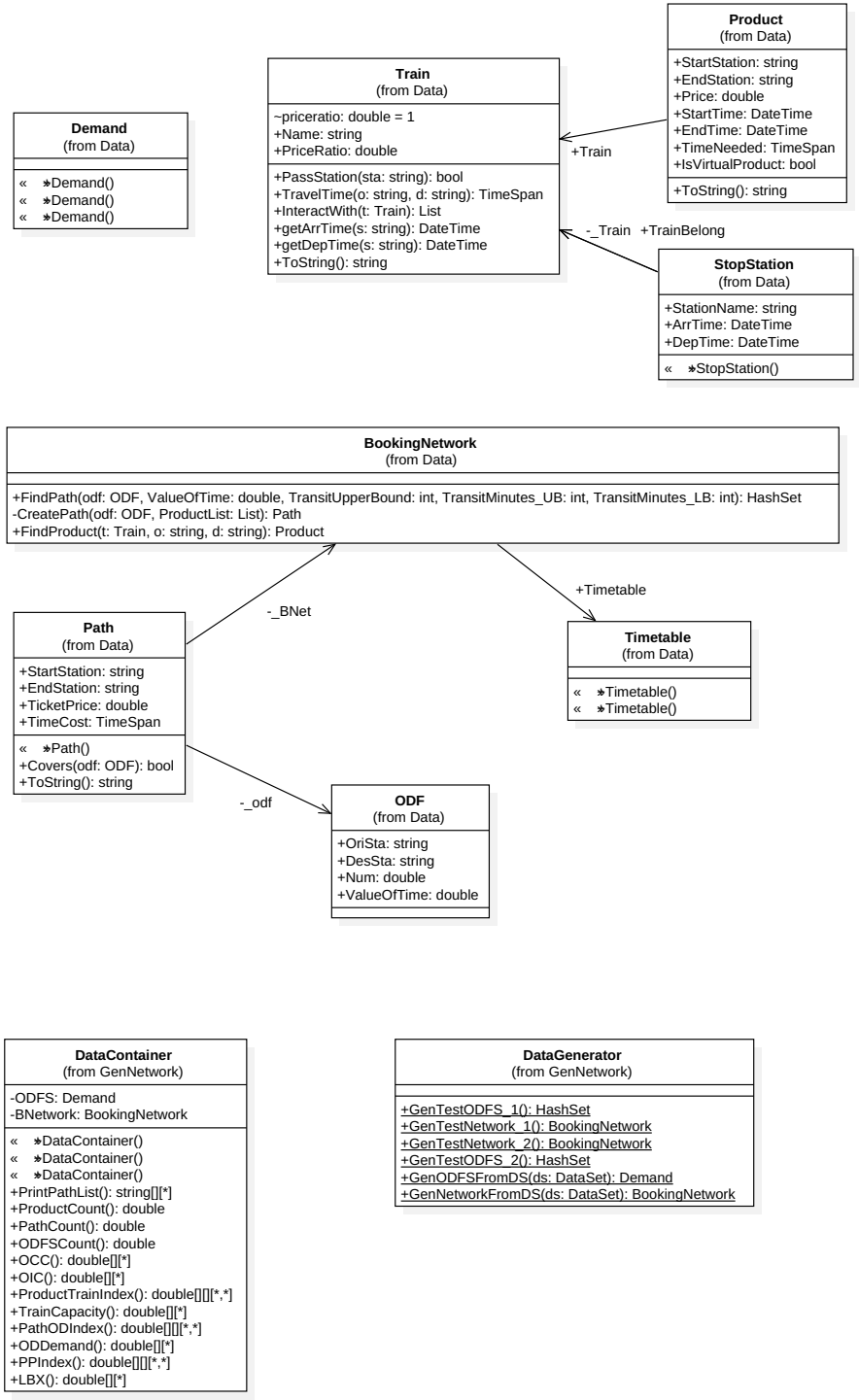
Model::Main


Demand
(from Data)
« Demand()
« Demand()
« Demand()
Train
(from Data)
~priceratio: double = 1
+Name: string
+PriceRatio: double
+PassStation(sta: string): bool
+TravelTime(o: string, d: string): TimeSpan
+InteractWith(t: Train): List
+getArrTime(s: string): DateTime
+getDepTime(s: string): DateTime
+ToString(): string
Product
(from Data)
+StartStation: string
+EndStation: string
+Price: double
+StartTime: DateTime
+EndTime: DateTime
+TimeNeeded: TimeSpan
+IsVirtualProduct: bool
+ToString(): string
+Train
-_Train
+TrainBelong
StopStation
(from Data)
+StationName: string
+ArrTime: DateTime
+DepTime: DateTime
« StopStation()
BookingNetwork
(from Data)
+FindPath(odf: ODF, ValueOfTime: double, TransitUpperBound: int, TransitMinutes_UB: int, TransitMinutes_LB: int): HashSet
-CreatePath(odf: ODF, ProductList: List): Path
+FindProduct(t: Train, o: string, d: string): Product
-_BNet
+Timetable
Path
(from Data)
+StartStation: string
+EndStation: string
+TicketPrice: double
+TimeCost: TimeSpan
« Path()
+Covers(odf: ODF): bool
+ToString(): string
Timetable
(from Data)
« Timetable()
« Timetable()
-_odf
ODF
(from Data)
+OriSta: string
+DesSta: string
+Num: double
+ValueOfTime: double
DataContainer
(from GenNetwork)
-ODFS: Demand
-BNetwork: BookingNetwork
« DataContainer()
« DataContainer()
« DataContainer()
+PrintPathList(): string[][*]
+ProductCount(): double
+PathCount(): double
+ODFSCount(): double
+OCC(): double[][*]
+OIC(): double[][*]
+ProductTrainIndex(): double[][][*,*]
+TrainCapacity(): double[][*]
+PathODIndex(): double[][][*,*]
+ODDemand(): double[][*]
+PPIndex(): double[][][*,*]
+LBX(): double[][*]
DataGenerator
(from GenNetwork)
+GenTestODFS_1(): HashSet
+GenTestNetwork_1(): BookingNetwork
+GenTestNetwork_2(): BookingNetwork
+GenTestODFS_2(): HashSet
+GenODFSFromDS(ds: DataSet): Demand
+GenNetworkFromDS(ds: DataSet): BookingNetwork