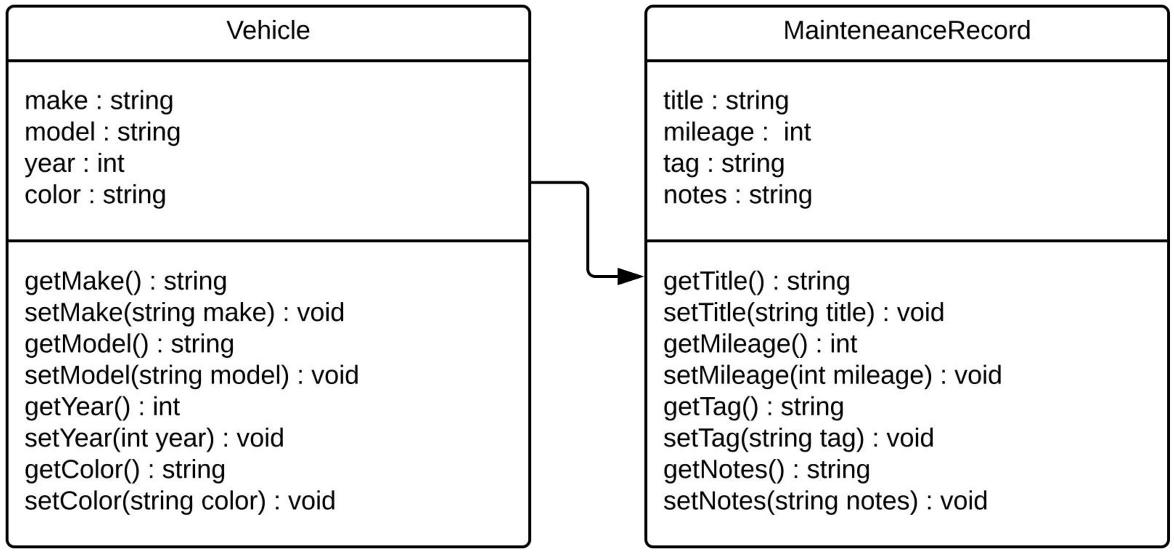
Title: Zach McKenna, Section A, Caliper

Problem Description: I will be creating a program to keep track of your automotive maintenance history. You will be able to create, edit, and destroy maintenance items. For example, Caliper can help keep track of item like oil changes, tire rotations, and part replacements. The program will store data in a .csv file that will be a helpful visual tool as well.

Data Description:

Caliper

Zachary McKenna | November 6, 2019



There are two major classes in Caliper, the Vehicle and the MaintenanceRecord class. The Vehicle class is the parent and the MaintenanceRecord class is the child, the Vehicle has many MainteneanceRecords. The user will be able to create Vehicles and in turn create MaintenanceRecord for those vehicles. This will be the backbone of the application, Vehicles and MaintenanceRecords.

Procedural Description:

Ask user for input: Manage Vehicles (create, edit, destroy) or Select Vehicle

If user chooses Manage Vehicles allow user to create new vehicle, edit details of an existing vehicle, or delete a vehicle

If user chooses select vehicle, get vehicles from the data.csv and list the vehicles for the user to choose.

When the user chooses a vehicle, list the maintenance history in ascending order (Newest at the top) and allow the user to create new entries, edit the details of old entries, delete entries, or go back to main menu.

When the user creates a maintenance record, they are able to set the attributes of the entry (title, mileage, tag, and notes).

Special Needs or Concerns: Storing the vehicle data may be tricky without use of a formal database such as PostgreSQL or MySQL since the data won't be as nicely formatted. This may end up making it difficult to make the data both easy to parse and easy to be read from the .csv by the user. It might not end up being practical to create a visually usable .csv file, and just create one that is easily parsable.