

# Class diagram

## Overview

A Class diagram falls under the structural diagramming family. It is a graphic representation of the static structural model. It shows [classes](#) and [interfaces](#), along with their internal structure and relationships. The classes represent types of objects that are handled in a system. The instances of those types (objects) are instantiated only on the runtime and are represented by an object and the interaction diagrams. A Class diagram does not show temporal information, it describes only the classification.

Classes can be related to each other in a number of ways: associated (connected to each other), dependent (one class depends/uses another class), specialized (one class is a subtype of another class), or packaged (grouped together as a unit – package). A Class diagram does not express anything specific about the relationships of a given object, but it does abstractly describe the potential relationships of one object with other objects.

A system typically has a number of Class diagrams – not all the classes are inserted into a single Class diagram. A class can have multiple levels of meaning and participate in several Class diagrams. A Class diagram is the logical map of an existing or future source code.

## Purpose

A class diagram describes the types of objects (e.g., people, materials, information, and behaviors) in the system and various kinds of static relationships that exist among them. Class diagrams also show the attributes and operations of a class and the constraints that apply to the way objects are connected.

## Usage

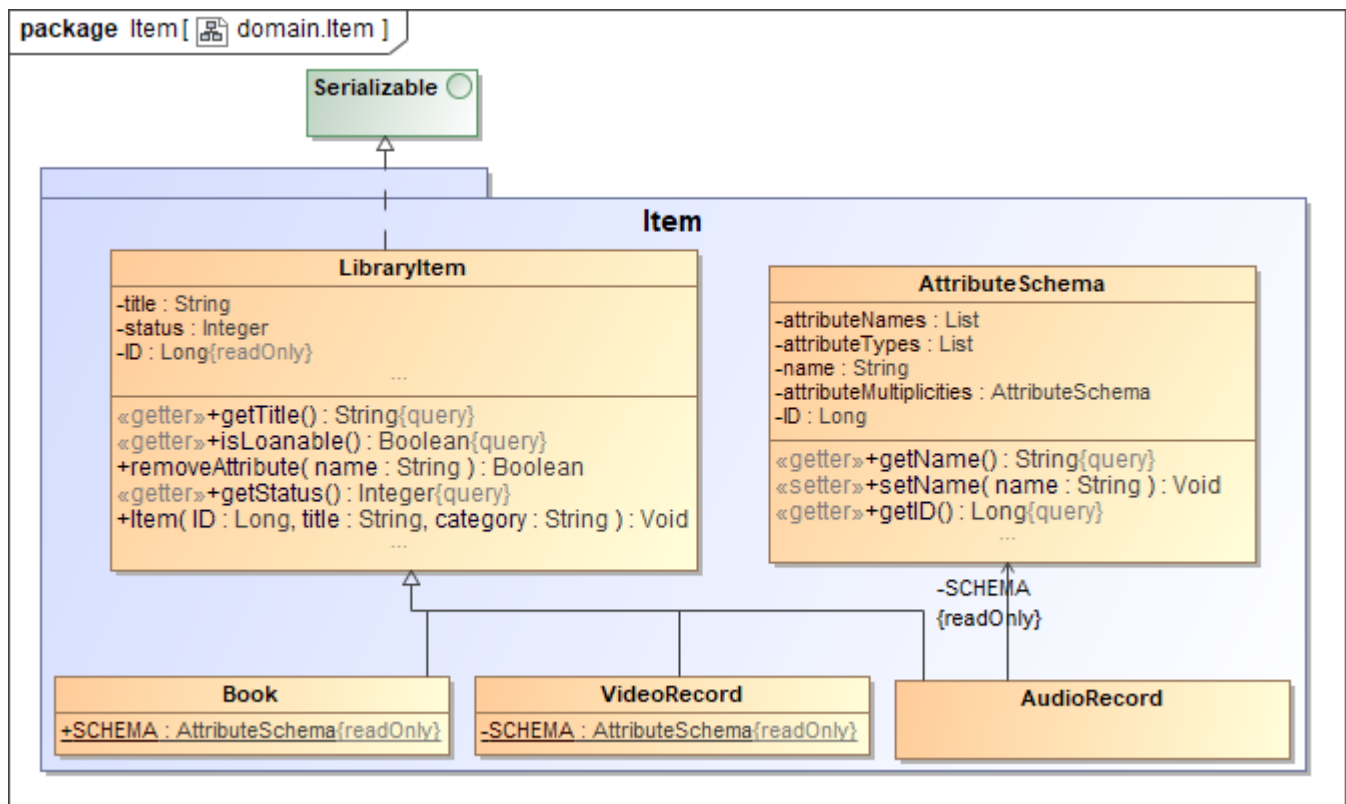
A class diagram can be used to:

- illustrate both simple and complex models for information systems
- understand the general overview of the schematics of an application
- visually express the specific needs of a system
- highlight the specific code that needs to be programmed and implemented to the structure
- provide an implementation-independent description of types used in a system
- share relevant information with stakeholders
- detect errors early in the lifecycle.

## Summary:

Class diagrams are valuable because they:

- define the essential resources of the system
- define relationships between resources
- generate code
- model code (reverse engineering)
- provide a focus for all the other diagrams.



Example of a Class diagram

**Related pages**

- [Creating diagrams](#)
- [Class](#)