pragma solidity ^0.4.19;

import "./Manager.sol";

contract portauthority is main { //this declares a complextype which will be used for variable later//

event departure(uint shipmentId); //we are creating a event for export//

event arrival(uint shipmentId);//we are creating a event for import //

function setshipmentId(uint orderId,uint \_shipmentId) {

itemMap[orderId].shipmentId = \_shipmentId;

}

function stateRequiredTimeToNextEntity(uint \_orderId, uint \_requiredTime){ // Give Estimate;

statsMap[\_orderId].timeToNextEntity = \_requiredTime;

}

function confirm\_shipment (uint orderId) public returns (bool success) {

require(msg.sender==flowOfObject[orderId].Addresses[currentaddress[orderId]]);

departure(itemMap[orderId].shipmentId) ;

statsMap[orderId].checkPoint="successfully containers load into the ship"; // Updates currentStatusOfOrder.

statsMap[orderId].timeTheEventCalled=now;

return true;

}

function shipmentLoaded (uint orderId) public returns (bool success) {

arrival(itemMap[orderId].shipmentId);

statsMap[orderId].checkPoint="successfully containers load into the port"; // Updates currentStatusOfOrder.

statsMap[orderId].timeTheEventCalled=now;

transferPossesion(orderId);

return true;

}

}