Design a system model

- **Scenario:**
  
  Motor vehicle could be a car, a bus, or a truck. Each vehicle has wheels, an engine. They are managed/driven by drivers. They can transport passengers and loads. All persons could be passengers and some of them could be a driver.
Classes of objects for our model

- Engine
- Wheel
- MotorVehicle
- Load
- Person
- Driver
- Car
- Bus
- Truck
Inheritance
Is-a relationship

- Engine
- Wheel
- MotorVehicle
- Load
- Person
- Driver
- Car
- Bus
- Truck
Association
Aggregation, composition
Attribute (global variables)  
Operations (methods)

MotorVehicle
- engine: Engine
- wheels: Collection
- loads: Collection
- passengers: Collection
- driver: Driver

Load
- weight: double
- size
+ getWeight(): double
+ getSize(): <unspecified>

Person
+ address: string
+ Person()

Driver
+ Driver()
+ driver()

Car
+ Car()

Bus
+ Bus()

Truck
+ Truck()
Public attributes

- All variables are private
- To access the value
  - getMethod
  - setMethod
- Input for operations

MotorVehicle
- engine : Engine
- wheels : Collection
- loads : Collection
- passengers : Collection
- driver : Driver

#MotorVehicle()
+ getEngine() : Engine
+ load(in passenger : Person) : bool
+ unload(in passenger : Person) : bool
+ load(in load : Load) : bool
+ unload(in load : Load) : bool
+ getWheels() : Collection
+ setWheels(in wheels : Collection)
+ getLoads() : Collection
+ setLoads(in loads : Collection)
+ getPassengers() : Collection
+ setPassengers(in passenger : Collection)
+ getDriver() : Driver
+ setDriver(in driver : Driver)
More information

Grady Booch

- Object-Oriented Analysis and Design
  - with Applications