Technical Concerns and Decisions

The code defines classes to represent different types of vehicles (Car, Truck, Yacht, Motorcycle) and their engines, which is a good approach for modeling real-world entities. The decision to use object-oriented programming (OOP) principles for modeling vehicles and engines allows for encapsulation, inheritance, and polymorphism, making the code more modular and extensible.

Each vehicle type (Car, Truck, Yacht, Motorcycle) is implemented as a subclass of the Vehicle class. While this promotes code reuse and maintains a hierarchical structure. Also, there is a relation of composition between class Engine and Vehicle because it is necessary the class Engine in order to function in a good way.

Class Engine was defined with the attributes given by the requirements, as like the class Vehicle. Nevertheless, with the subclasses of Vehicle were added some attributes in each one of them, in order to do the exercise more realistic and because in the requirements said that could be added some properties.

The code provides a basic command-line interface (CLI) for users to interact with the platform. If the user writes a wrong data there will be set as none or 0, in order to avoid problems and crashing in the code.

Finally, the code contains was documented the aim of explain the code to the person that read it.