# Object and Method Reference (Simulation-Ready)

This reference reflects the UML with Verilog-style heartbeat simulation methods (evaluate/commit/getState) added across all core objects.

## Restaurant

Attributes:  
- name: String  
- tables: List<Table>  
- menu: List<Platter>  
- staff: List<Staff>  
- revenue: double  
- avgWalkTime: double  
- globals: GlobalVariables  
- decisionEngine: GlobalMethods  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ simulateDay()  
+ decideStaffing()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Hostess

Attributes:  
- name: String  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ greetCustomer(Customer)  
+ seatCustomer(Customer, Table)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## CashRegister

Attributes:  
- totalStoreIncome: double  
- totalTips: double  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ processPayment(Server, Order, double tip)  
+ recordTip(Server, double tip)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Table

Attributes:  
- id: int  
- capacity: int  
- location: String  
- occupied: boolean  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ assignCustomer(Customer)  
+ clearTable()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Platter

Attributes:  
- name: String  
- ingredients: List<Ingredient>  
- cookingTime: int  
- price: double  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ prepare()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Ingredient

Attributes:  
- name: String  
- quantity: double  
- costPerUnit: double  
- expiryDate: Date  
- tickCounter: int  
- nextTickCounter: int

Methods:  
(no specific methods)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Cook

Attributes:  
- name: String  
- skillLevel: int  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ cook(Platter)  
+ prep(Platter)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Server

Attributes:  
- name: String  
- tablesAssigned: List<Table>  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ serve(Table)  
+ takeOrder(Customer)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## ServerTable

Attributes:  
- bell: boolean  
- orderQueue: Queue<Order>  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ announceOrder(Order)  
+ pickUpOrder(Server)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Order

Attributes:  
- platter: Platter  
- table: Table  
- timePlaced: int  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ getTable()  
+ getPlatter()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Customer

Attributes:  
- groupSize: int  
- patience: int  
- satisfaction: int  
- tablePreference: String  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ requestTable(String)  
+ order(Platter)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Inventory

Attributes:  
- ingredients: List<Ingredient>  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ restock()  
+ checkStock(Platter)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Refrigerator

Attributes:  
- capacity: double  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ store(Ingredient)  
+ remove(Ingredient)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Stove

Attributes:  
- slots: int  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ startCooking(Platter)  
+ finishCooking(Platter)  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## Finance

Attributes:  
- fixedCosts: double  
- laborCosts: double  
- tickCounter: int  
- nextTickCounter: int

Methods:  
+ calculateNetRevenue()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## SimulationClock

Attributes:  
- currentTime: int  
- tickRate: int

Methods:  
+ advanceTime()  
+ getCurrentTime()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]

## WaveformViewer

Attributes:  
- hb: HeartbeatManager  
- processes: List<SimProcess>

Methods:  
+ runDebug(int)  
+ tickle()  
+ evaluate(): boolean  
+ commit(): void  
+ getState(): String[][]