Unified Modeling Language

Graph Theory

Inheritance

Function Hash Signature

Contract Describe

Contract Unit Test Example

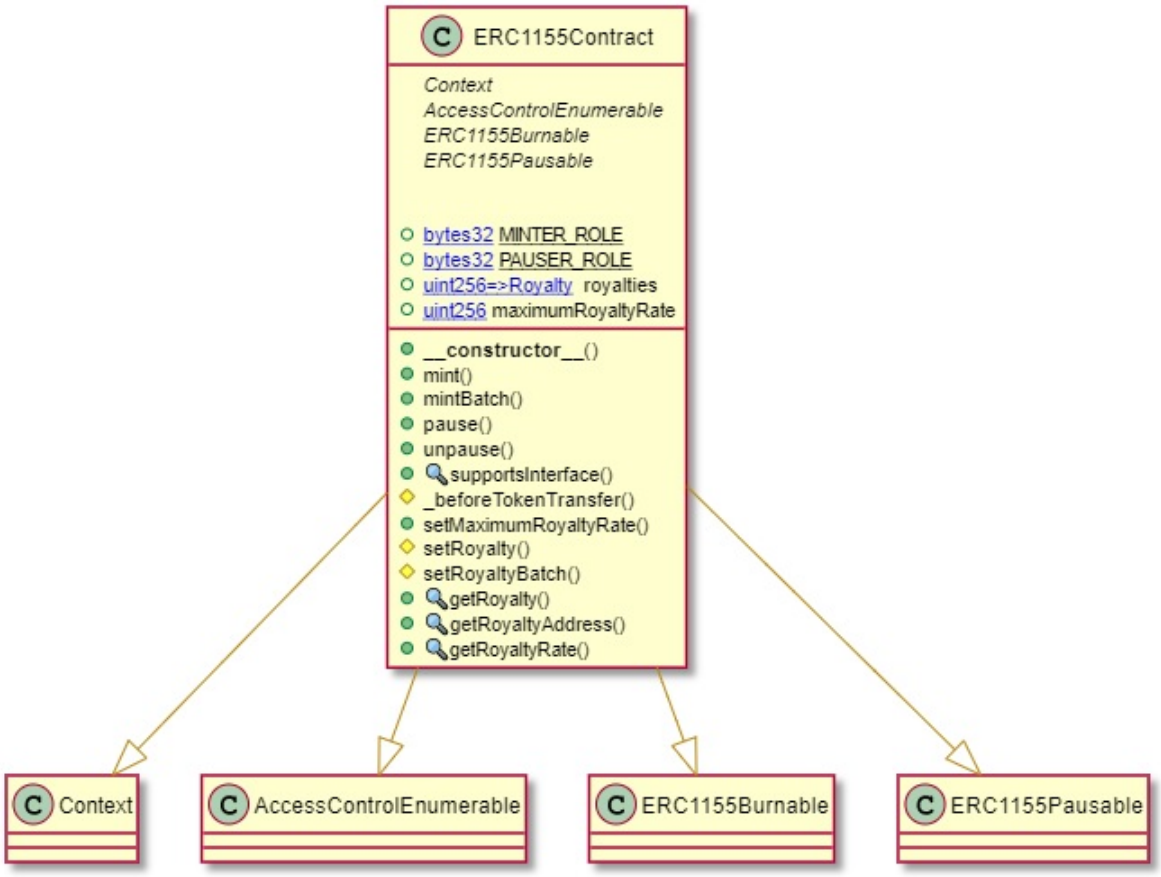
SERCAN ÇELENK

16 AUGUST 2021

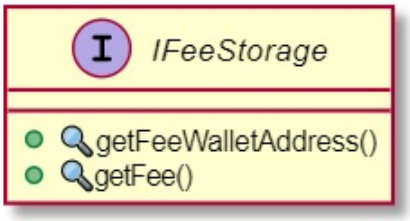
CONTENTS

1 Unified Modeling Language(UML)

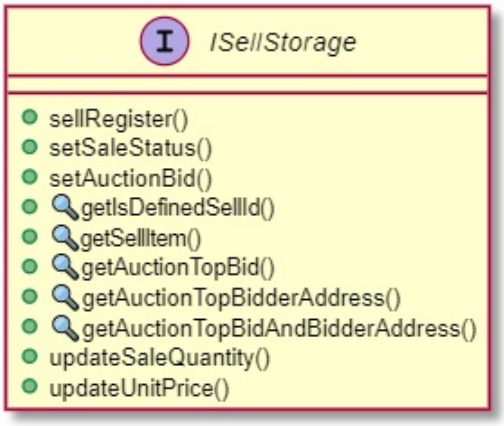
UML is a general-purpose modeling language in software engineering for visualizing the design of a system. It is not a written language. Although classified for different purposes, it is mainly used for modelling.

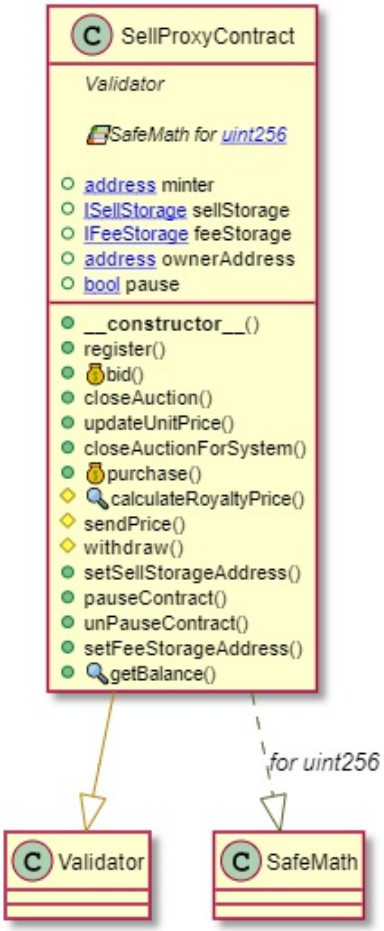


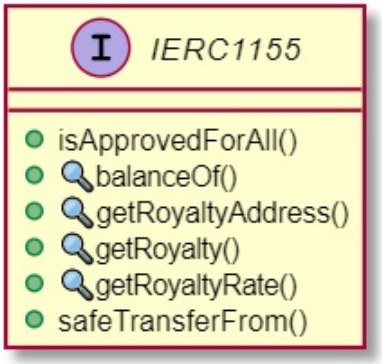

FeeStorageContract
AccessControl
bytes32 WRITER_ROLE
bytes32 READER_ROLE
bytes32 MINTER_ROLE
address minter
address feeWalletAddress
uint256 defaultFeeRate
address=>FeeOverride feeOverrides
__constructor__()
setFeeOverride()
setFeeOverrideBatch()
setWriterRole()
setReaderRole()
getContractAddress()
setFeeWalletAddress()
setDefaultFeeRate()
getFeeWalletAddress()
getFee()
AccessControl

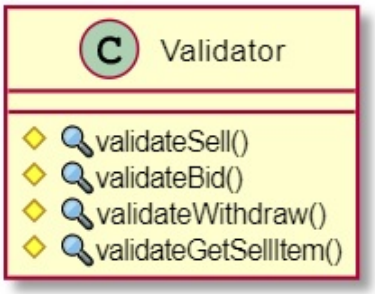

IFeeStorage
getFeeWalletAddress()
getFee()


SellStorageContract
AccessControl
bytes32 WRITER_ROLE
bytes32 READER_ROLE
bytes32 MINTER_ROLE
address minter
address ownerAddress
uint256=>SellLib.Sell sales
uint256=>SellLib.Bid bids
uint256=>SellLib.Metadata metadataList
__constructor__()
sellRegister()
setSaleItem()
setWriterRole()
setReaderRole()
getSellItem()
getContractAddress()
getIsDefinedSellId()
updateUnitPrice()
setSaleStatus()
updateSaleQuantity()
setAuctionBid()
getAuctionTopBid()
getAuctionTopBidderAddress()
getAuctionTopBidAndBidderAddress()
AccessControl


ISellStorage
sellRegister()
setSaleStatus()
setAuctionBid()
getIsDefinedSellId()
getSellItem()
getAuctionTopBid()
getAuctionTopBidderAddress()
getAuctionTopBidAndBidderAddress()
updateSaleQuantity()
updateUnitPrice()


SellProxyContract
Validator
SafeMath for uint256
address minter
ISellStorage sellStorage
IFeeStorage feeStorage
address ownerAddress
bool pause
__constructor__()
register()
bid()
closeAuction()
updateUnitPrice()
closeAuctionForSystem()
purchase()
calculateRoyaltyPrice()
sendPrice()
withdraw()
setSellStorageAddress()
pauseContract()
unPauseContract()
setFeeStorageAddress()
getBalance()
for uint256
Validator
SafeMath

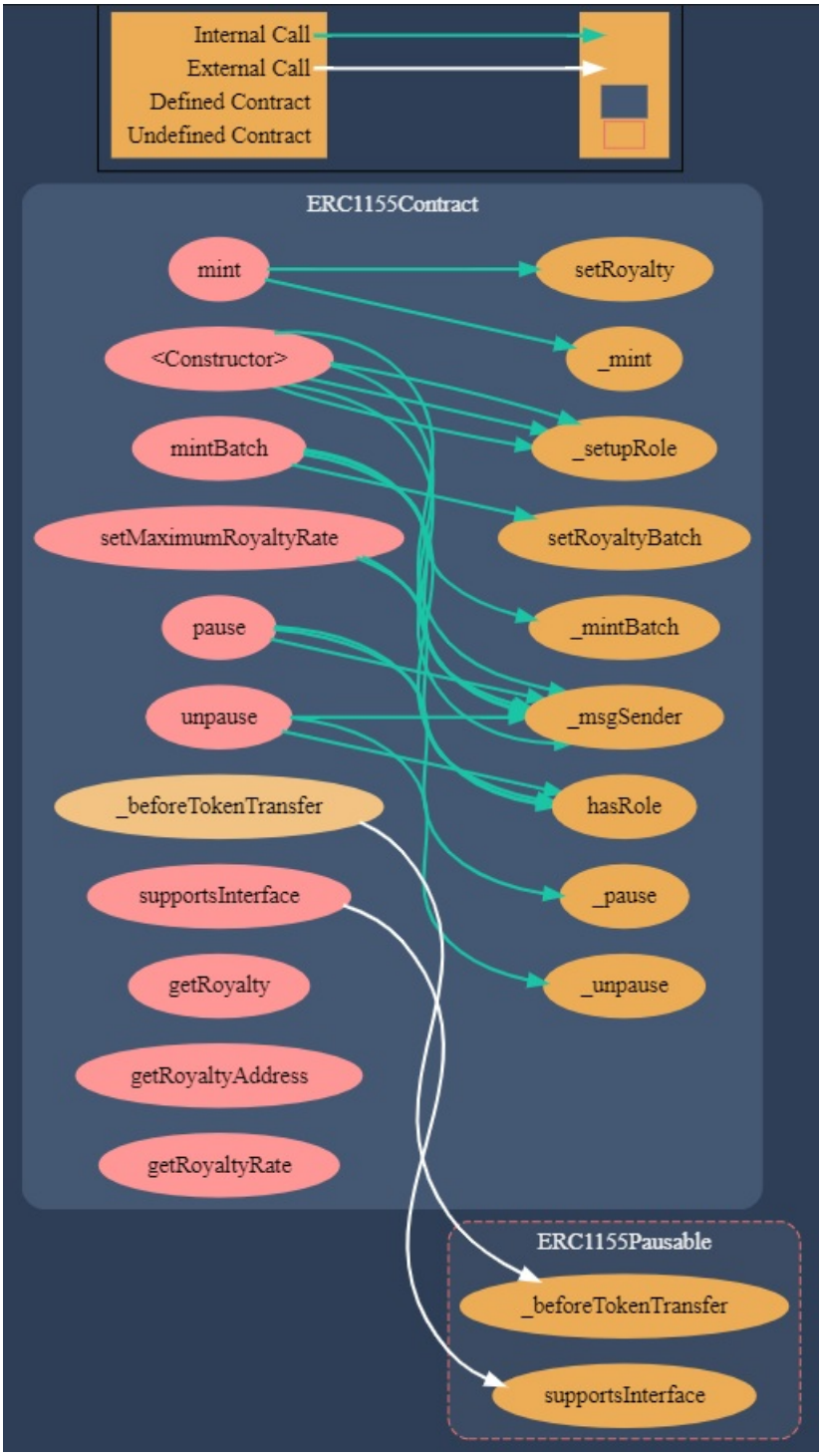

IERC1155
isApprovedForAll()
balanceOf()
getRoyaltyAddress()
getRoyalty()
getRoyaltyRate()
safeTransferFrom()


Validator
validateSell()
validateBid()
validateWithdraw()
validateGetSellItem()

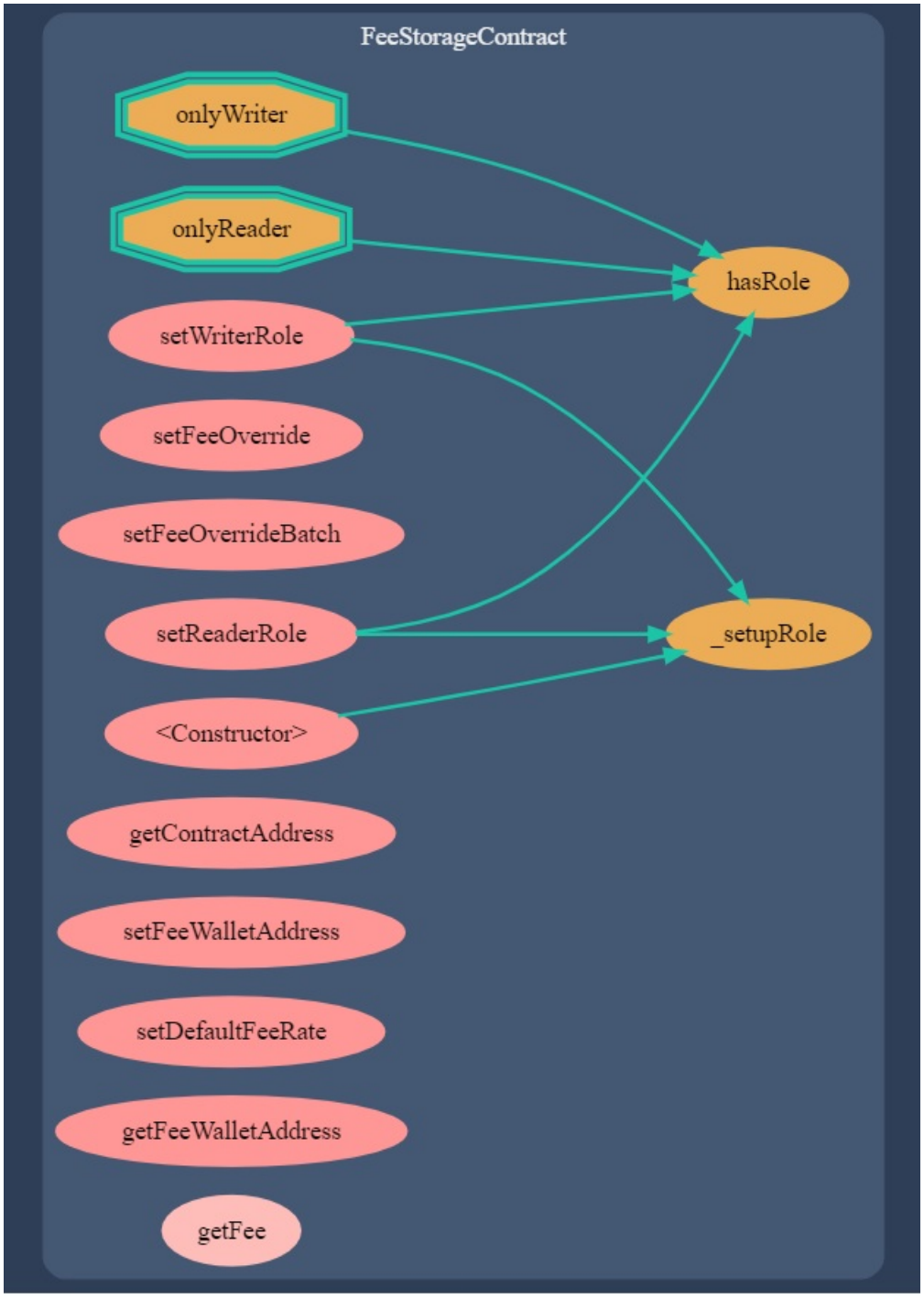
2 Graph Theory

In mathematics, a graph is a structure that determines a set of objects to which pairs of objects are in some sense related.

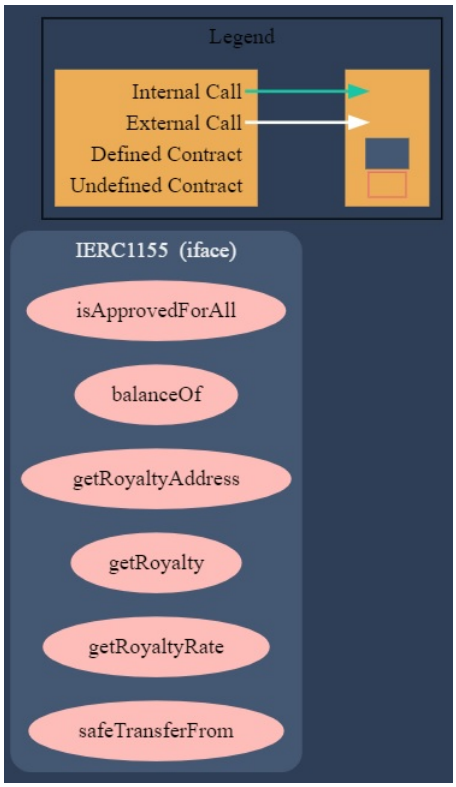
ERC1155Contract



FeeStorage

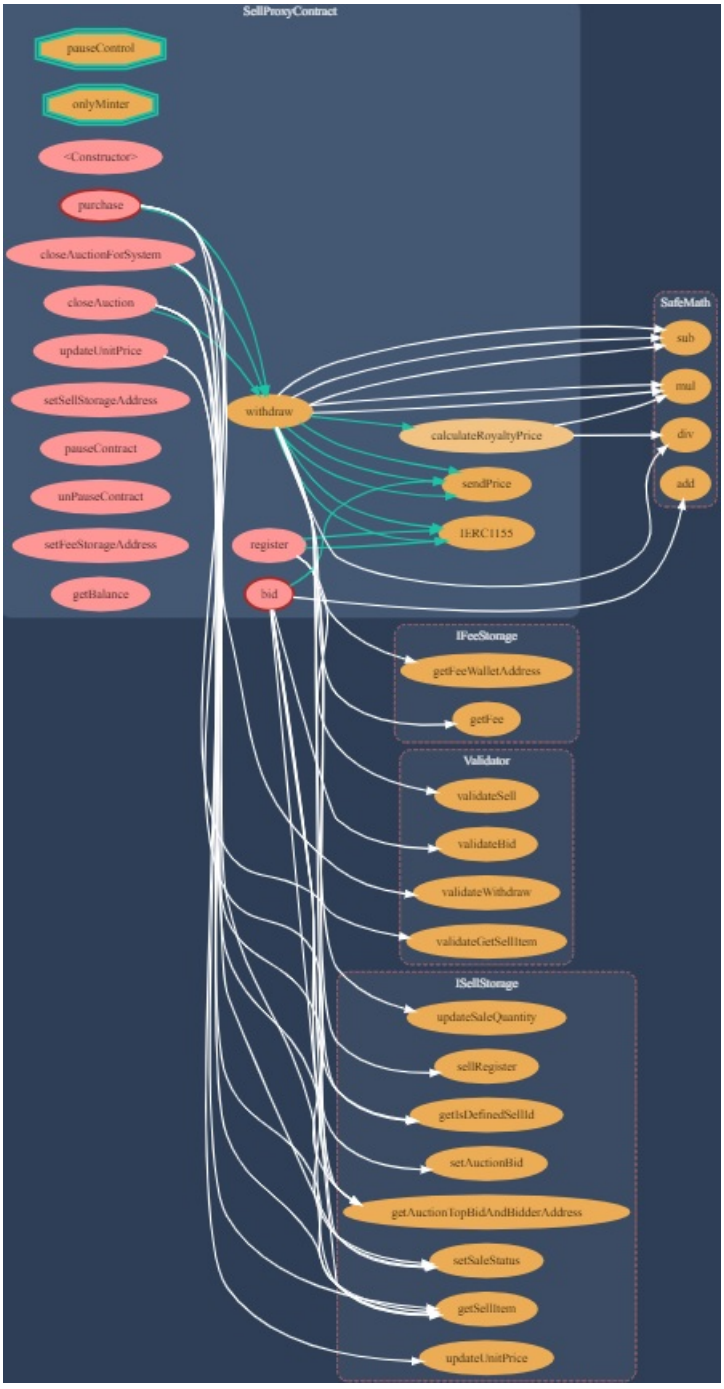

FeeStorageContract
onlyWriter
onlyReader
hasRole
setWriterRole
setFeeOverride
setFeeOverrideBatch
setReaderRole
_setupRole
<Constructor>
getContractAddress
setFeeWalletAddress
setDefaultFeeRate
getFeeWalletAddress
getFee

IERC1155

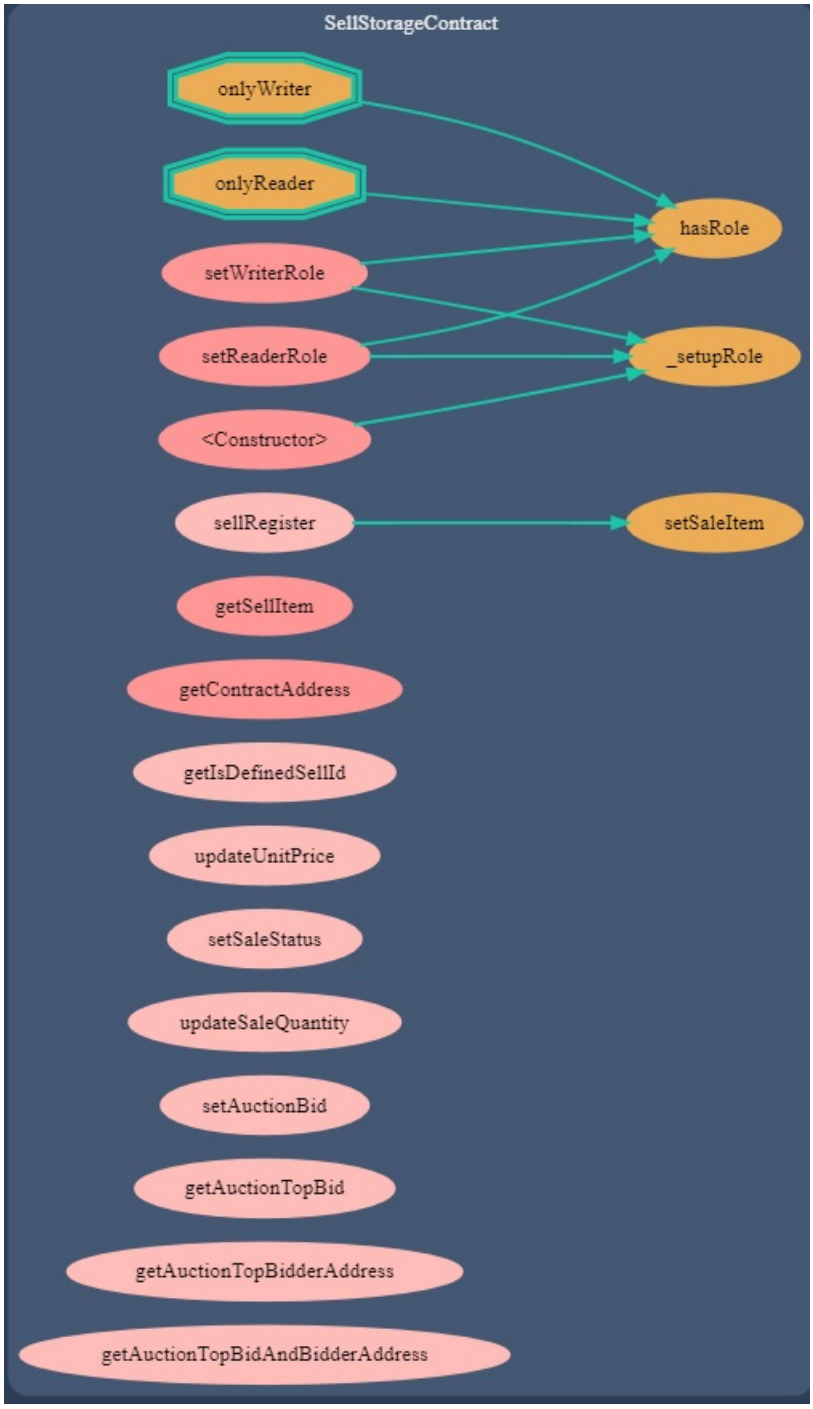


ISellStorage

ISellStorage



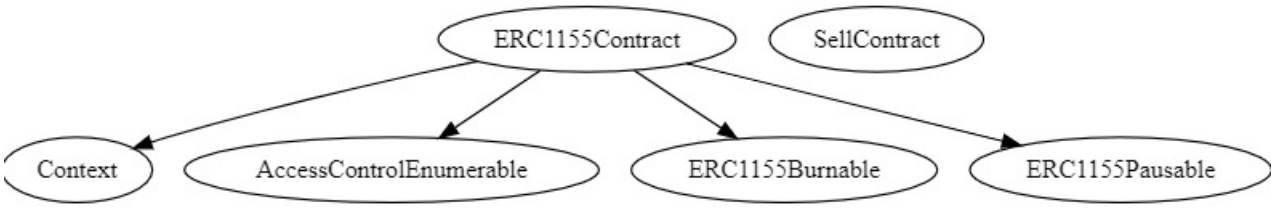
SellContractProxy



3 Inheritance

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important feature of Object Oriented Programming.

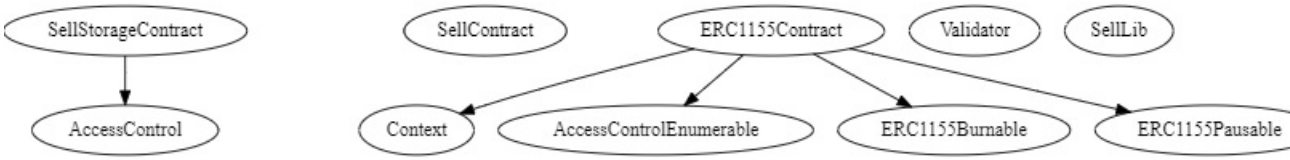
ERC1155



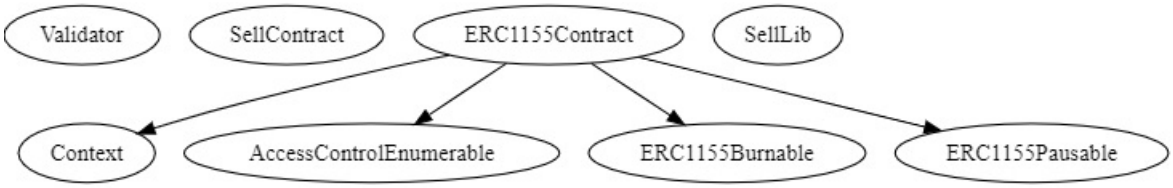
FeeStorage

SellProxyContract

SellStorage



Validator



4 Function Signature Hash

Signhash **====>** Function Signature

ERC1155Contrat.sol

44928394 => getRoyaltyAddress(uint256)

8773ebd7 => mint(address,uint256,uint256,uint256,bytes32,bytes32,uint8,bytes)

b05b1e1c => mintBatch(address,uint256[],uint256[],uint256[],bytes)

8456cb59 => pause()

3f4ba83a => unpause()

01ffc9a7 => supportsInterface(bytes4)

fe49010b => beforeTokenTransfer(address,address,address,uint256[],uint256[],bytes)

df7e849a => setMaximumRoyaltyRate(uint256)

9ed838e7 => setRoyalty(address,uint256,uint256)

4f20b87f => setRoyaltyBatch(address,uint256[],uint256[])

1af9cf49 => getRoyalty(uint256)

1524afc9 => getRoyaltyRate(uint256)

Validator.sol

241df2a8 => validateSell(SellLib.Sell)

e53453bd => validateBid(uint256,address)

1e0e0d6f => validateWithdraw(uint256,address)

0f3b2d36 => validateGetSellItem(SellLib.Sell,address,uint256)

FeeStorage.sol

5b75de36 => setFeeOverride(address,uint256,address)

94c0fe89 => setFeeOverrideBatch(address[],uint256[],address[])

7e82d8ae => setWriterRole(address)

1ed87da9 => setReaderRole(address)

32a2c5d0 => getContractAddress()

31503ec4 => setFeeWalletAddress(address)

5f70fdb5 => setDefaultFeeRate(uint256)

ec397b29 => getFeeWalletAddress()

b88c9148 => getFee(address)

SellStorage.sol a59de20b => sellRegister(SellLib.Sell)

f8e5cedb => setSaleItem(SellLib.Sell)

7e82d8ae => setWriterRole(address)

1ed87da9 => setReaderRole(address)

37a8a018 => getSellItem(uint256)

32a2c5d0 => getContractAddress()

0528ba60 => getIsDefinedSellId(uint256)

bac28c40 => updateUnitPrice(uint256,uint256)

b54c5c31 => setSaleStatus(uint256,bool)

0be799b5 => updateSaleQuantity(uint256,uint256)

266970c1 => setAuctionBid(uint256,address,uint256)

4689afec => getAuctionTopBid(uint256)

be59e51d => getAuctionTopBidderAddress(uint256)

d59365ca => getAuctionTopBidAndBidderAddress(uint256)

SellProxyContract.sol

58333316 => setFeeStorageAddress(IFeeStorage)

88af4af6 => register(SellLib.Sell)

9f04996d => bid(uint256,address)

236ed8f3 => closeAuction(uint256)

bac28c40 => updateUnitPrice(uint256,uint256)

35a8cdc8 => closeAuctionForSystem(uint256[])

ea3bd5df => purchase(uint256,uint256,address)

8b40f86e => calculateRoyaltyPrice(uint256,uint256)

16f64c9e => sendPrice(address,uint256)

c3fef492 => withdraw(SellLib.Sell,address,uint256,uint256,uint256)

57541a75 => setSellStorageAddress(ISellStorage)

439766ce => pauseContract()

bac15203 => unPauseContract()

12065fe0 => getBalance()

5 Contract Describe

Functions will be listed as;

([Pub]) public

([Ext]) external

([Prv]) private

([Int]) internal

A yellow; denotes a function is payable.

A red; indicates that it's able to modify state.

ERC1155 Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main> surya describe ERC1155Contract.sol
 +  ERC1155Contract (Context, AccessControlEnumerable, ERC1155Burnable, ERC1155Pausable)
    - [Pub] <Constructor> #
       - modifiers: ERC1155
    - [Pub] mint #
    - [Pub] mintBatch #
    - [Pub] pause #
    - [Pub] unpause #
    - [Pub] supportsInterface
    - [Int] _beforeTokenTransfer #
    - [Pub] setMaximumRoyaltyRate #
    - [Int] setRoyalty #
    - [Int] setRoyaltyBatch #
    - [Pub] getRoyalty
    - [Pub] getRoyaltyAddress
    - [Pub] getRoyaltyRate
```

FeeStorage Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Storage> surya describe FeeStorage.sol
 +  FeeStorageContract (AccessControl)
    - [Pub] <Constructor> #
    - [Pub] setFeeOverride #
       - modifiers: onlyWriter
    - [Pub] setFeeOverrideBatch #
       - modifiers: onlyWriter
    - [Pub] setWriterRole #
    - [Pub] setReaderRole #
    - [Pub] getContractAddress
    - [Pub] setFeeWalletAddress #
       - modifiers: onlyWriter
    - [Pub] setDefaultFeeRate #
       - modifiers: onlyWriter
    - [Pub] getFeeWalletAddress
       - modifiers: onlyReader
    - [Ext] getFee
       - modifiers: onlyReader
```

IERC1155 Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Interface> surya describe IERC1155.sol
 + [Int] IERC1155
    - [Ext] isApprovedForAll #
    - [Ext] balanceOf
    - [Ext] getRoyaltyAddress
    - [Ext] getRoyalty
    - [Ext] getRoyaltyRate
    - [Ext] safeTransferFrom #
```

IFeeStorage Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Interface> surya describe IFeeStorage.sol
 + [Int] IFeeStorage
    - [Ext] getFeeWalletAddress
    - [Ext] getFee
```

ISellStorage Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Interface> surya describe ISellStorage.sol
 + [Int] ISellStorage
    - [Ext] sellRegister #
    - [Ext] setSaleStatus #
    - [Ext] setAuctionBid #
    - [Ext] getIsDefinedSellId
    - [Ext] getSellItem
    - [Ext] getAuctionTopBid
    - [Ext] getAuctionTopBidderAddress
    - [Ext] getAuctionTopBidAndBidderAddress
    - [Ext] updateSaleQuantity #
    - [Ext] updateUnitPrice #
```

SellProxyContract Describe

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Proxy> surya describe SellProxyContract.sol
 +  SellProxyContract (Validator)
    - [Pub] <Constructor> #
    - [Pub] register #
       - modifiers: pauseControl
    - [Pub] bid ($)
       - modifiers: pauseControl
    - [Pub] closeAuction #
       - modifiers: pauseControl
    - [Pub] updateUnitPrice #
       - modifiers: pauseControl
    - [Pub] closeAuctionForSystem #
       - modifiers: pauseControl,onlyMinter
    - [Pub] purchase ($)
       - modifiers: pauseControl
    - [Int] calculateRoyaltyPrice
    - [Int] sendPrice #
    - [Int] withdraw #
    - [Pub] setSellStorageAddress #
       - modifiers: pauseControl,onlyMinter
    - [Pub] pauseContract #
       - modifiers: onlyMinter
    - [Pub] unPauseContract #
       - modifiers: onlyMinter
    - [Pub] setFeeStorageAddress #
       - modifiers: onlyMinter
    - [Pub] getBalance
       - modifiers: pauseControl
```

SellStorage Describe

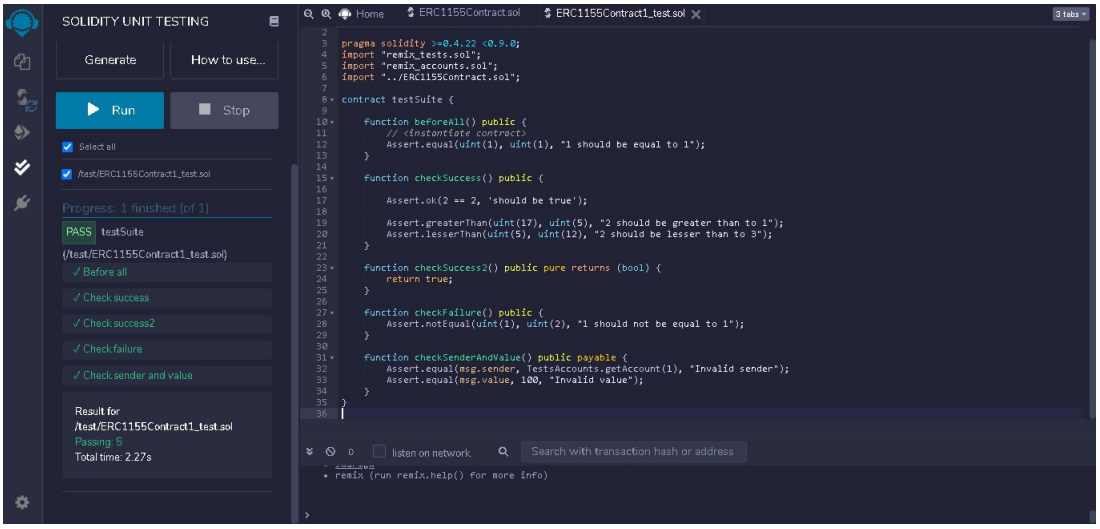
```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Storage> surya describe SellStorage.sol
 +  SellStorageContract (AccessControl)
    - [Pub] <Constructor> #
    - [Ext] sellRegister #
       - modifiers: onlyWriter
    - [Int] setSaleItem #
    - [Pub] setWriterRole #
    - [Pub] setReaderRole #
    - [Pub] getSellItem
       - modifiers: onlyReader
    - [Pub] getContractAddress
    - [Ext] getIsDefinedSellId
    - [Ext] updateUnitPrice #
       - modifiers: onlyWriter
    - [Ext] setSaleStatus #
       - modifiers: onlyWriter
    - [Ext] updateSaleQuantity #
       - modifiers: onlyWriter,onlyWriter
    - [Ext] setAuctionBid #
       - modifiers: onlyWriter
    - [Ext] getAuctionTopBid
       - modifiers: onlyReader
    - [Ext] getAuctionTopBidderAddress
       - modifiers: onlyReader
    - [Ext] getAuctionTopBidAndBidderAddress
       - modifiers: onlyReader
```

Validator

```
PS C:\Users\Bakor\Desktop\nft-marketplace-smart-contract-main\Validator> surya describe Validator.sol
 +  Validator
    - [Int] validateSell
    - [Int] validateBid
    - [Int] validateWithdraw
    - [Int] validateGetSellItem
```

6 Unit Test Example

ERC1155Contract



Sender