

My UML class diagram represents a class hierarchy for a vehicle simulation system, where **Vehicle** serves as the abstract base class. It doesn't specify any attributes or methods, functioning purely as a foundational class for inheritance. **TwoWheeled**, which inherits from **Vehicle**, acts as a category for two-wheeled vehicles like bicycles but doesn't introduce additional attributes or behaviors. **Bicycle**, a concrete class derived from **TwoWheeled**, is equipped with attributes such as **gears**, **cost**, **weight**, and **color**, and features methods including multiple constructors, getters, setters (with method chaining), and **outputData** to manage and display bicycle details. This setup highlights key OOP principles such as inheritance and encapsulation, structuring the classes to maintain a clear and functional hierarchy.