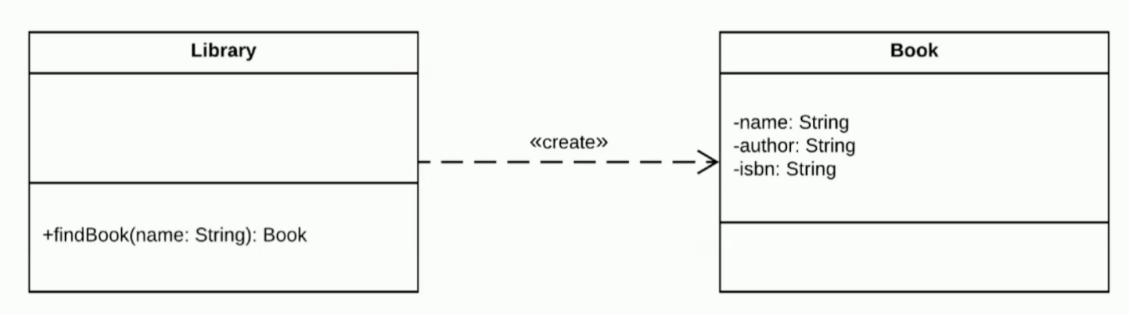
**UML Class Diagram**

1. **Relationships:**
2. **Dependency:**

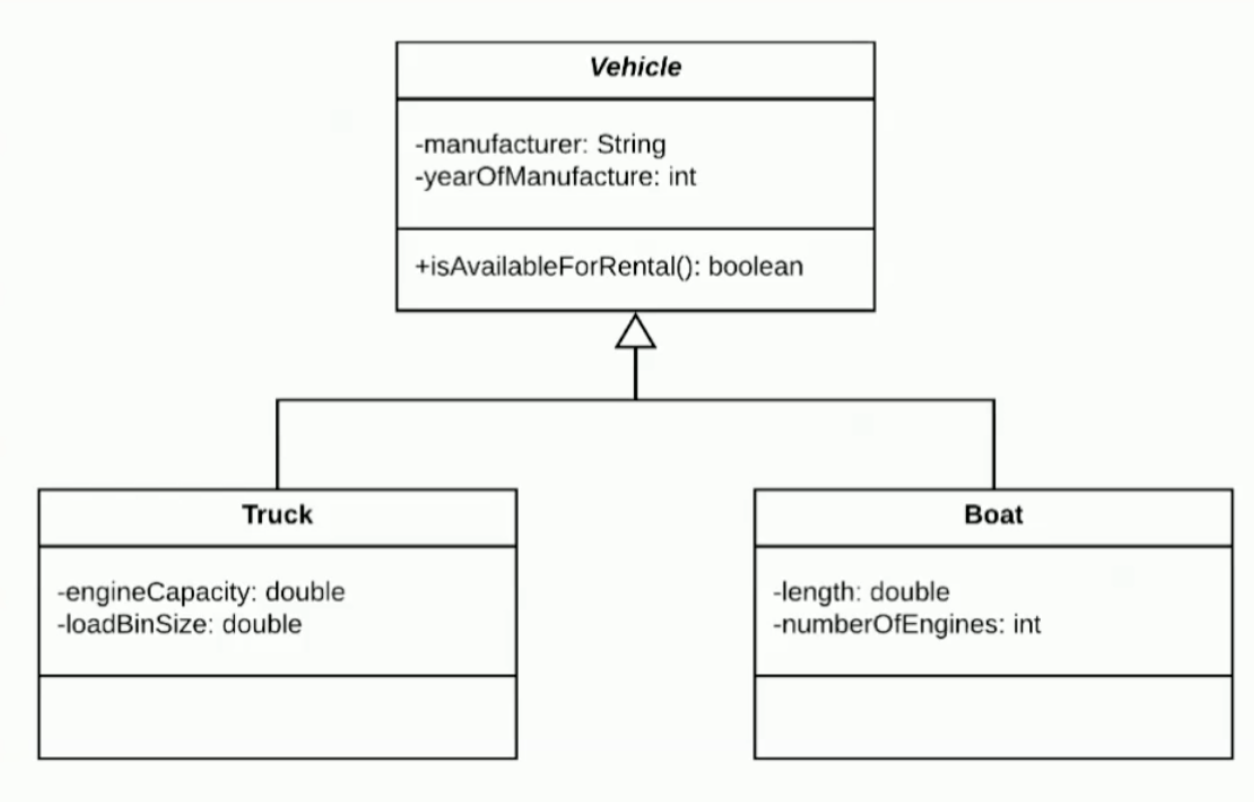
- Class A use instance of class B as a parameter in methods of Class A.

- Class A uses or depends on class B. If B is changed, it will affect to A.



1. **Generalization (Inheritance):**

- Class B inherits from Class A.



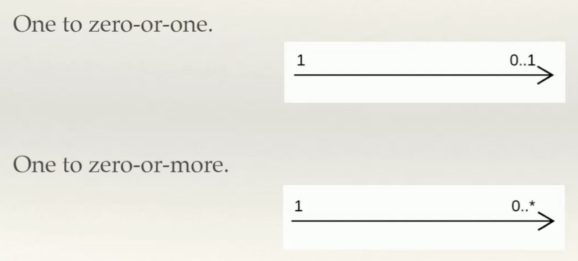
1. **Association:**

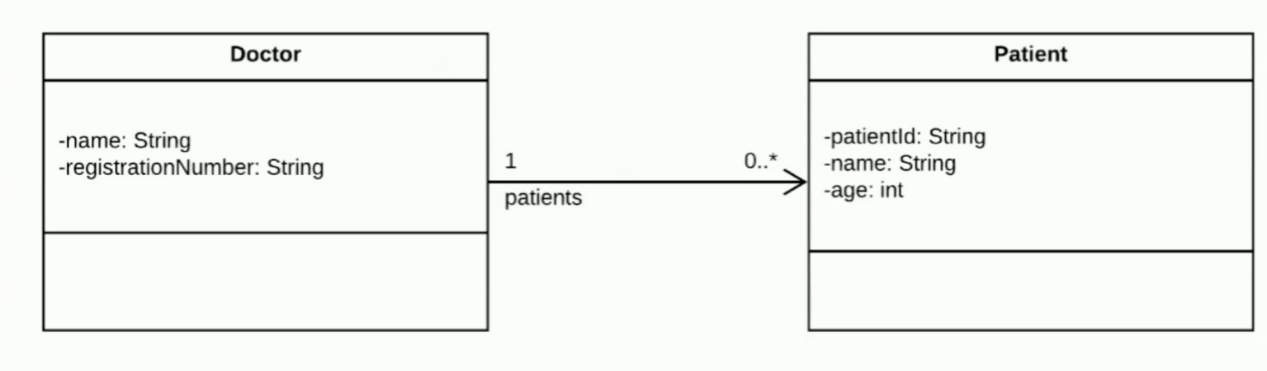
- class A has variables of class B. A and B will communicate with each other.

- each side of the association may include one or more instance of class in other side.

- Comparing Association & Dependency:

+ in association, A and B communicate with each other.

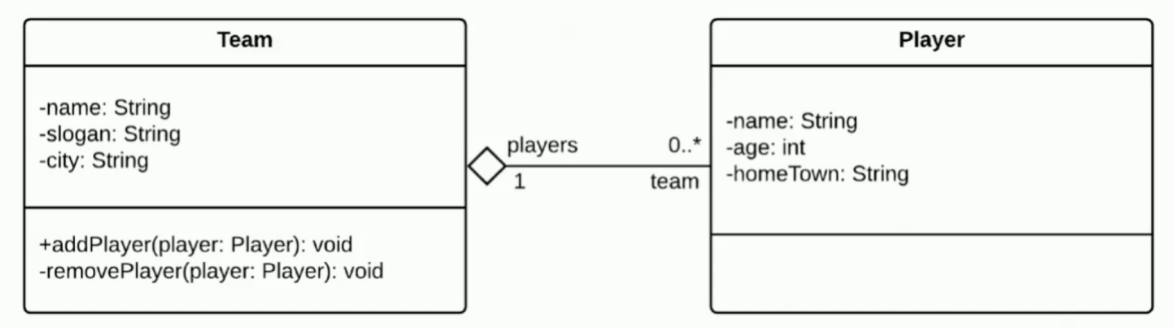




1. **Aggregation:**

- Class (owns) combine instance of other classes (child). Instances child classes can exist without an instance of owns class. It means that instances of child class is not initiated in instance of owns class.

- owns class has instances of child class and child class has the reference to the owns class.



1. **Composition:**

- Composition is a bi-directional relationship

- Class A composes objects of Class B.

- Class A cannot exist without objects of Class B, and Class B is not exist without object of Class A

