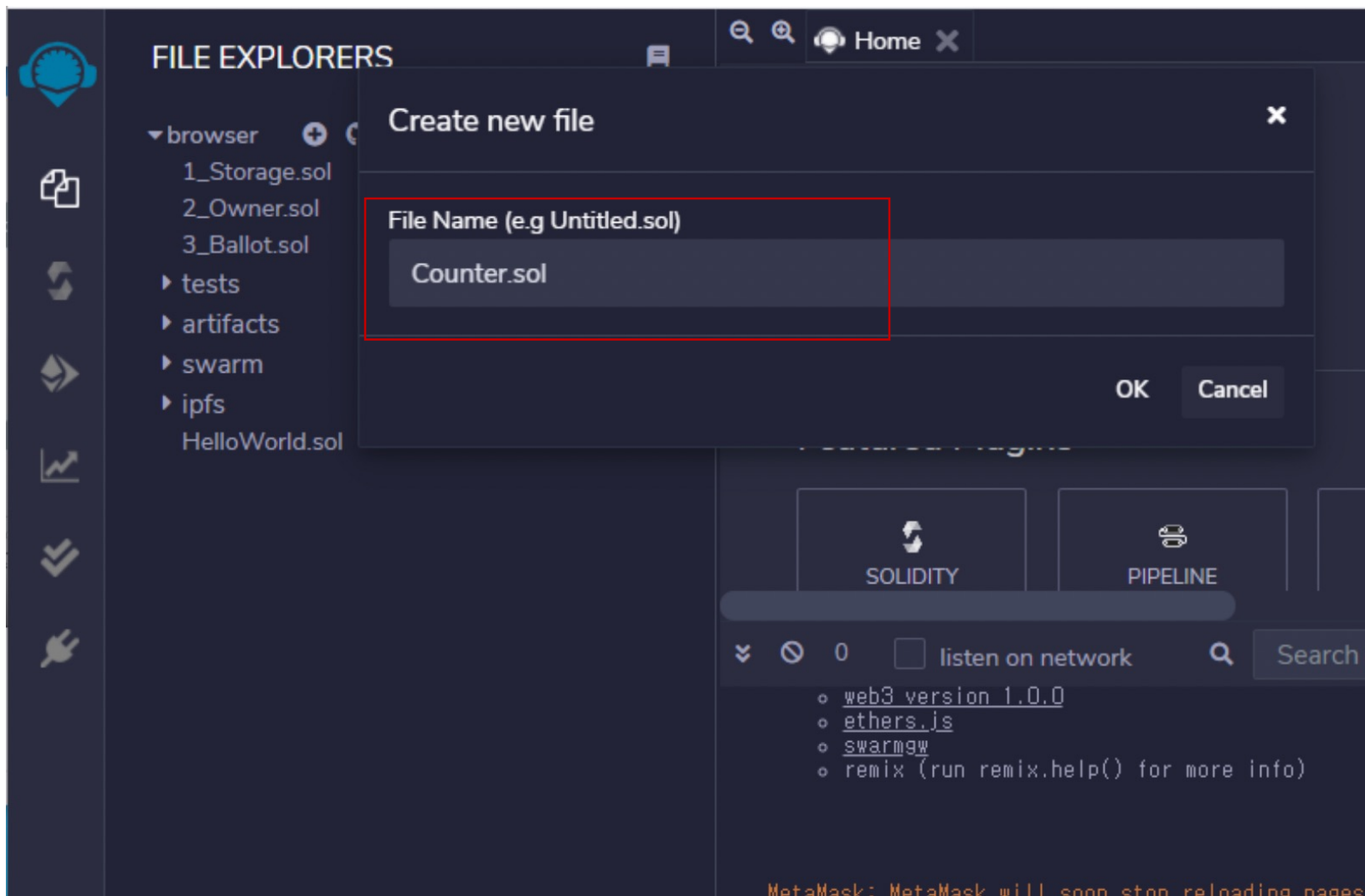
- 연습문제

자신의 이름의 컨트랙트를 만들고, 자신에게 인사하는 문자를 출력해보세요!(블록체인에 기록해보세요!)


- 컨트랙트 이름 : 예) DannyWorld
- 메시지 : "hi Dannyworld! Nice to meet you!!"

Counter 컨트랙트

Counter.sol 로 저장 후 코드 작성



Counter 컨트랙트



SOLIDITY COMPILER

COMPILER +


0.8.0+commit.c7dfd78e



☐ Include nightly builds



☐ Auto compile

☐ Hide warnings

Advanced Configurations

 Compile Counter.sol

Compile and Run script  

 Counter.sol

```
1 pragma solidity ^0.8.0;
2
3 contract Counter{
4
5 }
```

Counter.sol

```
1  // SPDX-License-Identifier: GPL-3.0
2  pragma solidity ^0.8.0;
3
4  contract Counter{
5      uint count;
6      function getCount() public view returns(uint){
7          return count;
8      }
9      function incrementCount() public{
10         count = count + 1;
11     }
12 }
```

Counter 컨트랙트 - 컴파일

컴파일 후, ABI, ByteCode 확인

The screenshot shows the Solidity Compiler interface. On the left, the 'SOLIDITY COMPILER' section is active, displaying the version '0.8.0+commit.c7dfd78e' and options for 'Include nightly builds', 'Auto compile', and 'Hide warnings'. A 'Compile Counter.sol' button is prominent. Below it, the 'CONTRACT' dropdown shows 'Counter (Counter.sol)'. Further down are buttons for 'Publish on Ipfs', 'Publish on Swarm', and 'Compilation Details'. At the bottom left, there are links for 'ABI' and 'Bytecode'. The main editor area shows the Solidity code for 'Counter.sol' with line numbers 1 through 12. The code includes a license header, a pragma statement, and two functions: 'getCount()' and 'incrementCount()'. The bottom status bar shows a transaction hash and the function being called: 'Counter.getCount()'.

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.0;
3
4 contract Counter{
5     uint count;
6     function getCount() public view returns(uint){
7         return count;
8     }
9     function incrementCount() public{
10         count = count + 1;
11     }
12 }
```

This screenshot shows a menu of options available after compilation. It includes a 'CONTRACT' dropdown set to 'Counter (Counter.sol)'. Below this are three large buttons: 'Publish on Swarm' with a Swarm icon, 'Publish on Ipfs' with an IPFS icon, and 'Compilation Details'. At the bottom, a red rectangular box highlights two buttons: 'ABI' and 'Bytecode', each with a document icon.

ABI 및 ByteCode 확인

ABI

```
[ { "inputs": [], "name": "getCount", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function" }, { "inputs": [], "name": "incrementCount", "outputs": [], "stateMutability": "nonpayable", "type": "function" } ]
```

ByteCode

```
"608060405234801561001057600080fd5b50610172806100206000396000f3fe608060405234801561001057600080fd5b50600436106100365760003560e01c8063a87d942c1461003b578063e5071b8e14610059575b600080fd5b610043610063565b6040516100509190610092565b60405180910390f35b61006161006c565b005b60008054905090565b600160005461007b91906100ad565b600081905550565b61008c81610103565b82525050565b60006020820190506100a76000830184610083565b92915050565b60006100b882610103565b91506100c383610103565b9250827fffffffffffffffffffffffffffffffffffffffff038211156100f8576100f761010d565b5b828201905092915050565b6000819050919050565b7f4e487b71000000000000000000000000000000000000000000000000000000000600052601160045260246000fdfea26469706673582212209b884dcc54e5fd09bf4c2a353fe8847890fb19bce37163f1f98e02a6bc0e083d64736f6c63430008000033"
```

배포(Deploy), 컨트랙트 주소 확인

The screenshot displays the Remix IDE interface during the deployment of a Solidity contract. The left sidebar contains the 'DEPLOY & RUN TRANSACTIONS' panel, which is currently active. It shows the following configuration:

- VM**: Selected as the environment.
- ACCOUNT**: Set to '0x5B3...eddC4 (99.99999)'.
- GAS LIMIT**: Set to '3000000'.
- VALUE**: Set to '0' with the unit 'Wei'.
- CONTRACT**: Set to 'Counter - contracts/Counter.sol'.
- Deploy**: A prominent orange button to initiate the deployment.
- Publish to IPFS**: An unchecked checkbox.
- OR**: A separator between deployment methods.
- At Address**: A button to load a contract from a specific address.
- Transactions recorded**: A section showing 3 recorded transactions.
- Deployed Contracts**: A section showing the deployed contract 'COUNTER AT 0xF8E...9FBE8 (ME)'.

The main editor area displays the Solidity code for 'Counter.sol' and '2_Owner.sol'. The 'Counter.sol' code is as follows:

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.0;
3
4 contract Counter{
5     uint count;
6     function getCount() public view returns(uint){
7         return count;
8     }
9     function incrementCount() public{
10         count = count + 1;
11     }
12 }
```

The bottom right panel shows the transaction details for the deployment:

- creation of Counter pending...**
- [vm] from: 0x5B3...eddC4 to: Counter.(constructor)**
- value: 0 wei data: 0x608...00033 logs: 0 hash: 0x6c2...4c7b0**
- Debug**: A button to debug the transaction.

트랜잭션 결과 확인(함수실행 테스트)

Deployed Contracts

▼ COUNTER AT 0XF8E...9FBE8 (ME)

incrementCount

getCount

0: uint256: 1

Low level interactions

CALLDATA

Transact

0 ☐ listen on all transactions

Search with transaction hash or address

CAL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

L to: Counter.getCount() data: 0xa87...d942c

transact to Counter.incrementCount pending ...

✓ [vm] from: 0x5B3...eddC4

to: Counter.incrementCount() 0xf8e...9fBe8 value: 0 wei

data: 0xe50...71b8e logs: 0 hash: 0xe16...fc4ae

call to Counter.getCount

CAL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

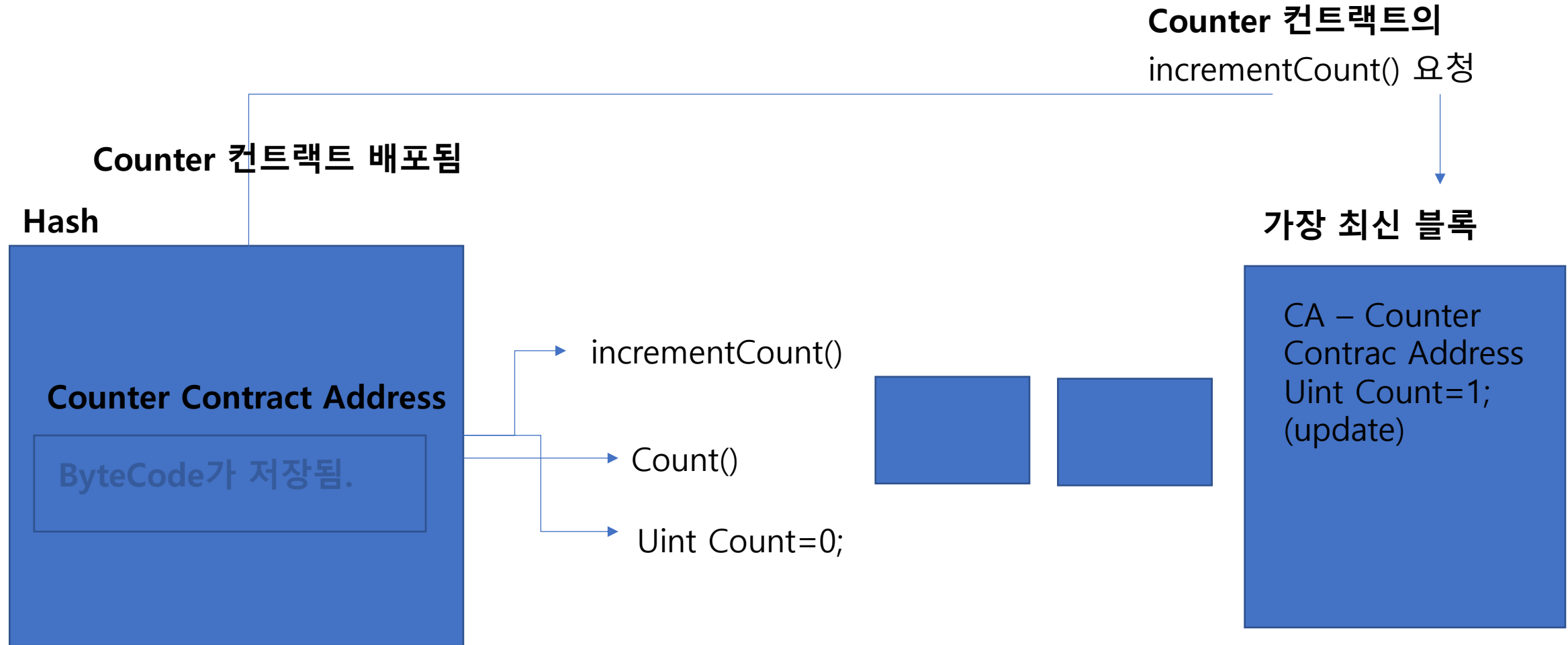
L to: Counter.getCount() data: 0xa87...d942c

Debug

Debug

Debug

트랜잭션 결과 확인(함수실행 테스트)



코드 리팩토링 후 재테스트

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.0;
3
4 contract Counter{
5     uint public count;
6     function incrementCount() public{
7         count++;
8     }
9 }
```