pragma solidity ^ 0.5.1;

/// @title Auction abstract contract

/// @author Andrea Bruno 585457

/// @notice This contract implements the shared functionalities between the Dutch and Vickrey auction.

/// @dev The following comments are written using the Solidity NatSpec Format.

contract Auction {

/// @notice This struct incorporates the basic info about the auction.

struct Description {

address payable seller;

string itemName;

uint startBlock;

address winnerAddress;

uint winnerBid;

}

/// @dev The description is public to allow people to read it without paying any gas.

Description public description;

/// @dev This modifier allow only the seller to invoke the function.

modifier onlySeller() {

require(msg.sender == description.seller, "Only the seller can run this function");

\_;

}

/// @notice This event communicate the beginning of the auction.

event auctionStarted();

/// @notice This event communicate the end of the auction

/// @param winnerAddress is the address of the winner

/// @param winnerBid is the final price that the winner has paid.

/// @param surplusFounds is the eventual surplus left to the contract.

event auctionFinished(address winnerAddress, uint winnerBid, uint surplusFounds);

/// @notice This abstract function allow to activate the auction.

/// @dev For security reason, it is better to equip this function with `onlySeller` modifier.

function activateAuction() public;

/// @notice This abstract function finalize and close the auction.

/// @dev For security reason, it is better to equip this function with `onlySeller` modifier.

function finalize() public;

}