

Classes and interfaces

Party: main abstract class for all classes which

Farmer: abstract class for farmer classes

Wheat farmer: produces wheat

Cow farmer: produces meat and milk products

Orange farmer: produces oranges

Processor: abstract class for processor classes

Meat processor: produces sausages

Bakery processor: produces bread, cream cheese cake with oranges

Alcohol processor: produces vodka

Juice processor: produces juice

Distributor: distributes products between retailers

Retailer: sales products to customers

Customer: buies products

Storage: abstract class for storage classes

Processor storage: stores products for processor

Distributor storage: stores products for processor

Retailer storage: stores products for processor

Reportable: interface for product, to report information

Product: product entity in simulation

Product in process : decorator for product when product is in parties process

State: state of "product in process"

Handler: interface for handlers which handle exact processes in production

Eco-system : general class for simulation

Discreet step: one day in simulation, keeps all events that were made by parties this day

Observer: interface for observers

Subject: interface for step

Transaction report observer: reports information about transactions made in one "Discreet step"

Security report observer: reports information about violations happened in one "Discreet step"

Container: container interface for iterator

Block: unit in block chain

Iterator: interface for Party iterator

Party iterator: iterates parties

Transaction: operations between parties

Money unit: money unit in the simulation

Channel: channel for transactions

Channel type: channel type

Request: request for one transaction from one party to another