Advanced Multi-Resource Management

- You are tasked with building a **.NET 8 console application** that manages three types of resources:
  1. **PhysicalResource** – tangible goods (e.g., items in a warehouse).
  2. **DigitalResource** – intangible or digital assets (e.g., software licenses).
  3. **VehicleResource** – various vehicles that need tracking.

- Create a **.NET 8 Console Application**.
  - In your entry point (top-level statements or `Main` method), you'll demonstrate how each resource type is created, stored, and displayed.

- PhysicalResource
  - Define an **enum** `PhysicalCategory` for categorizing physical items (e.g., `Electronics`, `Furniture`, `Perishable`).
  - Include the following **required** fields:
    - `Name` : `string`
    - `Category` : `PhysicalCategory`
    - `Quantity` : `int`
    - `ExpirationDate` :`DateTime?`

  - DigitalResource
    - Define an **enum** (`DigitalType`) for categorizing digital items (e.g., `License`, `Document`, `Video`).

        - Include the following **required** fields:
            - `Title` : `string`
            - `ResourceId` : `string` (unique identifier)
            - `Type` : `DigitalType`
            - `LastAccessed` : `DateTime?`.

    - VehicleResource
        - Define an **enum** (e.g., `VehicleType`) for categorizing vehicles (e.g., `Car`, `Truck`, `Motorcycle`).
        - Include the following **required** fields:
            - `Model` : `string`
            - `Type` : `VehicleType`
            - `Year` : `int`
            - `LastInspection` : `DateTime?`.

    - **Custom Collection for Vehicles**
        - Create a dedicated class that manages a list or array of **VehicleResource** objects.
        - Provide methods (`Add`, `Remove`) to manage these vehicles.
        - Implement a **specialized approach** to iterating through the internal list of vehicles (in a certain order or with certain checks).

    - Modern C# 12+ Features
        - Use `required` properties or **primary constructors** to ensure essential data is provided at object creation.
        - Use **nullable value types** for optional fields (e.g., `DateTime?`).

- Leverage **enums** for clear, maintainable categories.

- Console Demonstration
  - In your Program (or top-level statements):
    1. **Instantiate** multiple `PhysicalResource` and `DigitalResource` objects, demonstrating both required and optional fields.
    2. **Create** your custom collection for vehicles, add several `VehicleResource` objects, optionally remove one, and **iterate** over them.
    3. **Print** or **log** relevant information to the console to confirm correct behavior.

- You must implement a **lookup table** for `DigitalResource` objects using a `Dictionary<string, DigitalResource>`.
  - The `Dictionary` key should be the `ResourceId` of the digital resource.
  - The value should be the `DigitalResource` object.
- Requirements:
  1. Add a new `DigitalResourceCollection` class to manage `DigitalResource` objects using a `Dictionary<string, DigitalResource>`.
  2. Provide methods for:
     - Adding a new resource to the dictionary (ensure no duplicate `ResourceId`s).
     - Removing a resource by its `ResourceId`.
     - Retrieving a resource by its `ResourceId`.
  3. Demonstrate the following in the console app:
     - Add at least two `DigitalResource` objects to the dictionary.

- Search for one resource by `ResourceId`.
- Try adding a resource with a duplicate `ResourceId` (and handle it gracefully).

- You need to maintain a collection of **unique vehicle types** (e.g., `Car`, `Truck`, `Motorcycle`) using a `HashSet<VehicleType>`.
- Requirements:

1. Add a `HashSet<VehicleType>` to the `VehicleCollection` class to track unique vehicle types automatically whenever a new vehicle is added.
2. Modify the `Add` method in `VehicleCollection`:
   - Add the vehicle's `Type` to the `HashSet`.
3. Provide a method in `VehicleCollection` to display all unique vehicle types (from the `HashSet`).
4. Demonstrate the following in the console app:
   - Add at least three vehicles to the `VehicleCollection`.
   - Print the list of unique vehicle types.