* CONTRACT: ERC721 for erc721 interface
* CONTRACT: ClockAuctionBase representing auction core (models, variables, and internal methods)
  + VARIABLES
    - auction struct
      * [address] address of erc721 token
      * [address] address of erc721 token seller
      * [uint128] price (wei) at beginning of auction
      * [uint128] price (wei) at end of auction
      * [uint64] duration (seconds) of auction
      * [uint64] time when auction started
    - [uint256][public] cut owner takes from auction measured in basis points (1/100 of a percent)
    - [mapping]{address}{mapping(uint256 => Auction} map from token ID to corresponding auction
  + EVENTS
    - Auction created
    - Auction successful
    - Auction cancelled
  + CALLBACK FUNCTION
    - “function() external{}”
    - Refunds funds given to contract (none should be held by contract)
  + MODIFIERS
    - canBeStoredWith64Bits()
      * Require that value can be safely stored with 64 bits
    - canBeStoredWith128Bits()
      * Require that value can be safely stored with 128 bits
  + FUNCTIONS
    - \_owns
      * Description: [internal][view][returns:bool] returns true if claimant owns the token
      * Params:
        + Address \_nft

Address of ERC721 token

* + - * + Address \_claimaint

Address of person claming to own ERC721 token

* + - * + Uint256 \_tokenId

ID of ERC721 token

* + - * Create ERC721 variable (nonFungibleCOntract) by setting equal to \_getNft(\_nft)
        + \_getNft returns ERC721 token object??
      * Return true if owner of nonFungibleContract is equal to param claimant
    - \_escrow
      * Description: [internal] escrows the ERC721 token, assigning ownership to this contract (throw if escrow fails)
      * Params:
        + Address \_nft

Address of ERC721 token

* + - * + Address \_owner

Current owner of ERC721 token in escrow

* + - * + Uint256 \_tokenId

ID of erc721 token to verify

* + - \_transfer
      * Description: [internal] transfers ERC721 token owned by exchange contract to buyer address
    - \_addAuction
      * Description: [internal] adds auction to list of open auctions, fires auctionCreated event
    - \_cancelAuction
      * Description: [internal] cancels auction by calling removeAuction(), transferring ERC721 token back to seller address, and calling auctionCancelled()
    - \_bid
      * Description: [internal][returns:uint256] computes price and transfers winnings, doesn’t transfer ownership of token
      * Does not call transfer function
    - \_removeAuction
      * Description: [internal] removes auction from list of open auctions
    - \_isOnAuction
      * Description: [internal][view][returns:bool] returns true if ERC721 token is on auction
    - \_currentPrice
      * Description: [internal][view][returns:uint256] returns current price (in erc20 token) of erc721 token on auction
    - \_computeCurrentPrice
      * Description: [internal][pure][returns:uint256] computer current price of auction, factored out from \_currentPrice so we can run unit tests
    - \_computeCut
      * Description: [internal][view][returns:uint256] returns sellers cut of sale
    - \_getNft
      * Description: [internal][view][returns(ERC721)]gets ERC721 object from address
* CONTRACT: Ownable
  + Use as is from OpenZeppelin
* CONTRACT: Pausable (is Ownable)
  + Use as is from OpenZeppelin
* CONTRACT: ClockAuction (is Pausable, ClockAuctionBase) represents clock auction for ERC7221 token
  + FUNCTIONS
    - CONSTRUCTOR
      * Description: [public] verifies owner cut is in valid range
    - withdrawBalance
      * Description: [external] remove all ether from this contract (which is the owner’s cuts as well as any Ether sent directly to the contract address and send to msg.sender if msg.sender is owner
    - createAuction
      * Description: [public][whenNotPaused][canBeStoredWith128Bits(\_startingPrice)[canBeStoredWith128Bits(\_endingPrice)][canBeStoredWith64Bits(\_duration)] starts new auction
      * Require that owner of ERC721 token is msg.sender
      * Put ERC721 token in escrow
      * Create new auction struct
      * Call \_addAuction
    - Bid
      * Description: [public][payable][whenNotPaused] bids on an open auction, completing the auction and transferring ownership of ERC721 token if enough ether is supplied
      * Call \_bid
      * Call \_transfer
    - cancelAuction
      * Description: [public] cancels an auction that hasn’t been won yet and returns ERC721 token to original owner
      * Reference auction to cancel
      * Require that auction is live
      * Require that the msg.sender is the auction seller
      * Call \_cancelAuction