Diagram

Description automatically generated

The UML diagram that I created above represents the code that I viewed in Eclipse. The diagram represents four classes with their associated methods, variables, and relationships. The Bicycle class has private variables (annotated with a - sign) and public variables (annotated with a + sign). Private variables dictate that these variables can not be accessed by other classes or functions besides the Bicycle class. This would be a great example of the object-oriented programming principle of encapsulation. Public variables are accessible by other classes when called upon. The public variables are examples of the getter/setter methodology, which allows other classes to use this data.

The open arrows between Vehicle, TwoWheeled, and Bicycle represent the object-oriented programming principle of inheritance. Despite there being no properties, methods, or variables in the classes Vehicle and TwoWheeled, eventually Bicycle will use the principle of inheritance to utilize attributes of those two classes. The Bicycle class will be inheriting attributes and methods from Vehicle and TwoWheeled classes.

The Bicycle class also exhibits the objected-oriented programming principle of polymorphism with the use of its constructors. This will allow us to perform a single action in many ways. This is evident by the inheritance shown in the diagram.

Lastly, while not explicitly seen in the code I analyzed or in the UML diagram, abstraction could be utilized in the program. Anyone utilizing the program may only need access to essential details of the class.