

<<Abstract>>
Ownable
cs2/Ownable.sol

Private:
_owner: address

Internal:
_checkOwner()
_transferOwnership(newOwner: address)
Public:
<<event>> OwnershipTransferred(previousOwner: address, newOwner: address)
<<modifier>> onlyOwner()
constructor(initialOwner: address)
owner(): address
renounceOwnership() <<onlyOwner>>
transferOwnership(newOwner: address) <<onlyOwner>>

<<Abstract>>
ERC20
cs2/ERC20.sol

Private:
_balances: mapping(address=>uint256)
_allowances: mapping(address=>mapping(address=>uint256))
_totalSupply: uint256
_name: string
_symbol: string

Internal:
_transfer(from: address, to: address, value: uint256)
_update(from: address, to: address, value: uint256)
_mint(account: address, value: uint256)
_burn(account: address, value: uint256)
_approve(owner: address, spender: address, value: uint256)
_approve(owner: address, spender: address, value: uint256, emitEvent: bool)
_spendAllowance(owner: address, spender: address, value: uint256)
Public:
constructor(name_: string, symbol_: string)
name(): string
symbol(): string
decimals(): uint8
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(to: address, value: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, value: uint256): bool
transferFrom(from: address, to: address, value: uint256): bool

Geth
cs2/Geth.sol

Private:
GETH_TO_WEI: uint64
operator: address

Private:
calculateTokenAmount(weiAmount: uint256): uint256
calculateWeiAmount(tokenAmount: uint256): uint256
External:
<<payable>> purchaseTokens()
purchaseWei(tokenAmount: uint256)
setOperator(newOperator: address) <<onlyOwner>>
destroy() <<onlyOwner>>
Public:
constructor()
decimals(): uint8
allowance(owner: address, spender: address): uint256

