


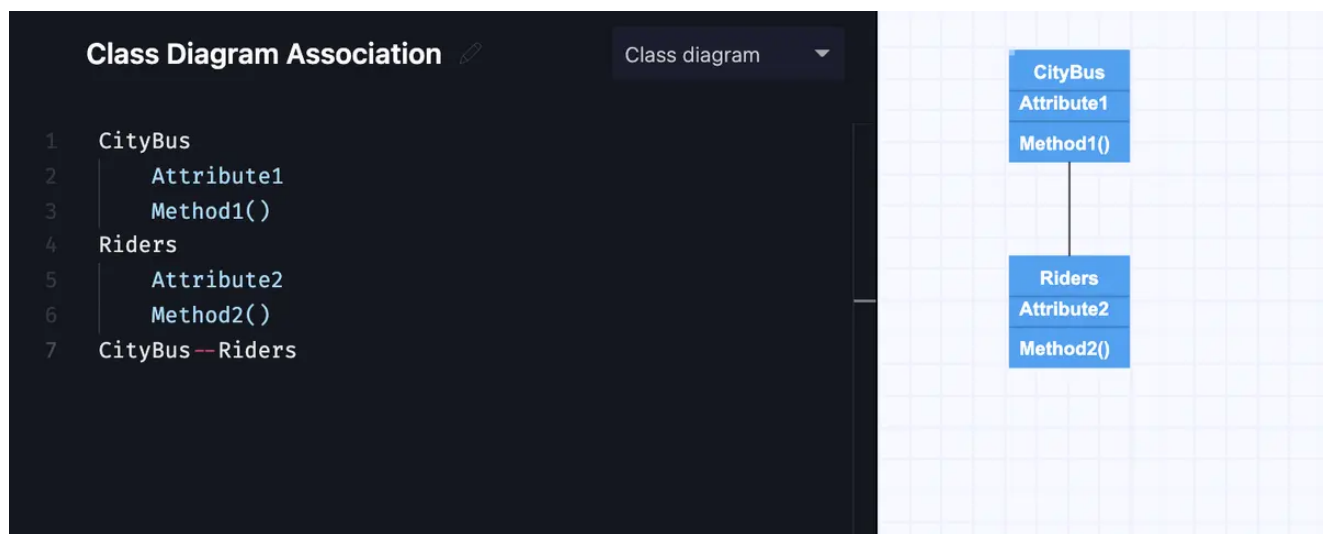
UML class diagram arrow types | Gleek

 gleek.io/blog/class-diagram-arrows

What are class diagram relationships?

Association

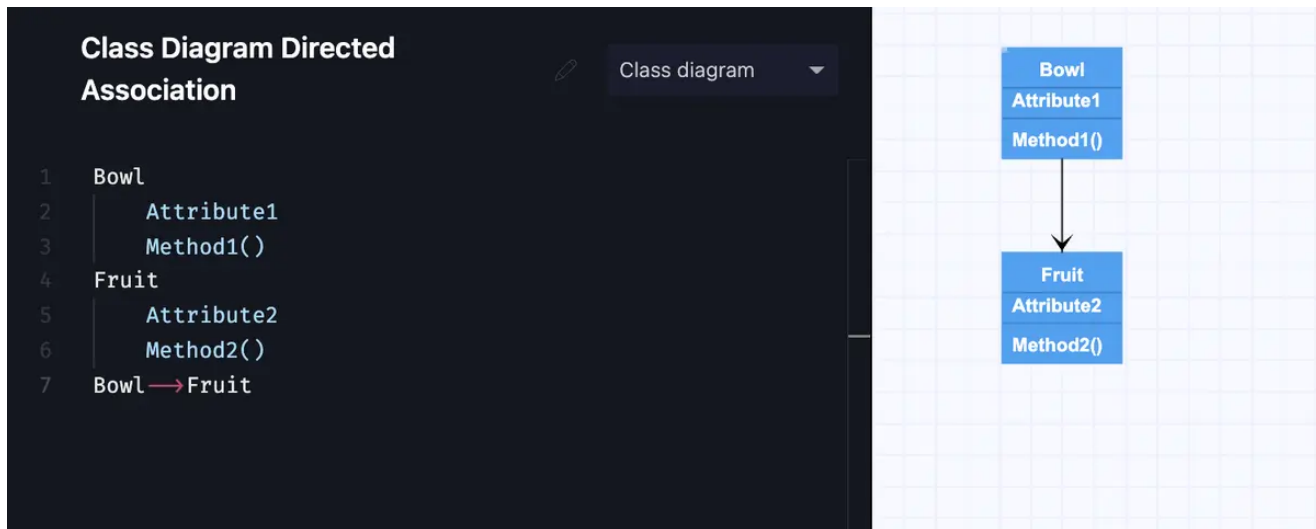
Association is the most basic of relationships. Association means any type of relationship or connection between classes. For example, we show a direct link between a city bus and its riders using an association line. We show a simple association with a straight line. In gleek.io we create this by typing two hyphens: —



There are several sub-types of association.

Directed association

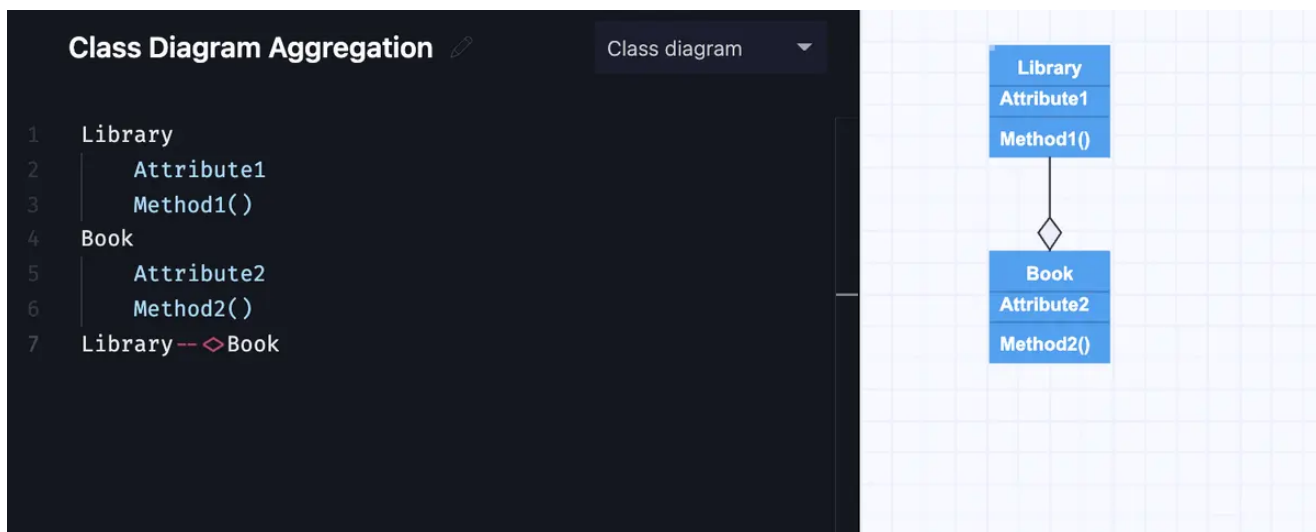
Directed association shows a strong relationship between classes. The classes must communicate. We represent a directed association with an arrow pointing to our object class. For example, a bowl might contain fruit. The bowl acts as a container class for the fruit class. In gleek.io we create this association with two hyphens and a greater-than symbol. The syntax looks like this: —>



Make your own UML class diagram with Gleek.

Aggregation

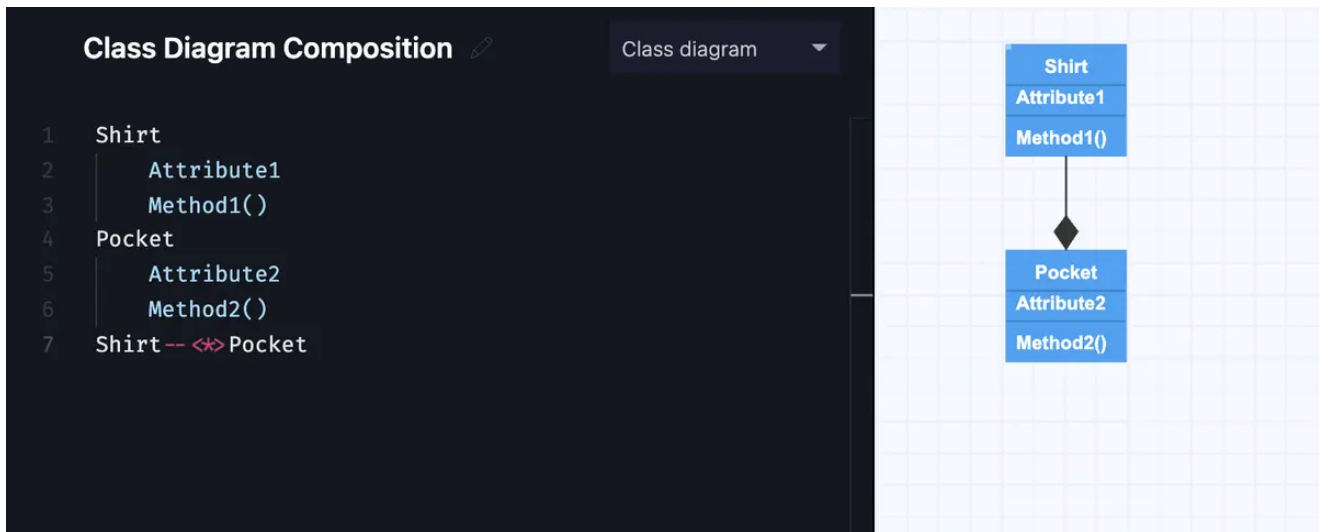
We use aggregation arrows when we want to convey that two classes are associated, but not as close as in direct association. The child class can exist independent of the parent element. For example, a book still exists if somebody checks it out from the library. In gleek.io we create aggregation arrows by typing two hyphens followed by a lesser-than symbol followed by a greater-than symbol. The syntax looks like this: `-->`



Composition

Composition arrows show up in UML class diagrams when we want to show a similar association to aggregation, with a key difference. Composition associations show relationships where the sub-object exists only as long as the container class exists. The classes have a common lifecycle. For example, a pocket on the front of a shirt cannot exist if

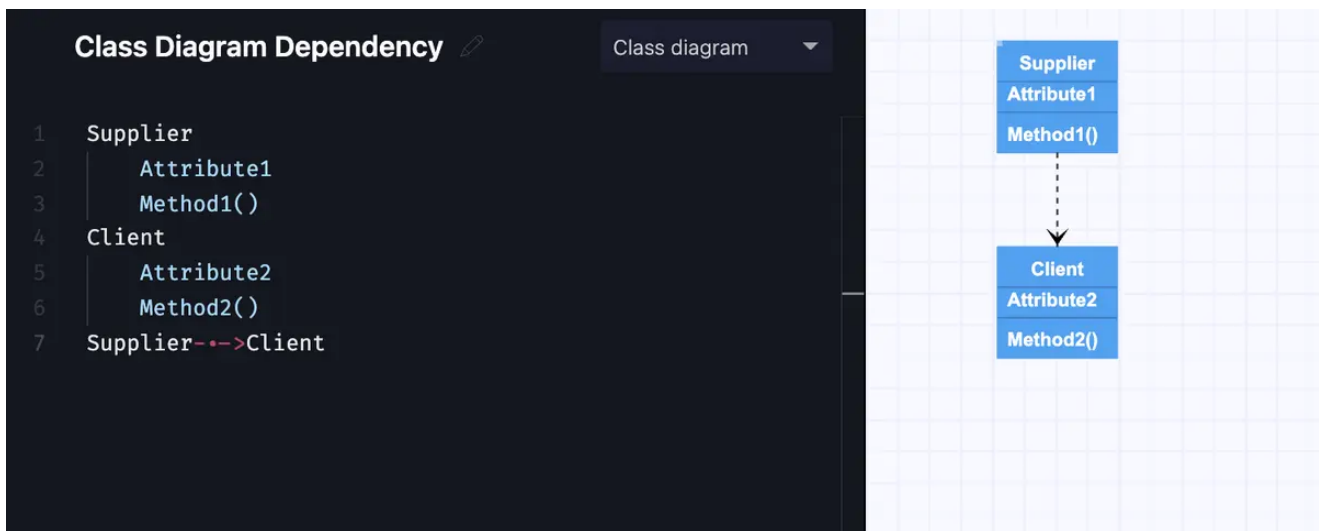
we destroy the shirt. In gleek.io we create a composition arrow by typing two hyphens followed by a star inside a lesser-than and greater-than symbol. The syntax looks like this: —<*>



Make your own UML class diagram with Gleek.

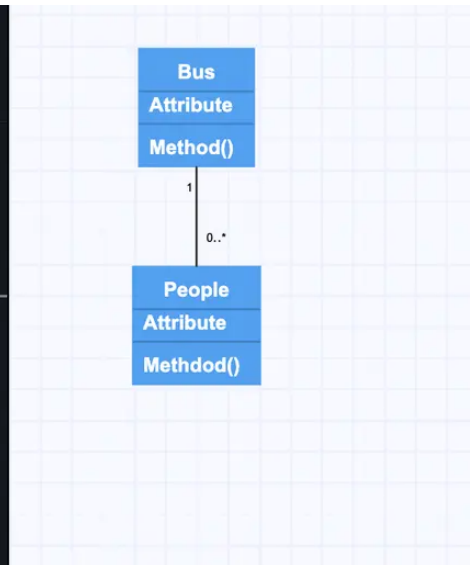
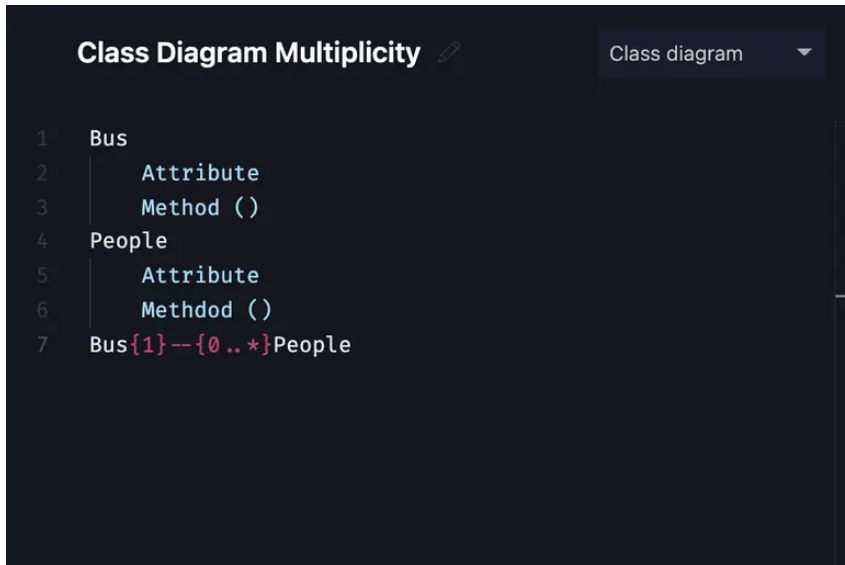
Dependency

Dependency arrows show us where two elements depend on each other, but in a less strong relationship than a basic association. Changes to the parent class will also affect the child class. Dependency shows a supplier-client type of relationship. In gleek.io we create a dependency arrow with a hyphen, a period followed by another hyphen, and a greater-than symbol. Our syntax will look like this: `-.->`



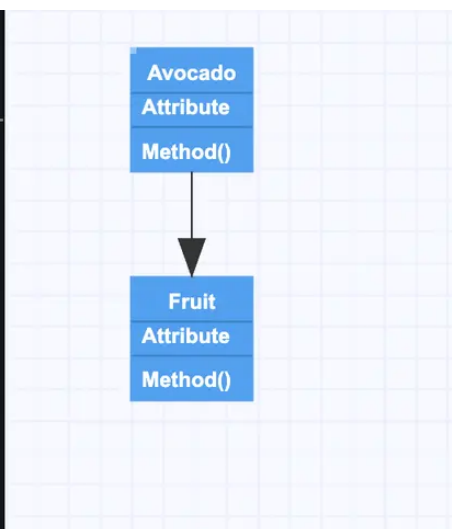
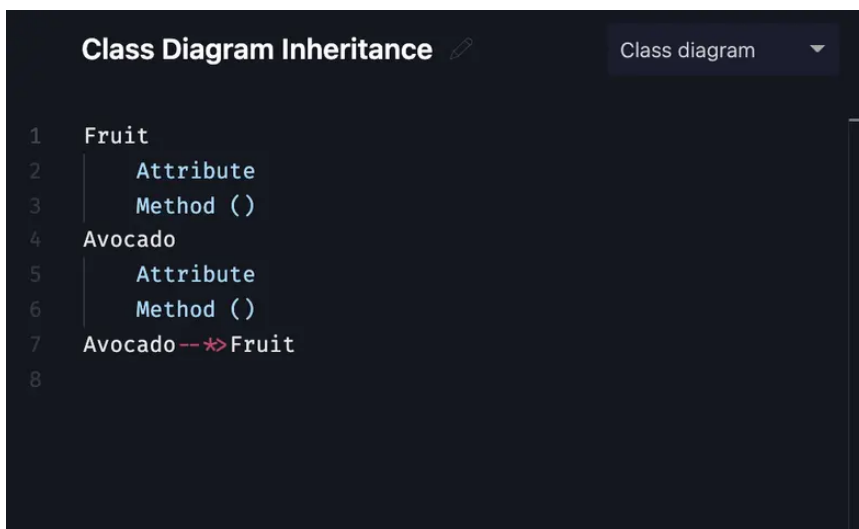
Multiplicity

Multiplicity or cardinality arrows show a place in our UML diagram where a class might contain many (or none!) items. For example, a city bus might have any number of riders at a given time. People constantly get on and off as the bus moves through the streets. We show this in our diagram with the notation 0..* meaning our class might contain zero to many objects. In gleek.io we create multiplicity with numbers inside curly brackets with two hyphens in the middle. Our syntax looks like this: **{1}--{0..*}**



Inheritance

We use Inheritance arrows to show a child class inherits functionality from the parent class. For example, an avocado is a type of fruit. Fruit is the super-class. Avocado is the sub-class. The avocado inherits its fruitiness from its fruit parent. To show inheritance in our UML class diagram in gleek.io, we type two hyphens followed by a star and a greater-than symbol. Our syntax will look like this: **--*>**



Realization/Implementation

We use realization or implementation arrows to indicate a place where one class implements the function defined in another class. For example, the printer setup interface sets the printing preferences that are being implemented by the printer. The arrangement shows a realization association. To show the relation in gleeek.io, we type a hyphen, a period followed by another hyphen, a star symbol, and a greater-than symbol. Our syntax will look like this:

`-.-*>`

