

Relationships and their representation in a class diagram. Inheritance of class attributes and operations.



Agenda

Introduction

Building confidence

Engaging the audience

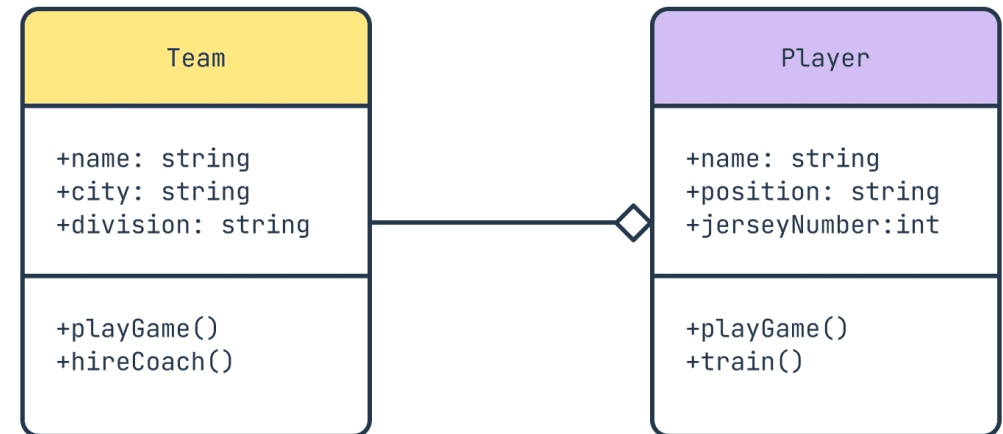
Visual aids


Final tips & takeaways

Class diagram


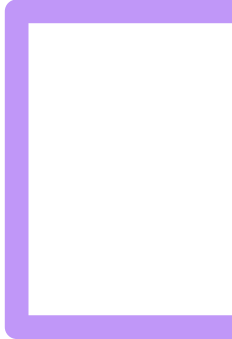
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

- classes,
- their attributes,
- operations (or methods),
- and the relationships among objects.



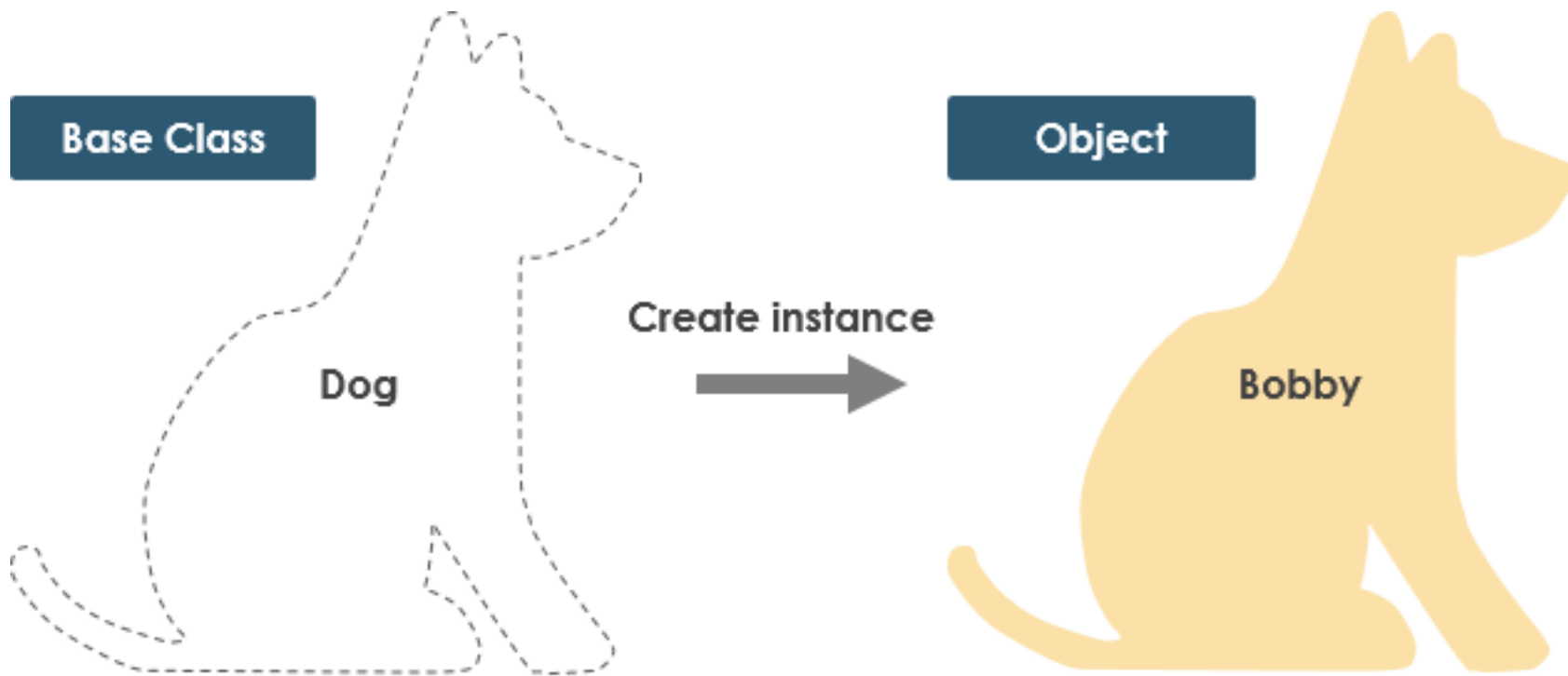


Class diagrams offer several benefits for any organization. Use UML class diagrams to:

- Illustrate data models for information systems, no matter how simple or complex.
 - Better understand the general overview of the schematics of an application.
 - Visually express any specific needs of a system and disseminate that information throughout the business.
 - Create detailed charts that highlight any specific code needed to be programmed and implemented to the described structure.
 - Provide an implementation-independent description of types used in a system that are later passed between its components.
- 
- 

Class

- A Class is a blueprint for an object. Objects and classes go hand in hand. We can't talk about one without talking about the other. And the entire point of Object-Oriented Design is not about objects, it's about classes, because we use classes to create objects. So, a class describes what an object will be, but it isn't the object itself.
- In fact, classes describe the type of objects, while objects are usable instances of classes. Each Object was built from the same set of blueprints and therefore contains the same components (properties and methods). The standard meaning is that an object is an instance of a class and object - Objects have states and behaviors.



Properties

Color
Eye Color
Height
Length
Weight

Methods

Sit
Lay Down
Shake
Come

Property Values

Color: Yellow
Eye Color: Brown
Height: 17 in
Length: 35 in
Weight: 24 pounds

Methods

Sit
Lay Down
Shake
Come

Example

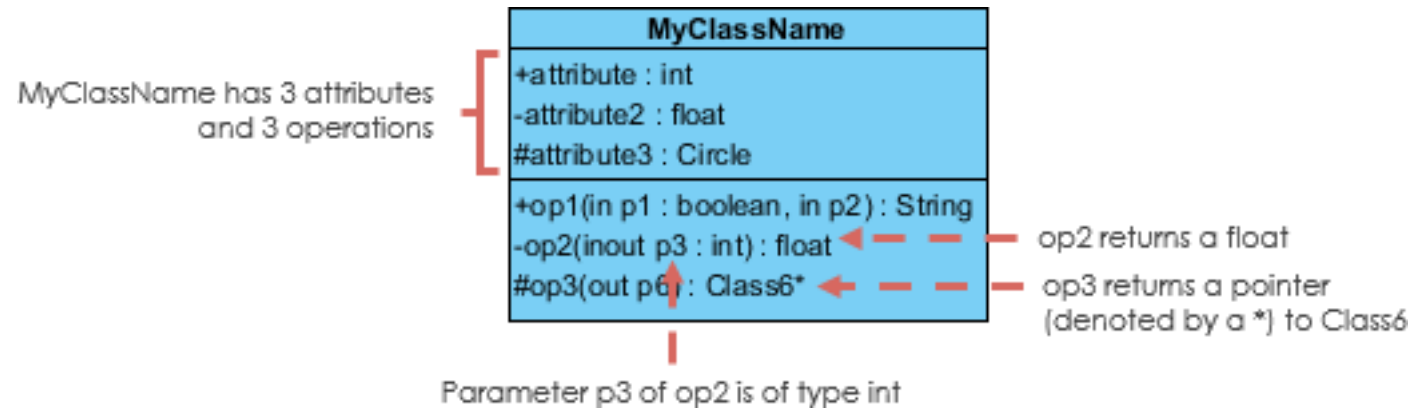
Basic components of a class diagram

- **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
- **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
- **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.



Class Operations (Methods):

- The return type of a method is shown after the colon at the end of the method signature.
- The return type of method parameters are shown after the colon following the parameter name. Operations map onto class methods in code



Class Visibility

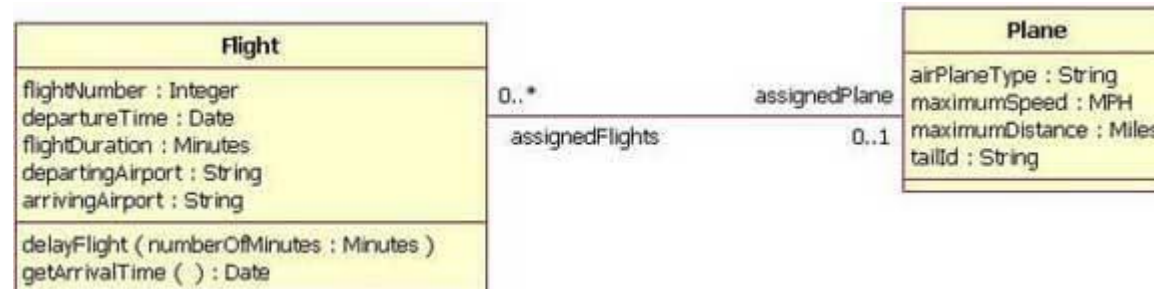
- + denotes public attributes or operations
- - denotes private attributes or operations
- # denotes protected attributes or operations

Relationships between classes



Associations

- An association represents a bi-directional relationship between two classes. It indicates that instances of one class are connected to instances of another class.
- They are represented by a solid line between classes. Associations are typically named using a verb or verb phrase which reflects the real-world problem domain. Associations are always assumed to be bi-directional; this means that both classes are aware of each other and their relationship, unless you qualify the association as some other type.



Uni-directional association

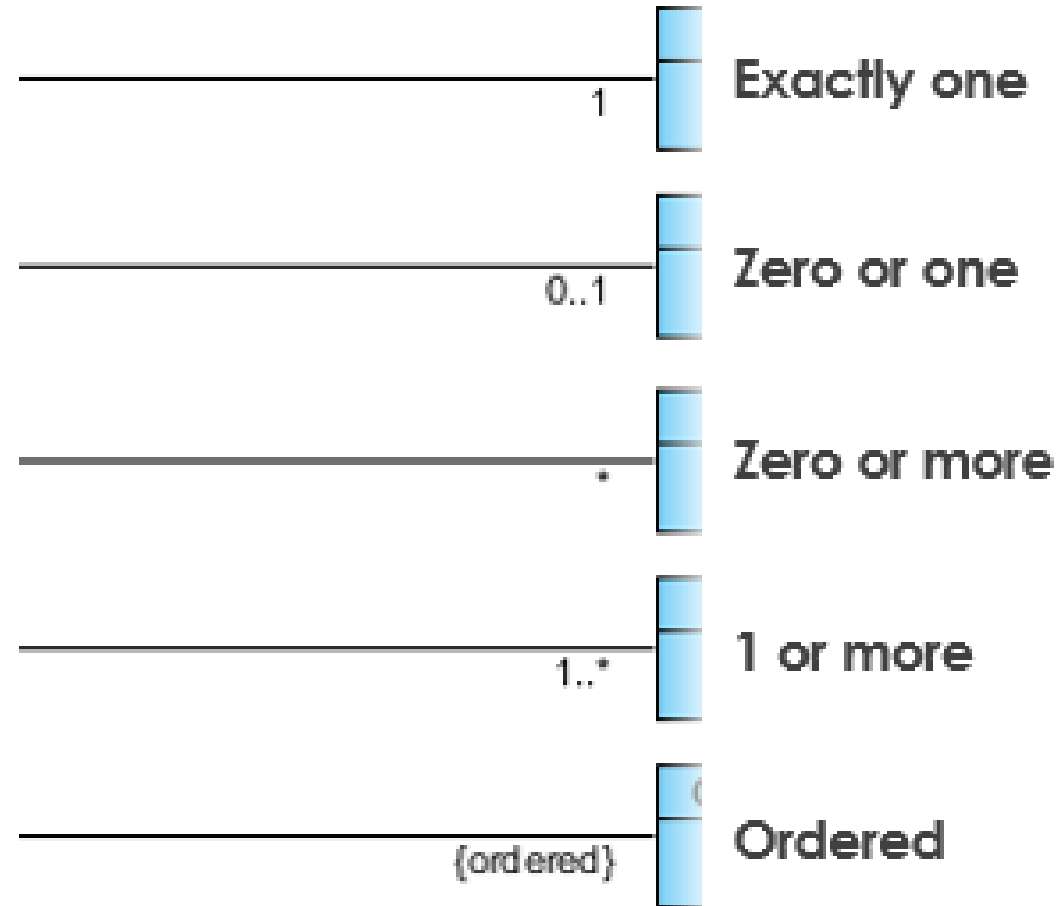
- In a uni-directional association, two classes are related, but only one class knows that the relationship exists.



Cardinality

Cardinality is expressed in terms of:

- one to one
- one to many
- many to many



Inheritance

- Refers to the ability of one class (child class) to *inherit* the identical functionality of another class (super class), and then add new functionality of its own. (In a non-technical sense, imagine that I inherited my mother's general musical abilities, but in my family I'm the only one who plays electric guitar.) To model inheritance on a class diagram, a solid line is drawn from the child class (the class inheriting the behavior) with a closed, unfilled arrowhead (or triangle) pointing to the super class.

Aggregation

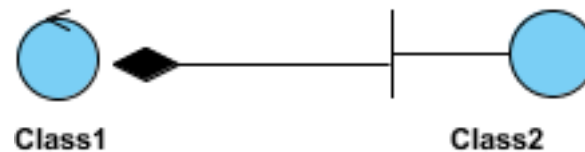
- It represents a "part of" relationship.
- Class2 is part of Class1.
- Many instances (denoted by the *) of Class2 can be associated with Class1.
- Objects of Class1 and Class2 have separate lifetimes.
- Aggregation is represented by a diamond shape on the side of the whole class. In this kind of relationship, the child class can exist independently of its parent class.

Composition

A special type of aggregation where parts are destroyed when the whole is destroyed.

- Objects of Class2 live and die with Class1.
- Class2 cannot stand by itself.

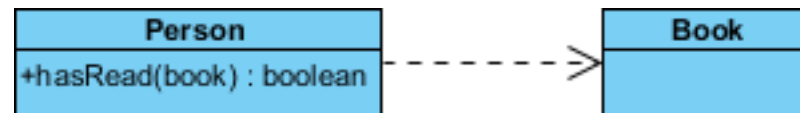
The relationship is displayed as a solid line with a filled diamond at the association end, which is connected to the class that represents the whole or composite.



Dependency

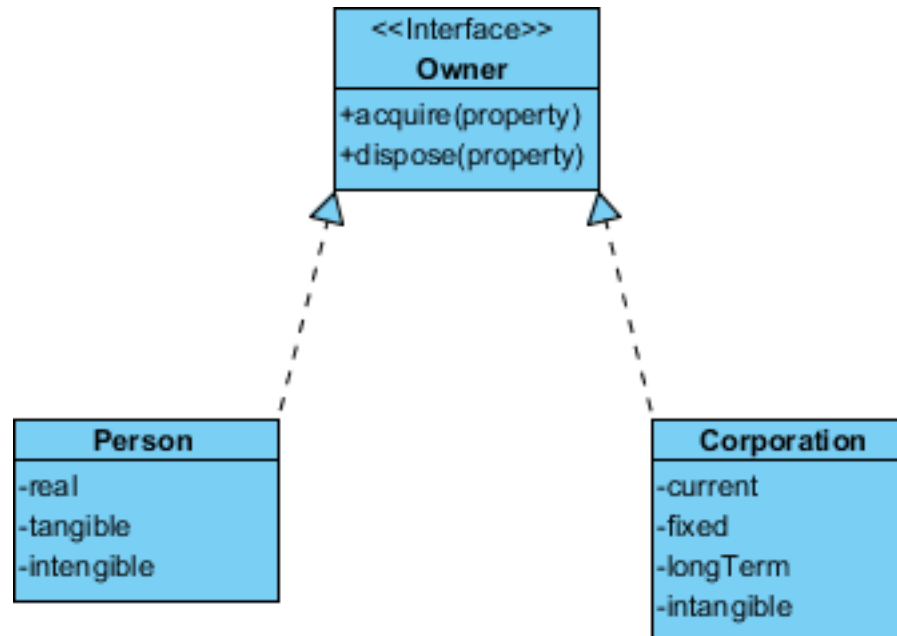
An object of one class might use an object of another class in the code of a method. If the object is not stored in any field, then this is modeled as a dependency relationship.

- A special type of association.
- Exists between two classes if changes to the definition of one may cause changes to the other (but not the other way around).
- Class1 depends on Class2



Realization

- Realization is a relationship between the blueprint class and the object containing its respective implementation level details. This object is said to realize the blueprint class. In other words, you can understand this as the relationship between the interface and the implementing class.



A large blue circle is centered on the left side of the slide. To its right, there are orange lines forming a partial square and a triangle. In the top-left corner, there is a purple circle. In the bottom-left corner, there is a green circle. In the bottom-center, there is a blue circle.

Thank you

Brita Tamm

502-555-0152

brita@firstupconsultants.com

www.firstupconsultants.com