```
/*
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

namespace org.reliance.network

//Creating concept location to track latitude longitude location of the trader concerned.
//Not part of problem statement

concept Location{
  o String latitude
  o String longitude
}

// Defining the participants on the network

abstract participant Trader{
  o String address
  o Double balance
  o Location location
}

participant Exporter identified by emailID extends Trader{
  o String emailID
}

participant Importer identified by emailID extends Trader{
  o String emailID
}

participant Shipper identified by emailID extends Trader{
  o String emailID
}


// Defining all the lists required for creating assets

enum AssetType{
  o medicine
  o fuel
  o trucks
  o wires
  o tables
  o laptops
  o chairs
}

enum ShipmentStatus{
  o created
  o inTransit
```

```
  o arrived
}

enum CompassDirection{
  o N
  o S
  o E
  o W
}


// Defining the assets on the chain

asset Shipment identified by shipmentID{
  o String shipmentID
  o AssetType assettype
  o ShipmentStatus status
  o Long units
  o Contract contract
  o TempReading[] temperatures optional
  o AccelerationReading[] accelerations optional
  o GPSReading[] gpsreadings optional
}

asset Contract identified by contractID{
  o String contractID
  --> Exporter exporter
  --> Importer importer
  --> Shipper shipper
  o DateTime arrival
  o Double unitprice
  o Double mintemp
  o Double maxtemp
  o Double minpanaltyfactor
  o Double maxpenaltyfactor
  o Double maxacceleration
}


 // Defining the transactions

abstract transaction ShipmentTransaction{
  --> Shipment shipment
}

transaction AccelerationReading extends ShipmentTransaction{
  o Double accelerationx
  o Double accelerationy
  o Double accelerationz
  o String latitude
  o String longitude
  o String readingtime
}

transaction TempReading extends ShipmentTransaction{
  o Double celcius
  o String latitude
  o String longitude
  o String readingtime
}
```

```
transaction GPSReading extends ShipmentTransaction{
  o String latitude
  o CompassDirection latitudedirection
  o String longitude
  o CompassDirection longitudedirection
  o String readingtime
  o String readingdate
}

transaction ShipmentReceived{
  --> Shipment shipment
}


// Defining the events that trigger transactions on the network

event TemperatureThreshold{
  o Double temperature
  o String message
  o String latitude
  o String longitude
  o String readingtime
  --> Shipment shipment
}

event AccelerationThreshold{
  o Double accelerationx
  o Double accelerationy
  o Double accelerationz
  o String latitude
  o String longitude
  o String readingtime
  --> Shipment shipment
}

event ShipmentInPort{
  o String message
  --> Shipment shipment
}
```