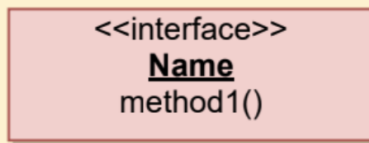
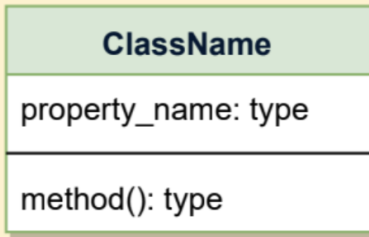


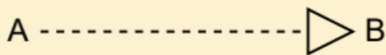
UML Conventions



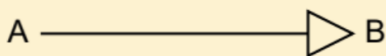
Interface: Classes implement interfaces, denoted by Generalization.



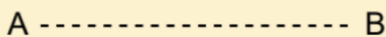
Class: Every class can have properties and methods.
Abstract classes are identified by their *Italic* names.



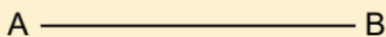
Generalization: A implements B.



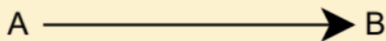
Inheritance: A inherits from B. A "is-a" B.



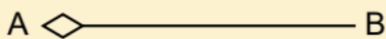
Use Interface: A uses interface B.



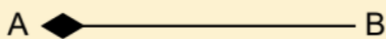
Association: A and B call each other.



Uni-directional Association: A can call B, but not vice versa.



Aggregation: A "has-an" instance of B. B can exist without A.



Composition: A "has-an" instance of B. B cannot exist without A.

Relationships between Classes

- **Composition:** A strong relationship where one class (the whole) contains another class (the part), and the part cannot exist without the whole.
- **Aggregation:** A weaker relationship than composition, where one class is associated with another, but the associated class can exist independently.
- **Association (Unidirectional):** A relationship where one class is aware of another, allowing it to interact, but the reverse is not necessarily true.
- **Association (Bidirectional):** A mutual relationship where both classes are aware of each other, enabling interaction in both directions.
- **Inheritance:** A relationship where a subclass inherits attributes and behaviors from a superclass, facilitating code reuse and hierarchy.
- **Implementation:** When a class adheres to an interface or abstract class's contract, it must provide specific behaviors defined by that contract.