UML Diagram Components

1. **Classes**:
   - Each class is represented as a rectangle divided into three sections: the top for the class name, the middle for attributes, and the bottom for methods.
2. **Abstract Classes**:
   - Abstract classes are depicted similar to regular classes but with the class name italicized.
3. **Inheritance**:
   - Inheritance is shown with a solid line and a closed arrow pointing from the subclass to the superclass.
4. **Interfaces**:
   - Interfaces are shown similar to classes but with the label `<<interface>>` above the interface name, and all methods are assumed to be abstract.
5. **Aggregation**:
   - Aggregation, indicating a whole-part relationship but with a non-exclusive ownership (parts can belong to multiple wholes), is depicted with a hollow diamond at the whole end connected to the part with a line.
6. **Composition**:
   - Composition, a strong whole-part relationship where the part cannot exist independently of the whole, is shown with a filled diamond at the whole end connected to the part with a line.

Example UML Diagram Structure

Here's how the UML diagram might look based on the given code:

- **TransportVehicle (Abstract Class)**
  - **Attributes**:
    - -vehicleID: String
    - -capacity: Int
  - **Methods**:
    - +calculateFare(): double
- **Bus (Class)**
  - **Inherits**: TransportVehicle
  - **Attributes**:
    - -FARE_PER_KM: double = 0.50 (static, final)
  - **Methods**:
    - +calculateFare(): double
- **Train (Class)**
  - **Inherits**: TransportVehicle
  - **Attributes**:
    - -FARE_PER_KM: double = 1.50 (static, final)
  - **Methods**:
    - +calculateFare(): double
- **Station (Class)**
  - **Attributes**:

        - -name: String
        - -location: String
    - **Methods**:
        - +getName(): String
        - +getLocation(): String
- **Route (Class)**
    - **Attributes**:
        - -stations: List<Station>
    - **Methods**:
        - +getStations(): List<Station>
- **Printable (Interface)**
    - **Methods**:
        - +printDetails(): void
- **Ticket (Class)**
    - **Implements**: Printable
    - **Attributes**:
        - -ticketID: String
        - -price: double
    - **Methods**:
        - +printDetails(): void

Relationships

- **Route** has an aggregation relationship with **Station** (Route contains multiple stations).
- **TransportVehicle** has composition relationships suggesting it might contain components not explicitly defined in the code sample provided (e.g., an Engine class could be added as part of TransportVehicle).