**Diagram Description:**

1. **Car Class**
   * Inherits from Vehicle
   * Represents a car in the traffic simulation. It handles the specific behavior of cars, including their movement and color.
   * Attributes: Cell startCell, Color color
   * Methods: Car(Cell startCell, Color color), update(Grid grid), getColor() Color
2. **Vehicle Class**
   * Serves as a base class for all types of vehicles in the traffic simulation, including common attributes and methods.
   * Attributes: Cell startCell, Color color
   * Methods: Vehicle(Cell startCell, Color color), update(Grid grid), getColor() Color
3. **Cell Class**
   * Represents a cell in the traffic simulation grid, providing the basic properties and methods for different types of cells.
   * Attributes: int x, int y
   * Methods: Cell(int x, int y), getX() int, getY() int, update(), getColor() Color, toString() String
4. **Grid Class**
   * Represents the grid for the traffic simulation, consisting of RoadCell and IntersectionCell objects. Manages the layout and state of the grid.
   * Attributes: Cell[][] grid, int width, int height, List<Vehicle> vehicles, Random random
   * Methods: Grid(int width, int height, int trafficLightTiming), setTrafficLightTiming(int timing), addVehicle(Vehicle vehicle, int x, int y), updateGrid(), getGrid() Cell[][], getVehicles() List<Vehicle>, getWidth() int, getHeight() int, printGrid()
5. **IntersectionCell Class**
   * Inherits from Cell
   * Represents an intersection cell in the traffic simulation grid. Manages traffic light state and timing.
   * Attributes: boolean lightGreen, int timer, int timing
   * Methods: IntersectionCell(int x, int y, int timing), setTiming(int timing), changeLight(), update(), getColor() Color, toString() String, isLightGreen() boolean
6. **RoadCell Class**
   * Inherits from Cell
   * Represents a road cell in the traffic simulation. Road cells are traversable cells on which vehicles can move.
   * Methods: RoadCell(int x, int y), update(), getColor() Color, toString() String
7. **TrafficGUI Class**
   * Creates the graphical user interface for the traffic simulation. Includes controls for adjusting vehicle speed and traffic light timing, and displays the grid with vehicles and intersections.
   * Attributes: TrafficSimulation simulation, Timer timer, JSlider vehicleSpeedSlider, JSlider trafficLightTimingSlider, static Color BACKGROUND\_COLOR
   * Methods: TrafficGUI(TrafficSimulation simulation), initialize(), initializeTimer()
8. **TrafficMain Class**
   * Contains the main method to start the traffic simulation. It creates an instance of TrafficSimulation and TrafficGUI to run and display the simulation.
   * Methods: main(String[] args)
9. **TrafficSimulation Class**
   * Manages the traffic simulation. Initializes the grid, handles the state of the simulation, and manages the vehicles and traffic lights.
   * Attributes: Grid grid, boolean isRunning, int vehicleSpeed, int trafficLightTiming
   * Methods: TrafficSimulation(int width, int height, int trafficLightTiming), start(), pause(), reset(), update(), setVehicleSpeed(int speed), setTrafficLightTiming(int timing), getGrid() Grid, getVehicleSpeed() int, getTrafficLightTiming() int, initializeVehicles()

**Relationships:**

* Car inherits from Vehicle
* Cell is the superclass of IntersectionCell and RoadCell
* Grid has a composition relationship with Cell, Vehicle, IntersectionCell, and RoadCell
* TrafficGUI has a composition relationship with TrafficSimulation
* TrafficMain has a composition relationship with TrafficSimulation and TrafficGUI
* TrafficSimulation has a composition relationship with Grid, Vehicle, and Cell

**Connections:**

* Use inheritance arrows from Car to Vehicle, IntersectionCell to Cell, and RoadCell to Cell
* Use composition arrows to connect Grid with Cell, Vehicle, IntersectionCell, and RoadCell
* Use composition arrows to connect TrafficGUI with TrafficSimulation
* Use composition arrows to connect TrafficMain with TrafficSimulation and TrafficGUI
* Use composition arrows to connect TrafficSimulation with Grid, Vehicle, and Cell