```
<<Abstract>>
                                          Ownable
                                        AuctionLogic
        Private:
         owner: address
         ownerMap: mapping(address=>OwnerData)
        Internal:
          checkOwner()
          transferOwnership(newOwner: address)
          addOwnerToMap()
        Public:
         <<event>> OwnershipTransferred(previousOwner: address, newOwner: address)
         <<modifier>>> onlyOwner()
         constructor()
         owner(): address
         renounceOwnership()
         transferOwnership(newOwner: address)
         getOwnerData(ownerAddress: address): OwnerData
                                                                       <<Abstract>>>
                                                                       Ownable2Step
                                                                       AuctionLogic
                                     Private:
                                      _pendingOwner: address
                                      renounceProcessAccepted: bool
                                     Internal:
          <<Struct>>
                                        transferOwnership(newOwner: address)
          OwnerData
                                     Public:
         AuctionLogic
                                       <<event>> OwnershipTransferStarted(previousOwner: address, newOwner: address)
ownerBlockNumber: uint256
                                       <<event>>> RenounceOwnershipProcessStarted(owner: address)
ownerBlockTimestamp: uint256
                                       <<event>>> RenounceOwnershipProcessReset(owner: address)
ownerBlockhash: bytes32
                                       pendingOwner(): address
                                       renounceProcessAccepted(): bool
                                       transferOwnership(newOwner: address)
                                       acceptOwnership()
                                       resetPendingOwner()
                                       startRenounceOwnershipProcess()
                                       resetRenounceOwnershipProcess()
                                      renounceOwnership()
                                                                       <<Abstract>>
                                                                        Cancellable
                                                                       AuctionLogic
                                                      Private:
                                                        cancelled: bool
                                                      Internal:
                                                        cancelAuction()
                                                      Public:
                                                        <<abstract>>> startAuctionCancellationProcess()
                                                        <<abstract>> resetAuctionCancellationProcess()
                                                        <<abstract>>> cancelAuction2Step()
                                                        <<event>>> AuctionCancelled(owner: address)
                                                        <<modifier>> whenNotCancelled()
                                                       cancelled(): bool
                                                                       AuctionLogic
                                                                       AuctionLogic
Private:
 _bidAmountsOfBidders: mapping(address=>uint256)
  auctionHighestBidAmount: uint256
 auctionWinner: address
  auctionStartBlock: uint256
 auctionEndBlock: uint256
  cancellationProcessAccepted: bool
 startingPrice: uint256
  bidIncrement: uint256
 closedAuction: bool
 whitelistedParticipants: mapping(address=>bool)
 reservePrice: uint256
Internal:
 isWhitelisted(participant_: address): bool
Public:
 <<pre><<pre>payable>>> placeBid(): bool
 <event>> BidPlacedSuccessfully(bidder : address, previousHighestBidAmount : uint256, newHighestBidAmount : uint256)
 <<event>> WithdrewSuccessfully(entity_: address, withdrawAmount_: uint256)
 <<modifier>> onlyWhenNotOwner()
 <<modifier>>> onlyWhenAfterStartBlock()
 <<modifier>>> onlyWhenBeforeEndBlock()
 constructor(auctionStartBlock: uint256, auctionEndBlock: uint256, startingPrice: uint256, bidIncrement: uint256, reservePrice: uint256, closedAuction: bool)
 auctionHighestBidAmount(): uint256
 auctionWinner(): address
 auctionStartBlock(): uint256
 auctionEndBlock(): uint256
 getBidAmountOfBidder(bidder_: address): uint256
 cancellationProcessAccepted(): bool
 startingPrice(): uint256
 bidIncrement(): uint256
 closedAuction(): bool
 reservePrice(): uint256
 startAuctionCancellationProcess()
 resetAuctionCancellationProcess()
 cancelAuction2Step()
 addParticipantToWhitelist(participant_: address): bool
 addMultipleParticipantsToWhitelist(participants : address[]): bool
 withdraw(): bool
```