Diagram Description:

1. **Car Class**
    - Inherits from Vehicle
    - Represents a car in the traffic simulation, handling the specific behavior of cars, including their movement and color.
    - Attributes:
        - `int direction`
    - Methods:
        - `Car(Cell initialCell, Color color)`
        - `int[] getNextMove()`
2. **Vehicle Class**
    - Serves as a base class for all types of vehicles in the traffic simulation, including common attributes and methods.
    - Attributes:
        - `Cell currentCell`
        - `Color color`
        - `int gridHeight`
    - Methods:
        - `Vehicle(Cell initialCell, Color color)`
        - `Cell getCurrentCell()`
        - `Color getColor()`
        - `void move(Cell newCell)`
        - `abstract int[] getNextMove()`
        - `void setGridHeight(int gridHeight)`
        - `int getGridHeight()`
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)`
        - `int getX()`
        - `int getY()`
        - `abstract void update()`
        - `abstract Color getColor()`
        - `void setTiming(int timing)`
        - `boolean canEnter(Vehicle vehicle)`
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`
        - `List<Vehicle> vehicles`
        - `int width`
        - `int height`
        - `int trafficLightTiming`
    - Methods:

        - `Grid(int width, int height, int trafficLightTiming)`
        - `void setTrafficLightTiming(int timing)`
        - `void addVehicle(Vehicle vehicle, int x, int y)`
        - `void updateGrid()`
        - `Cell[][] getGrid()`
        - `List<Vehicle> getVehicles()`
        - `int getWidth()`
        - `int getHeight()`
5. **IntersectionCell Class**
    - Inherits from Cell
    - Represents an intersection cell in the traffic simulation grid. Manages traffic light state and timing.
    - Attributes:
        - `int timing`
        - `int timer`
        - `boolean isGreen`
    - Methods:
        - `IntersectionCell(int x, int y, int timing)`
        - `void setTiming(int timing)`
        - `void update()`
        - `Color getColor()`
        - `boolean canEnter(Vehicle vehicle)`
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)`
        - `void update()`
        - `Color getColor()`
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`
        - `JTextField speedField`
        - `JTextField timingField`
        - `int defaultSpeed`
        - `int defaultTiming`
        - `int minSpeed`
        - `int maxSpeed`
        - `int minTiming`
        - `int maxTiming`
    - Methods:
        - `TrafficGUI(TrafficSimulation simulation)`
        - `void window()`
        - `void drawGrid(Graphics graphic)`
        - `void setTimer()`
        - `void resetSimulation()`

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:
        - `static void 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)`
        - `void start()`
        - `void pause()`
        - `void reset()`
        - `void update()`
        - `void setVehicleSpeed(int speed)`
        - `void setTrafficLightTiming(int timing)`
        - `Grid getGrid()`
        - `int getVehicleSpeed()`
        - `int getTrafficLightTiming()`

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**