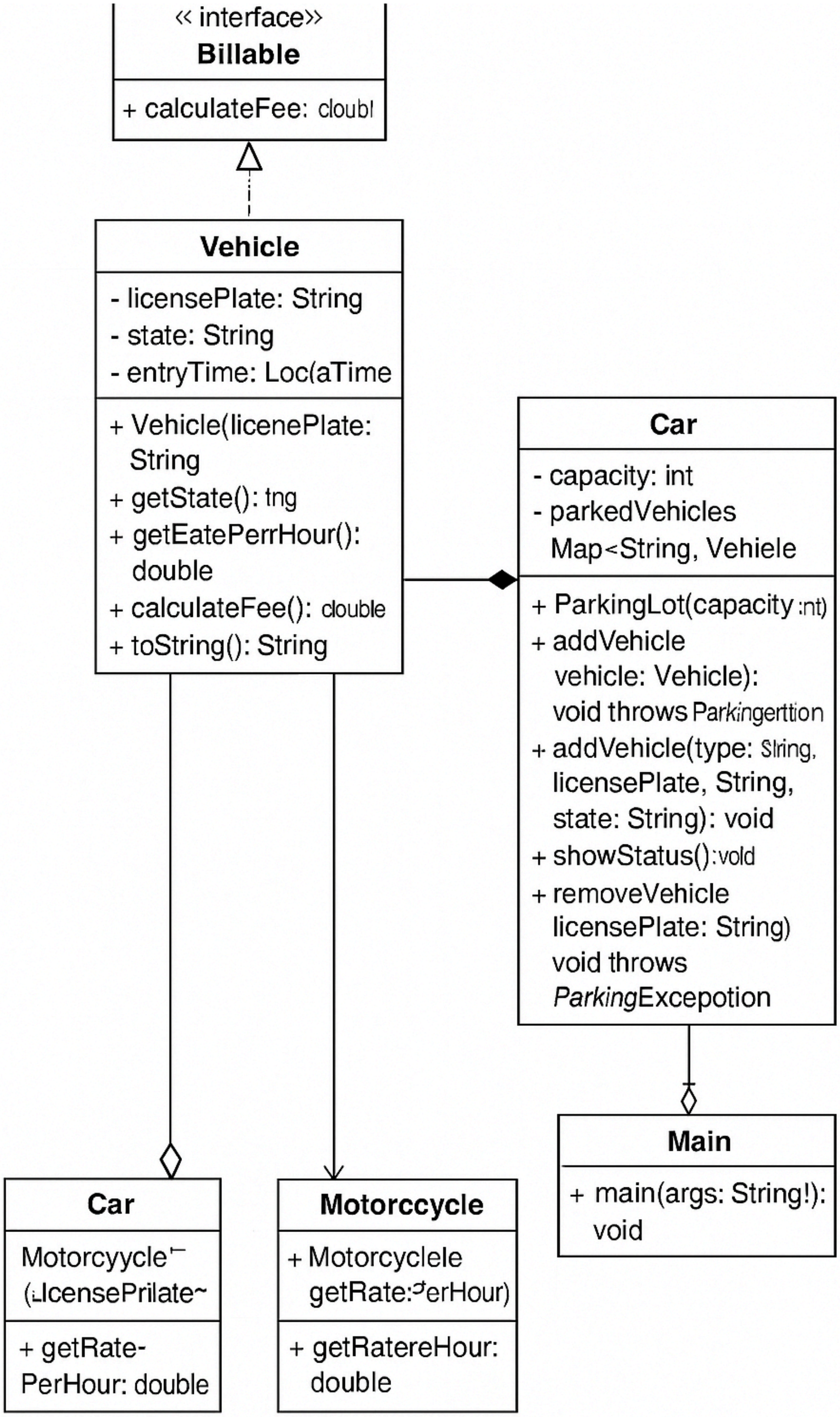
« interface»
Billable
+ calculateFee: cloubl
Vehicle
- licensePlate: String
- state: String
- entryTime: Loc(aTime
+ Vehicle(licenePlate: String
+ getState(): tng
+ getEatePerrHour(): double
+ calculateFee(): clouble
+ toString(): String
Car
- capacity: int
- parkedVehicles Map<String, Vehiele
+ ParkingLot(capacity :nt)
+ addVehicle vehicle: Vehicle): void throws Parkingerttion
+ addVehicle(type: Slring, licensePlate, String, state: String): void
+ showStatus():vold
+ removeVehicle licensePlate: String) void throws ParkingExcepotion
Main
+ main(args: String!): void
Car
Motorcyycle
(ꓕlcensePrilate~
+ getRate-PerHour: double
Motorccycle
+ Motorcyclele getRate:ᵓerHour)
+ getRatereHour: double

Classes and interfaces

- Billable (interfac)
    - + calculateFee(): double

Vehicle (abstract class, implements Billable)

- Fields (private)
    - licensePlate: String
    - state: String
    - entryTime: LocalDateTime
- Methods (public)
    - + Vehicle(licensePlate: String, state: String)
    - + getLicensePlate(): String
    - + getState(): String
    - + getEntryTime(): LocalDateTime
    - + getHoursParked(): long
    - + getRatePerHour(): double (abstract)
    - + calculateFee(): double
    - + toString(): String
- Relationships
    - implements Billable
    - superclass of Car and Motorcycle

Car (class, extends Vehicle)

- Methods
    - + Car(licensePlate: String, state: String)
    - + getRatePerHour(): double

Motorcycle (class, extends Vehicle)

- Methods
    - + Motorcycle(licensePlate: String, state: String)
    - + getRatePerHour(): double

ParkingLot (class)

- Fields (private)
    - capacity: int
    - parkedVehicles: Map<String, Vehicle>

- Methods (public)
    - + ParkingLot(capacity: int)
    - + addVehicle(vehicle: Vehicle): void throws ParkingException
    - + addVehicle(type: String, licensePlate: String, state: String): void throws ParkingException
    - + showStatus(): void
    - + removeVehicle(licensePlate: String): void throws ParkingException

- Relationships
    - uses/aggregates Vehicle
    - depends on ParkingException

ParkingException (class, extends Exception)

- Methods
    - + ParkingException(message: String)

Main (class)

- contains main(String[] args): void

Relationships

- uses ParkingLot

Relationships Summary

- Vehicle is an abstract class implementing the Billable interface
- Car and Motorcycle inherit from Vehicle
- ParkingLot aggregates many Vehicle objects in a map
- ParkingLot throws a ParkingException
- Main uses ParkingLot
- ParkingException extends Exception