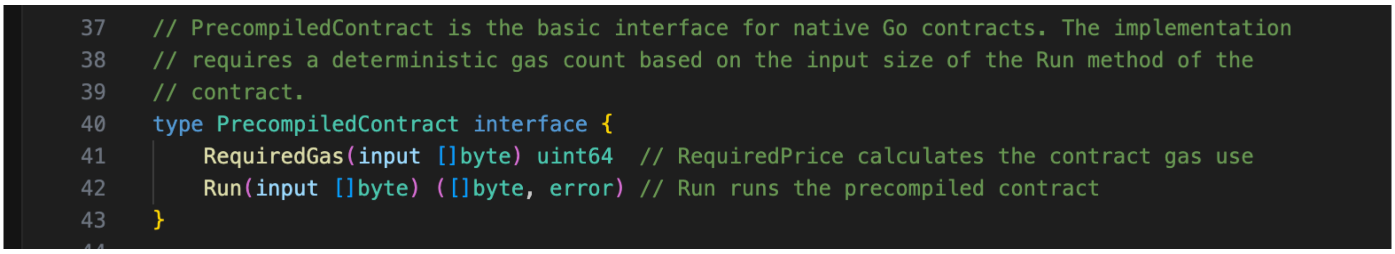
预编译合约编写规范

- 预编译合约接口实现定义：



```go
// PrecompiledContract is the basic interface for native Go contracts. The implementation
// requires a deterministic gas count based on the input size of the Run method of the
// contract.
type PrecompiledContract interface {
    RequiredGas(input []byte) uint64  // RequiredPrice calculates the contract gas use
    Run(input []byte) ([]byte, error) // Run runs the precompiled contract
}
```

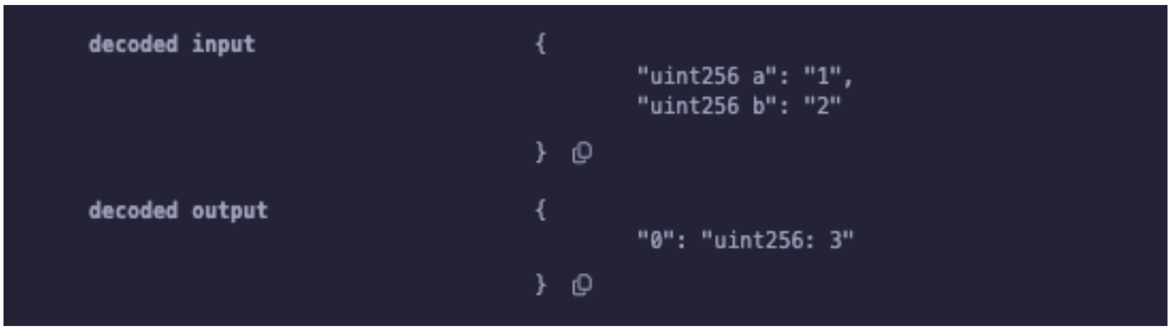
- 预编译合约实现Demo——Add操作

```go
// 构造预编译合约

var errPanguAdd = errors.New("error pangu add : input length must be 64 bytes")

type panguAdd struct{}

// 自定义Gas计算方法
func (p *panguAdd) RequiredGas(input []byte) uint64 {
  // Input为 tx msg 中的 data，如果需要按操作计算Gas，需要自行解析
  return 10
}

// 自定义合约执行逻辑
func (p *panguAdd) Run(input []byte) ([]byte, error) {
  if len(input) != 64 {
    return nil, errPanguAdd
  }

  // 需要对input做解析
  a := new(uint256.Int).SetBytes(input[:32])
  b := new(uint256.Int).SetBytes(input[32:])

  sum := new(uint256.Int).Add(a, b)

  return sum.Bytes(), nil
}

// 分配预编译合约地址
common.BytesToAddress([]byte{20}): &panguAdd{},
```

- solidity调用（需要在节点源码重编译后使用）

```solidity
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.0;

contract AddContract {
    function byteToUint(bytes1 b) public pure returns (uint8) {
        // Convert a single byte to uint8
        return uint8(b);
    }

    function byteArrayToUint(bytes memory b) public pure returns (uint256) {
        require(b.length <= 32, "Bytes array is too long to convert to uint256");

        uint256 result = 0;
        for (uint i = 0; i < b.length; i++) {
            result = result << 8;
            result = result | uint8(b[i]);
        }
        return result;
    }

    function add(uint a, uint b) public view returns (uint) {
        // 预编译合约地址（与底层源码相同）
        address customPrecompileAddress = address(0x14);

        // 构造intput
        bytes memory input = abi.encodePacked(a, b);

        // staticCall调用
        (bool success, bytes memory result) = customPrecompileAddress.staticcall(input);
        require(success, "PanguAdd call failed");

        // Decode the result
        return byteArrayToUint(result);
    }
}
```

- remix调用



```
decoded input                {
                                    "uint256 a": "1",
                                    "uint256 b": "2"
                             }
decoded output               {
                                    "0": "uint256: 3"
                             }
```