Software engineering and Project management-Lab

Submitted by: Jyotiraditya Singh

Enroll Number: R2142220530

Semester: 4

SAP ID: 500107065

Aim: To create a class and object diagram for two wheeler and four wheeler vehicles.

- Class diagram

  A static structure diagram serves as a fundamental tool in software engineering, offering a graphical depiction of a system's underlying architecture.

  Within this diagram, the various classes constituting the system are meticulously delineated, accompanied by descriptions of their attributes and operations.

  Moreover, the diagram elucidates the intricate web of relationships among these classes, providing invaluable insights into the structural integrity and organization of the system. By presenting a holistic view of the system's components and their interactions, this diagram facilitates effective communication among stakeholders and aids in the comprehensive understanding and management of the software development process.

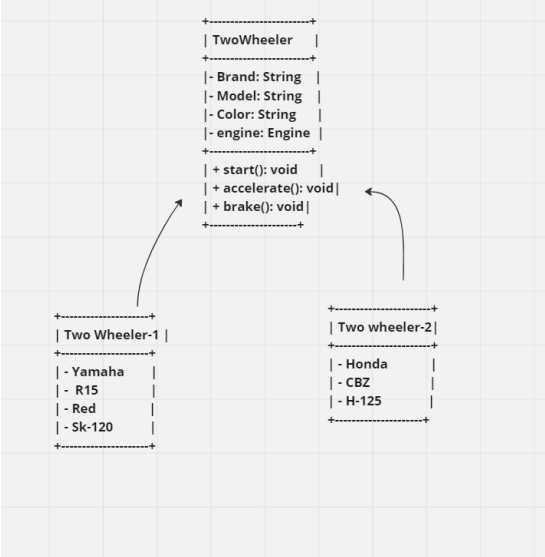
- Object diagram

  It is an instance of a class in particular moment in runtime that can have its own state and data values. Likewise, a static UML object diagram is an instance of a class diagram.

  It shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships which may be considered a special case diagram or a communication diagram.

UML diagram

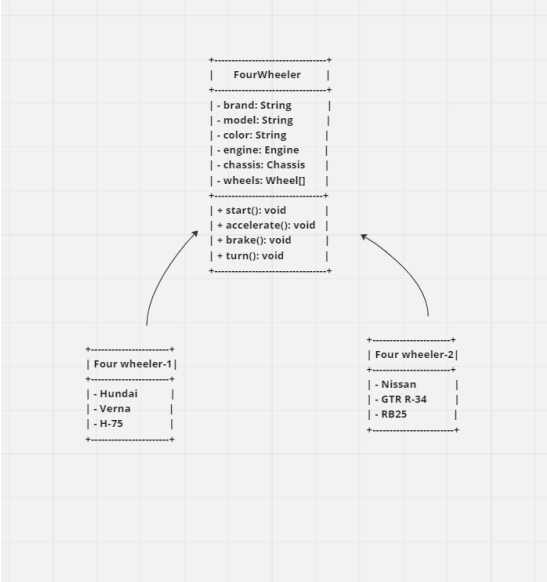
- Two wheelers:



About the above diagram

1. Class Name: TwoWheeler
    - Represents a two-wheeled vehicle, such as a motorcycle or scooter.
2. Attributes:
    - Brand: String: Indicates the brand or manufacturer of the two-wheeler.
    - Model: String: Specifies the model of the two-wheeler.
    - Color: String: Represents the color of the two-wheeler.
    - engine: Engine: Represents the engine component of the two-wheeler.
3. Methods:
    - start(): void: Initiates the engine to start the two-wheeler.
    - accelerate(): void: Causes the two-wheeler to accelerate.
    - brake(): void: Engages the brakes to slow down or stop the two-wheeler.
4. Functionality:
    - The class encapsulates basic functionalities and characteristics of a two-wheeled vehicle.
    - It can be instantiated to represent various specific types and instances of two-wheelers, each with its own brand, model, color, and engine.
    - The methods provide behavior for starting, accelerating, and braking, enabling the vehicle to operate on the road.
5. Composition:
    - Utilizes composition to include an Engine component as part of its structure.
    - This composition allows for encapsulation of the engine within the TwoWheeler, ensuring that each instance of the class has its own engine.

- Four wheelers:



About the above diagram

1. **Class Name**: FourWheeler
   - Represents a four-wheeled vehicle.
2. **Attributes**:
   - brand: String: Describes the brand or manufacturer of the four-wheeler.
   - model: String: Specifies the model of the four-wheeler.
   - color: String: Indicates the color of the four-wheeler.
   - engine: Engine: Represents the engine component of the four-wheeler.
   - chassis: Chassis: Represents the chassis component of the four-wheeler.
   - wheels: Wheel[]: An array of wheels that the four-wheeler possesses.
3. **Methods**:
   - start(): void: Initiates the engine to start the four-wheeler.
   - accelerate(): void: Causes the four-wheeler to accelerate.
   - brake(): void: Engages the brakes to slow down or stop the four-wheeler.
   - turn(): void: Executes a turning action, allowing the four-wheeler to change direction.
4. **Functionality**:
   - The class encapsulates basic functionalities and characteristics of a four-wheeled vehicle.
   - It can be instantiated to represent various specific types and instances of four-wheelers, each with its own brand, model, color, engine, chassis, and set of wheels.
   - The methods provide behavior for starting, accelerating, braking, and turning, enabling the vehicle to operate and maneuver on the road.
5. **Composition**:
   - Utilizes composition to include an Engine, Chassis, and an array of Wheel objects as part of its structure.

- This composition allows for encapsulation of these components within the FourWheeler, ensuring that each instance of the class has its own engine, chassis, and set of wheels.