

« interface » RoadSectionObserver

+ onSectionStateChanged(RoadSection, boolean): void

« interface » RoadSectionObservable

- addObserver(RoadSectionObserver): void
- removeObserver(RoadSectionObserver): void
- + notify(RoadSection, boolean): void

« interface » RoadNetworkObserver

+ onNumberOfVehiclesChanged(int, boolean): void

« enumeration » RoadType

- code: int
- + RoadType(int)
- + getCode(): int
- + valueOf(int): RoadType

Position

- + x: Integer + y: Integer
- + Position(Integer, Integer)

« interface » RoadNetworkObservable

- + addObserver(RoadNetworkObserver): void
- removeObserver(RoadNetworkObserver): void
- + notify(int): void

RoadNetwork

roadSections: RoadSection

- # lenathX: int
- # lengthY: int
- # active: boolean
- # interruptedImmediately: boolean # activeVehicles: List<Vehicle>
- # maxActiveVehicles: Integer
- # insertionRange: Integer
- # maxPossibleActiveVehicles: Integer
- # vehicleFactory: VehicleFactory
- # entryPoints: List<RoadSection> observers: List<RoadNetworkObserver>
- + RoadNetwork(int, int)
- + startSimulation(): void
- + stopSimulation(): void
- + immediatelyStopSimulation(): void
- + restartSimulation(): void
- + addVehicle(Vehicle): void
- + removeVehicle(Vehicle): void
- + getActiveVehicles(): List<Vehicle>
- setActiveVehicles(List<Vehicle>): void
- setMaxActiveVehicles(Integer): RoadNetwork
- getMaxActiveVehicles(): Integer
- + getLengthX(): int
- getLengthY(): int
- getMaxPossibleActiveVehicles(): Integer
- setMaxPossibleActiveVehicles(Integer): void
- isActive(): boolean
- + setActive(boolean): void
- isInterruptedImmediately(): boolean
- + setInterruptedImmediately(boolean): void
- + getRoadSections(): RoadSection[][]
- + setRoadSections(RoadSection[][]): void
- + getInsertionRange(): Integer
- + setInsertionRange(Integer): RoadNetwork + getRoadSectionAt(int, int): RoadSection
- + addEntryPoint(RoadSection): void
- + getEntryPoints(): List<RoadSection>
- + addObserver(RoadNetworkObserver): void removeObserver(RoadNetworkObserver): void
- + notify(int): void

« abstract » RoadSection

+ notify(RoadSection, boolean): void

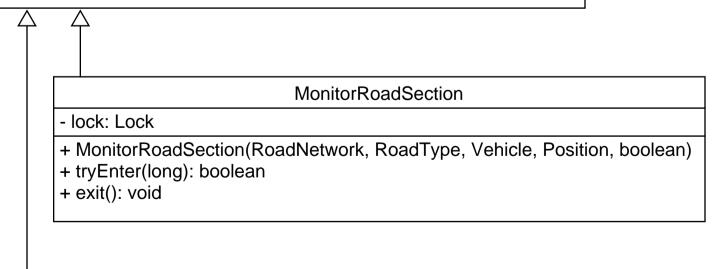
«interface»

RoadSectionObservable

+ addObserver(RoadSectionObserver): void

+ removeObserver(RoadSectionObserver): void

- observers: List<RoadSectionObserver>
- # roadNetwork: RoadNetwork
- # tvpe: RoadTvpe
- # imagePathStr: String
- # position: Position # isCrossroad: boolean
- # isExitPoint: boolean # isEntryPoint: boolean
- # vehicle: Vehicle
- pathStrategy: PathStrategy - possiblePaths: List<Position>
- + RoadSection(RoadNetwork, RoadType, Vehicle, Position, boolean)
- + abstract tryEnter(long): boolean
- + abstract exit(): void
- + forceExitCrossroad(): Position
- + getPossiblePaths(): List<Position>
- + getImageStrPath(): String
- + getType(): RoadType
- + getVehicle(): Vehicle
- + setVehicle(Vehicle): void
- getPosition(): Position + isCrossroad(): boolean
- + addObserver(RoadSectionObserver): void
- + removeObserver(RoadSectionObserver): void
- + notify(RoadSection, boolean): void



SemaphoreRoadSection

- + SemaphoreRoadSection(RoadNetwork, RoadType, Vehicle, Position, boolean)
- + tryEnter(long): boolean

semaphore: Semaphore

+ exit(): void

Thread

Vehicle

- # rand: Random
- # roadNetwork: RoadNetwork
- # roadSection: RoadSection
- # color: VehicleColor
- # sleepTime: Integer # imagePathStr: String
- # active: boolean + Vehicle(VehicleColor, Integer, RoadNetwork)
- + getImagePathStr(): String + getColor(): VehicleColor
- + getSleepTime(): Integer
- + getRoadSection(): RoadSection
- + setRoadSection(RoadSection): void
- + getRoadNetwork(): RoadNetwork
- + setRoadNetwork(RoadNetwork): void
- + isActive(): boolean
- + setActive(boolean): void
- + run(): void

«enumeration» VehicleColor

- code: int
- + BLACK: VehicleColor
- + BLUE: VehicleColor
- + BROWN: VehicleColor
- + CIAN: VehicleColor
- + GRAY: VehicleColor
- + GREEN: VehicleColor + ORANGE: VehicleColor
- + PINK: VehicleColor
- + PURPLE: VehicleColor
- + RED: VehicleColor
- + WHITE: VehicleColor + YELLOW: VehicleColor
- VehicleColor(code: int)
- + getCode(): int
- + valueOf(int): VehicleColor

RoadNetworkFactory

+ createRoadNetwork(String, ImplementationType): RoadNetwork getRoadSectionFactory (ImplementationType): RoadSectionFactory

« abstract » RoadSectionFactory + createRoadSection(RoadNetwork, RoadType, Vehicle, Position, boolean): RoadSection MonitorRoadSectionFactory + createRoadSection(RoadNetwork, RoadType, Vehicle, Position, boolean): RoadSection

SemaphoreRoadSectionFactory

+ createRoadSection(RoadNetwork, RoadType, Vehicle, Position, boolean): RoadSection

VehicleFactory

random: Random

+ createVehicle(RoadNetwork): Vehicle

getRandomColor(): VehicleColor # getRandomSleepTime(): Integer