Person

- firstName:string
- middleName:string
- lastName:string
- birthTimestamp:int
- height:double
- uuid:string
+ bankAccount:BankAccount
+ vehicles:Vehicle[0..*]

+ getFirstName():string
+ setFirstName(newName:string):bool
+ getMiddleName():string
+ setMiddleName(newName:string):bool
+ getLastName():string
+ setLastName(newName:string):bool
+ getBirthTimestamp():int
+ changeHeight(deltaHeight:double):bool
+ getAge():double
+ serializeToJSON():json
+ saveAsFile():void

BankAccount

- balance:double
- minimumBalance:double
- withdrawLimit:double
- depositLimit:double
- uuid:string

+ getBalance():double
+ getUUID():string
+ withdraw(amount:double):bool
+ deposit(amount:double):bool
+ checkWithdraw(proposedAmount:double):bool
+ checkDeposit(proposedAmount:double):bool
+ serializeToJSON():json
+ saveAsFile():void

<<Abstract>> Vehicle

name:string
price:double
wheels:int
doors:int
seats:int
maxPassengers:int
maxSpeed:double
manufacturer:string
mileage:double
horsepower:double
driver:Person[0..1]
started:bool
color:string
uuid:string
type:string

+ start():bool
+ stop():bool
+ drive(distance:double):bool
+ addDriver(driver:Person):bool
+ removeDriver():bool
+ changeColor(color:string):bool
+ changePrice(newPrice:double):bool
+ getMaxSpeed():double
+ getWheels():int
+ getDoors():int
+ getSeats():int
+ getHorsepower():double
+ getPrice():double
+ getMaxSpeed():double
+ getName():string
+ getUUID():string
+ saveAsFile():void
+ approximateFuelUsageFromKm(double):double
+ *serializeToJSON():json*
+ *toFormattedString():string*
+ getSanitizedJSON(json):json

Sedan

- trunkCapacity:double
- engineCylinderCount:int

+ getTrunkCapacity():double
+ getEngineCylinderCount():int
+ approximateFuelUsageFromKm(kilometres:double):double
+ serializeToJSON():json
+ deserializeFromJSON(data:json):Sedan
+ loadFromUUID(uuid:string):Sedan
+ loadFromPath(path:string):Sedan
+ toFormattedString:string

PickupTruck

- bedCapacity:double
- towingMaxLoad:double
- engineCylinderCount:int

+ getBedCapacity():double
+ getTowingMaxLoad():double
+ getEngineCylinderCount():double
+ approximateFuelUsageFromKm(kilometres:double):double
+ serializeToJSON():json
+ deserializeFromJSON(data:json):PickupTruck
+ loadFromUUID(uuid:string):PickupTruck
+ loadFromPath(path:string):PickupTruck
+ toFormattedString():string

Motorcycle

- engineSize:double
- maxAcceleration:double
- motorcycleType:MOTORCYCLE_TYPE

+ getEngineSize():double
+ getMaxAccel():double
+ approximateFuelUsageFromKm(kilometres:double)
+ serializeToJSON():json
+ deserializeFromJSON(data:json):Motorcycle
+ loadFromUUID(uuid:string):Motorcycle
+ loadFromPath(path:string):Motorcycle
+ toFormattedString():string

VehicleDealership

- name:string
- uuid:string
- bankAccount:BankAccount
+ vehicles:Vehicle[0..*]

+ getName():string
+ buyVehicleFrom(idx:int, buyerBankAccount:BankAccount):Vehicle
+ sellVehicleTo(vehicle:Vehicle, sellerBankAccount:BankAccount):int
+ getVehicle(vehicle:Vehicle):int
+ serializeToJSON():json
+ saveAsFile():void
+ deserializeFromJSON(data:json):VehicleDealership
+ loadFromUUID(uuid:string):VehicleDealership
+ loadFromPath(path:string):VehicleDealership
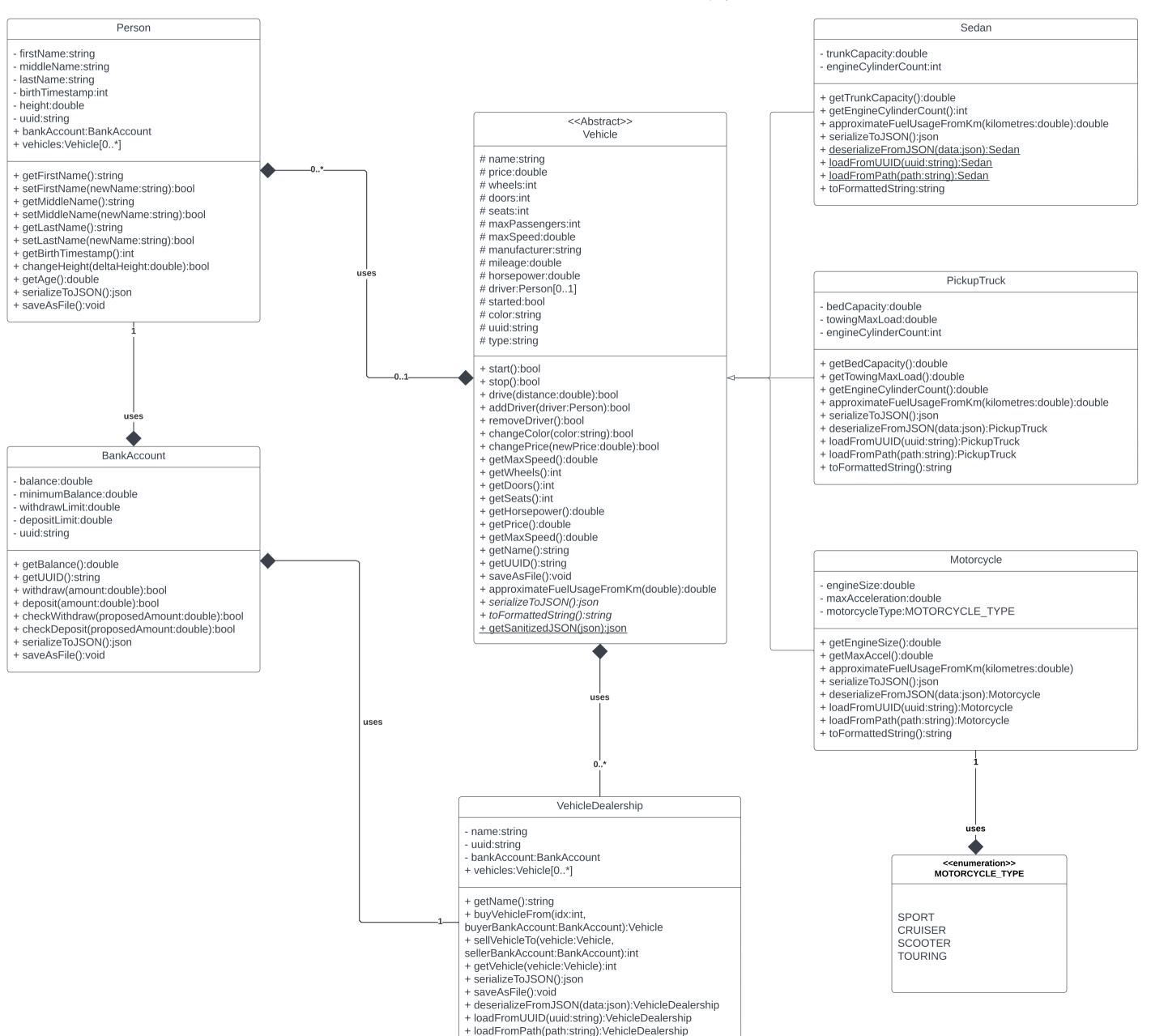
std::numpunct<char>

+ id:locale_id

+ decimal_point():char
+ thousands_sep():char
+ grouping():string
+ truename():string
+ falsename():string
# *do_decimal_point():char*
# *do_thousands_sep():char*
# *do_grouping():string*
# *do_truename():string*
# *do_falsename():string*

CommaNumPunct

---

do_decimal_point():char
do_thousands_sep():char
do_grouping():string
do_truename():string
do_falsename():string

<<enumeration>> MOTORCYCLE_TYPE

SPORT
CRUISER
SCOOTER
TOURING

Relationships:
- Person 0..* uses Vehicle 0..1
- Person 1 uses BankAccount
- BankAccount 1 uses VehicleDealership
- VehicleDealership 0..* uses Vehicle
- Motorcycle 1 uses MOTORCYCLE_TYPE