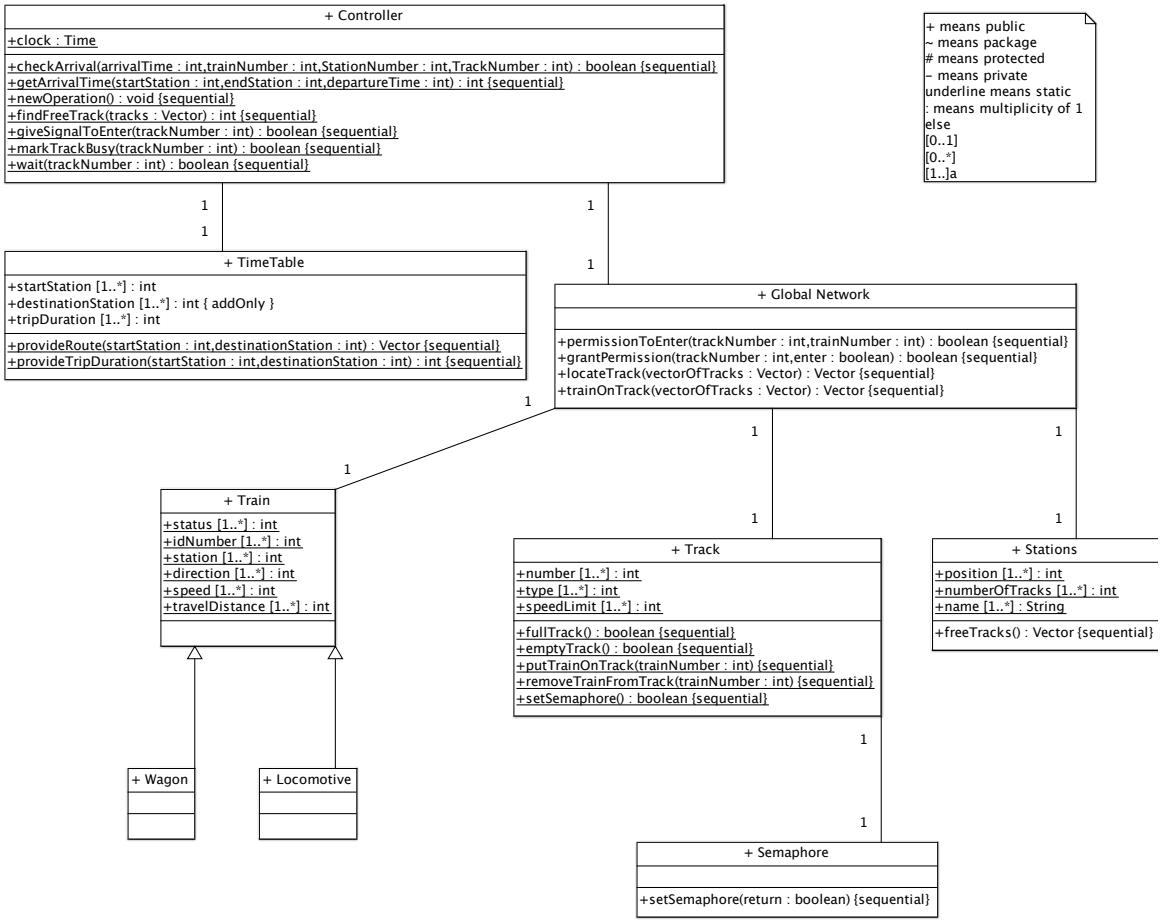
+ Controller
+clock : Time
+checkArrival(arrivalTime : int,trainNumber : int,StationNumber : int,TrackNumber : int) : boolean {sequential}
+getArrivalTime(startStation : int,endStation : int,departureTime : int) : int {sequential}
+newOperation() : void {sequential}
+findFreeTrack(tracks : Vector) : int {sequential}
+giveSignalToEnter(trackNumber : int) : boolean {sequential}
+markTrackBusy(trackNumber : int) : boolean {sequential}
+wait(trackNumber : int) : boolean {sequential}
+ means public
~ means package
# means protected
- means private
underline means static
: means multiplicity of 1
else
[0..1]
[0..*]
[1..]a
+ TimeTable
+startStation [1..*] : int
+destinationStation [1..*] : int { addOnly }
+tripDuration [1..*] : int
+provideRoute(startStation : int,destinationStation : int) : Vector {sequential}
+provideTripDuration(startStation : int,destinationStation : int) : int {sequential}
+ Global Network
+permissionToEnter(trackNumber : int,trainNumber : int) : boolean {sequential}
+grantPermission(trackNumber : int,enter : boolean) : boolean {sequential}
+locateTrack(vectorOfTracks : Vector) : Vector {sequential}
+trainOnTrack(vectorOfTracks : Vector) : Vector {sequential}
+ Train
+status [1..*] : int
+idNumber [1..*] : int
+station [1..*] : int
+direction [1..*] : int
+speed [1..*] : int
+travelDistance [1..*] : int
+ Track
+number [1..*] : int
+type [1..*] : int
+speedLimit [1..*] : int
+fullTrack() : boolean {sequential}
+emptyTrack() : boolean {sequential}
+putTrainOnTrack(trainNumber : int) {sequential}
+removeTrainFromTrack(trainNumber : int) {sequential}
+setSemaphore() : boolean {sequential}
+ Stations
+position [1..*] : int
+numberOfTracks [1..*] : int
+name [1..*] : String
+freeTracks() : Vector {sequential}
+ Wagon
+ Locomotive
+ Semaphore
+setSemaphore(return : boolean) {sequential}

+ Wagon