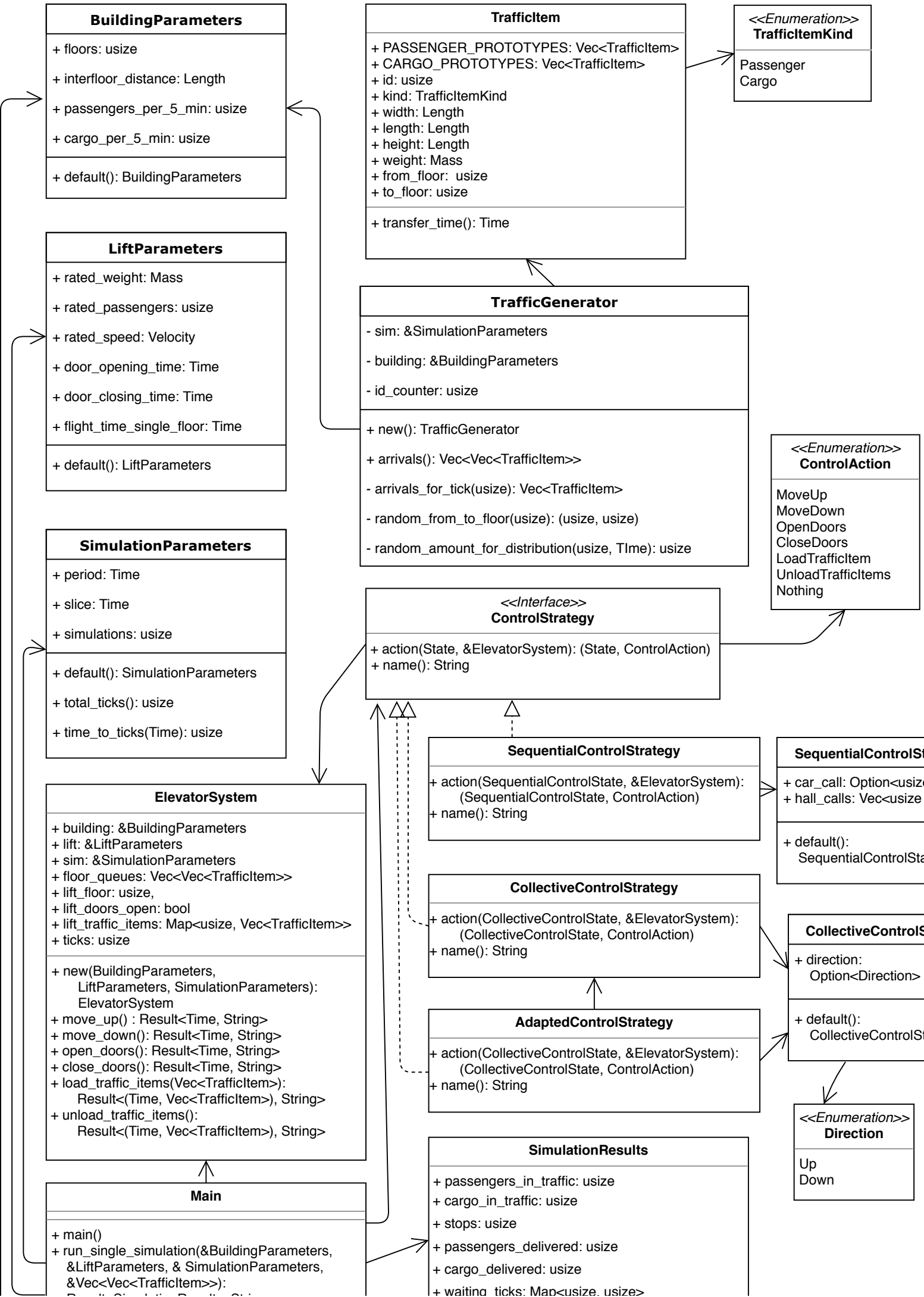
BuildingParameters
+ floors: usize
+ interfloor_distance: Length
+ passengers_per_5_min: usize
+ cargo_per_5_min: usize
+ default(): BuildingParameters
TrafficItem
+ PASSENGER_PROTOTYPES: Vec<TrafficItem>
+ CARGO_PROTOTYPES: Vec<TrafficItem>
+ id: usize
+ kind: TrafficItemKind
+ width: Length
+ length: Length
+ height: Length
+ weight: Mass
+ from_floor: usize
+ to_floor: usize
+ transfer_time(): Time
<<Enumeration>>
TrafficItemKind
Passenger
Cargo
LiftParameters
+ rated_weight: Mass
+ rated_passengers: usize
+ rated_speed: Velocity
+ door_opening_time: Time
+ door_closing_time: Time
+ flight_time_single_floor: Time
+ default(): LiftParameters
TrafficGenerator
- sim: &SimulationParameters
- building: &BuildingParameters
- id_counter: usize
+ new(): TrafficGenerator
+ arrivals(): Vec<Vec<TrafficItem>>
- arrivals_for_tick(usize): Vec<TrafficItem>
- random_from_to_floor(usize): (usize, usize)
- random_amount_for_distribution(usize, TIme): usize
<<Enumeration>>
ControlAction
MoveUp
MoveDown
OpenDoors
CloseDoors
LoadTrafficItem
UnloadTrafficItems
Nothing
SimulationParameters
+ period: Time
+ slice: Time
+ simulations: usize
+ default(): SimulationParameters
+ total_ticks(): usize
+ time_to_ticks(Time): usize
<<Interface>>
ControlStrategy
+ action(State, &ElevatorSystem): (State, ControlAction)
+ name(): String
SequentialControlStrategy
+ action(SequentialControlState, &ElevatorSystem):
(SequentialControlState, ControlAction)
+ name(): String
SequentialControlState
+ car_call: Option<usize>
+ hall_calls: Vec<usize
+ default():
SequentialControlState
ElevatorSystem
+ building: &BuildingParameters
+ lift: &LiftParameters
+ sim: &SimulationParameters
+ floor_queues: Vec<Vec<TrafficItem>>
+ lift_floor: usize,
+ lift_doors_open: bool
+ lift_traffic_items: Map<usize, Vec<TrafficItem>>
+ ticks: usize
+ new(BuildingParameters,
LiftParameters, SimulationParameters):
ElevatorSystem
+ move_up() : Result<Time, String>
+ move_down(): Result<Time, String>
+ open_doors(): Result<Time, String>
+ close_doors(): Result<Time, String>
+ load_traffic_items(Vec<TrafficItem>):
Result<(Time, Vec<TrafficItem>), String>
+ unload_traffic_items():
Result<(Time, Vec<TrafficItem>), String>
CollectiveControlStrategy
+ action(CollectiveControlState, &ElevatorSystem):
(CollectiveControlState, ControlAction)
+ name(): String
CollectiveControlStat
+ direction:
Option<Direction>
+ default():
CollectiveControlState
AdaptedControlStrategy
+ action(CollectiveControlState, &ElevatorSystem):
(CollectiveControlState, ControlAction)
+ name(): String
<<Enumeration>>
Direction
Up
Down
Main
+ main()
+ run_single_simulation(&BuildingParameters,
&LiftParameters, & SimulationParameters,
&Vec<Vec<TrafficItem>>):
SimulationResults
+ passengers_in_traffic: usize
+ cargo_in_traffic: usize
+ stops: usize
+ passengers_delivered: usize
+ cargo_delivered: usize
+ waiting_ticks: Map<usize, usize>

e

te
e


Result<SimulationResults, String>
+ print_results(&Vec<SimulationResults>)
+ ride_ticks: Map<usize, usize>