UML

Class Diagram

1. Used to model the type of objects and relationships among them
2. One of the most commonly used UML diagram
3. Drawn by developers as a part of design activity



**Classifier**

An abstract metaclass whose concrete subclasses classify different types of values

* Concrete Class
* Abstract Class
* Interface
* Enumerations
* Generic Class

**Feature**

Structural and behavioral characteristics of a classifier

* Structural

Properties or attributes

* Behavioral

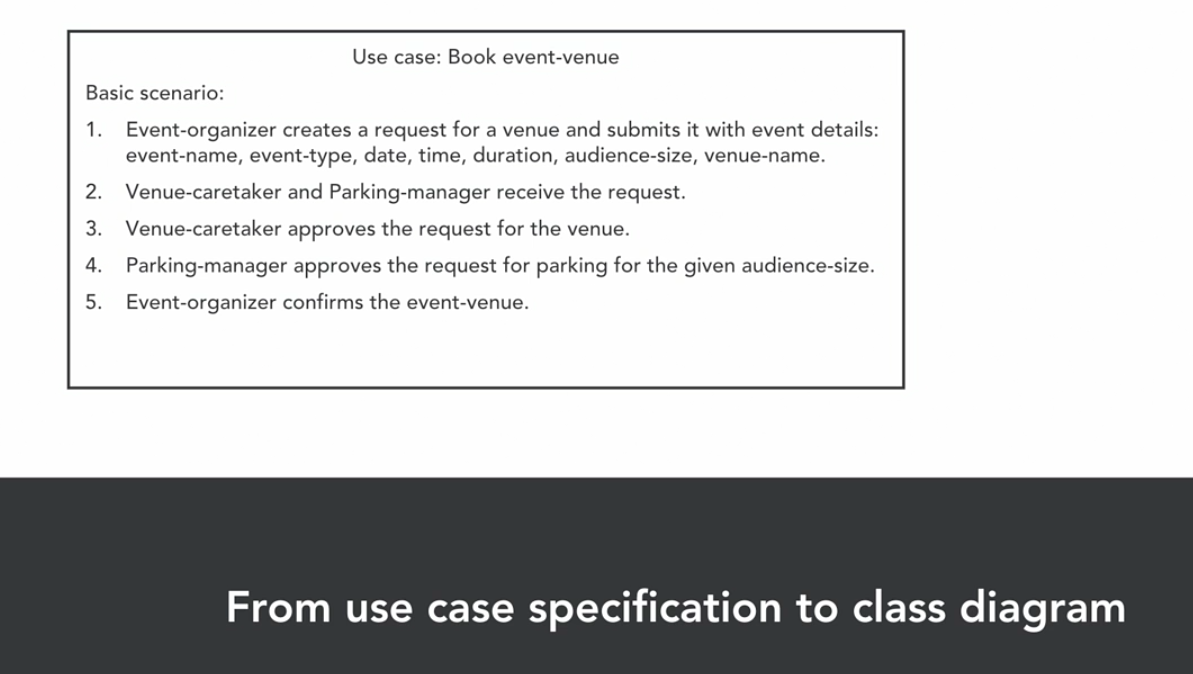
Operations or methods

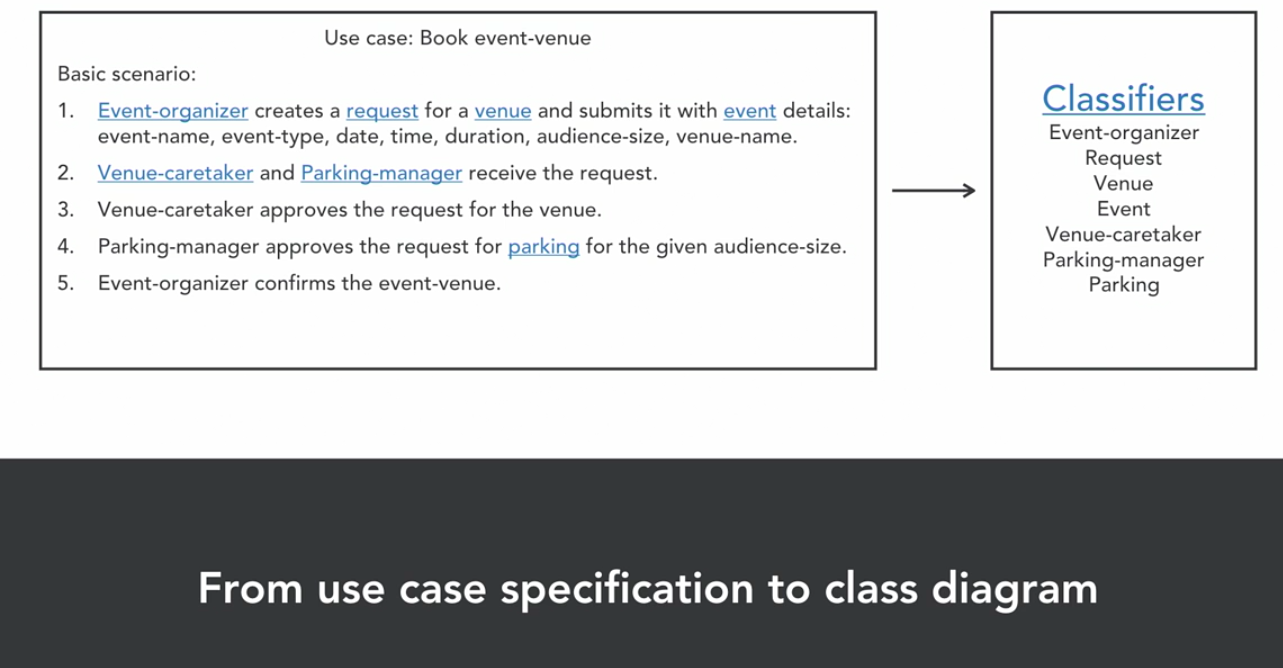
**Relationship**

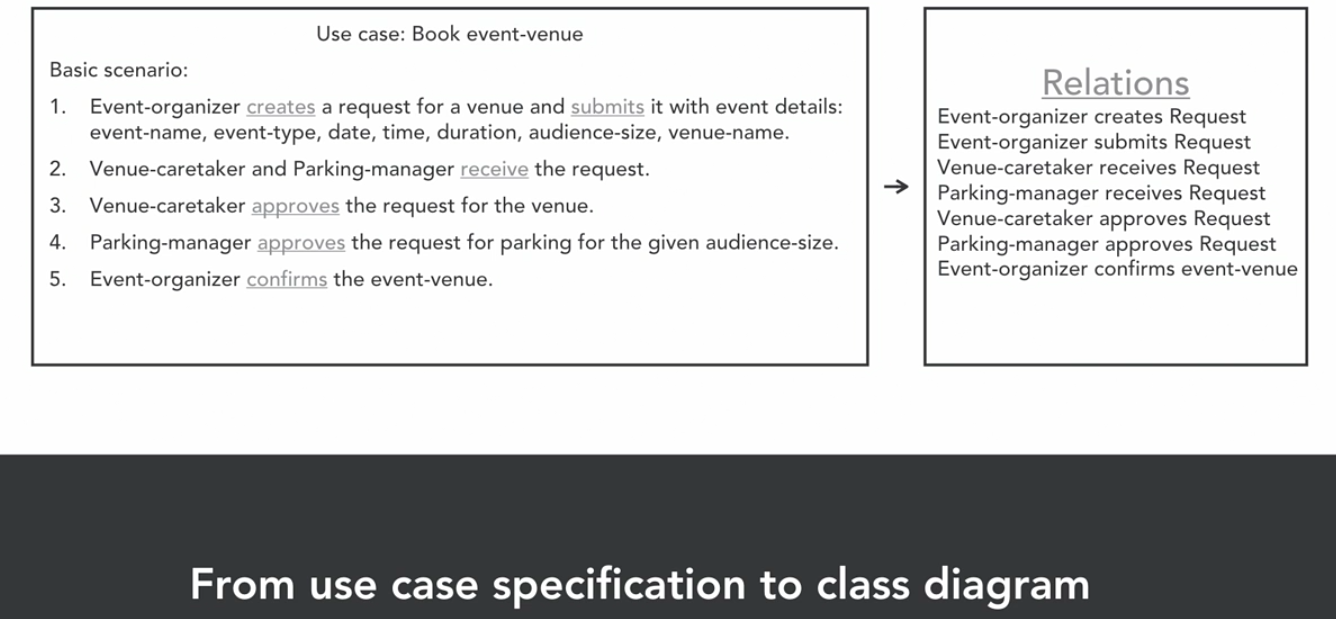
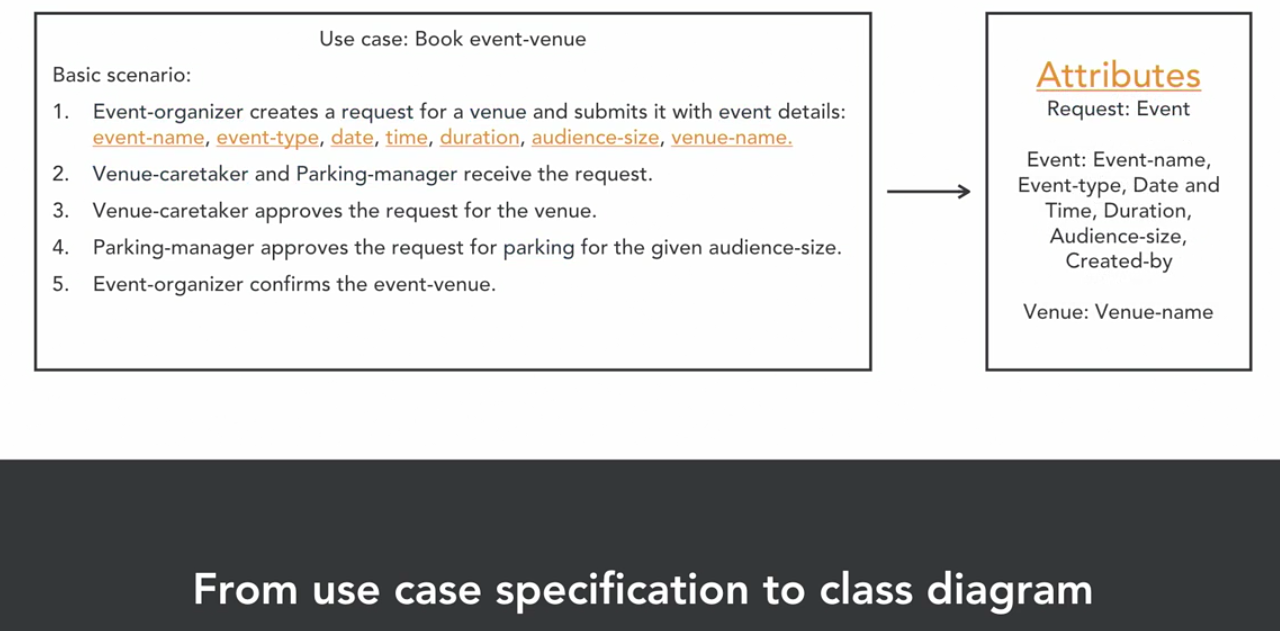
Relationship among classifiers

* Association
* Generalization (Inheritance)
* Dependency ---->

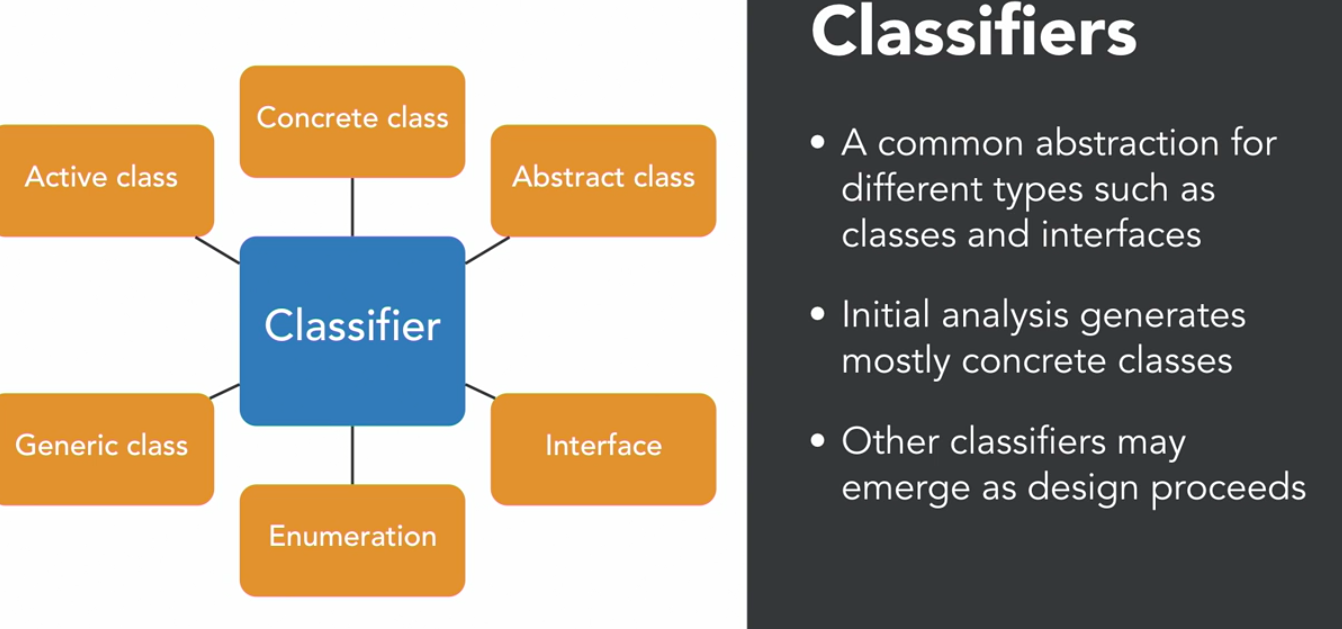
A sample scenario:

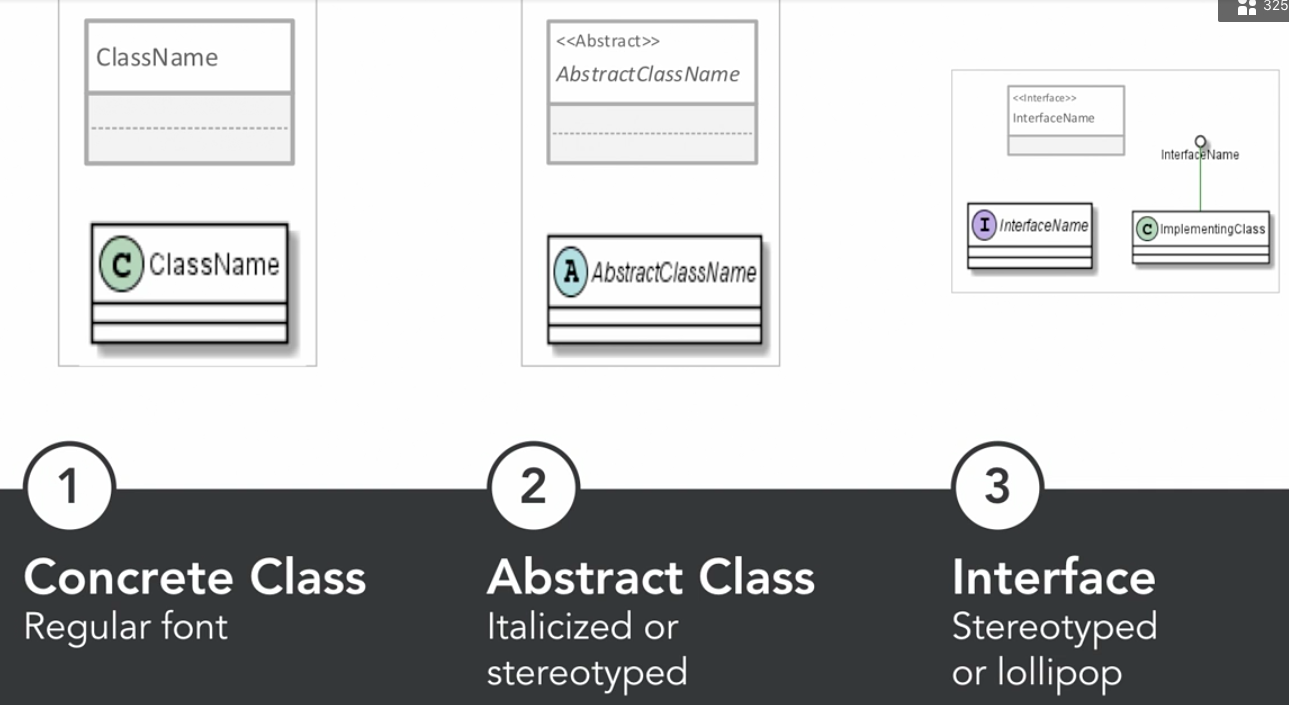


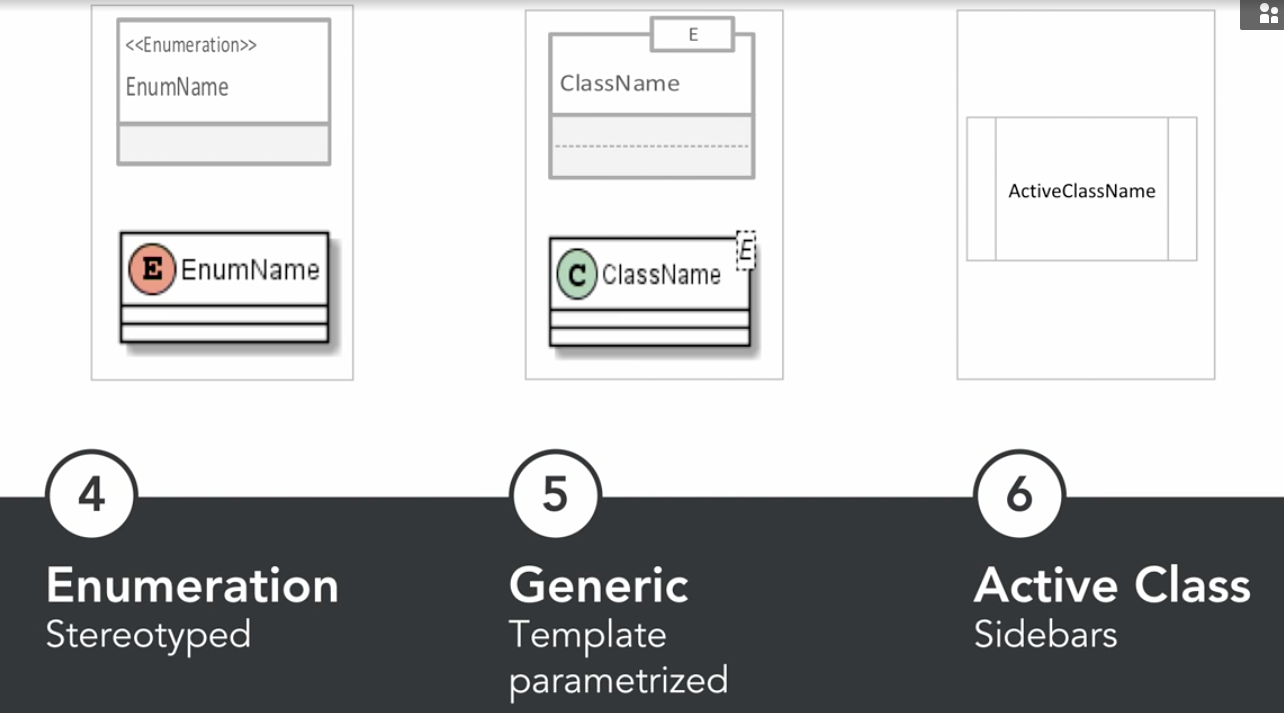




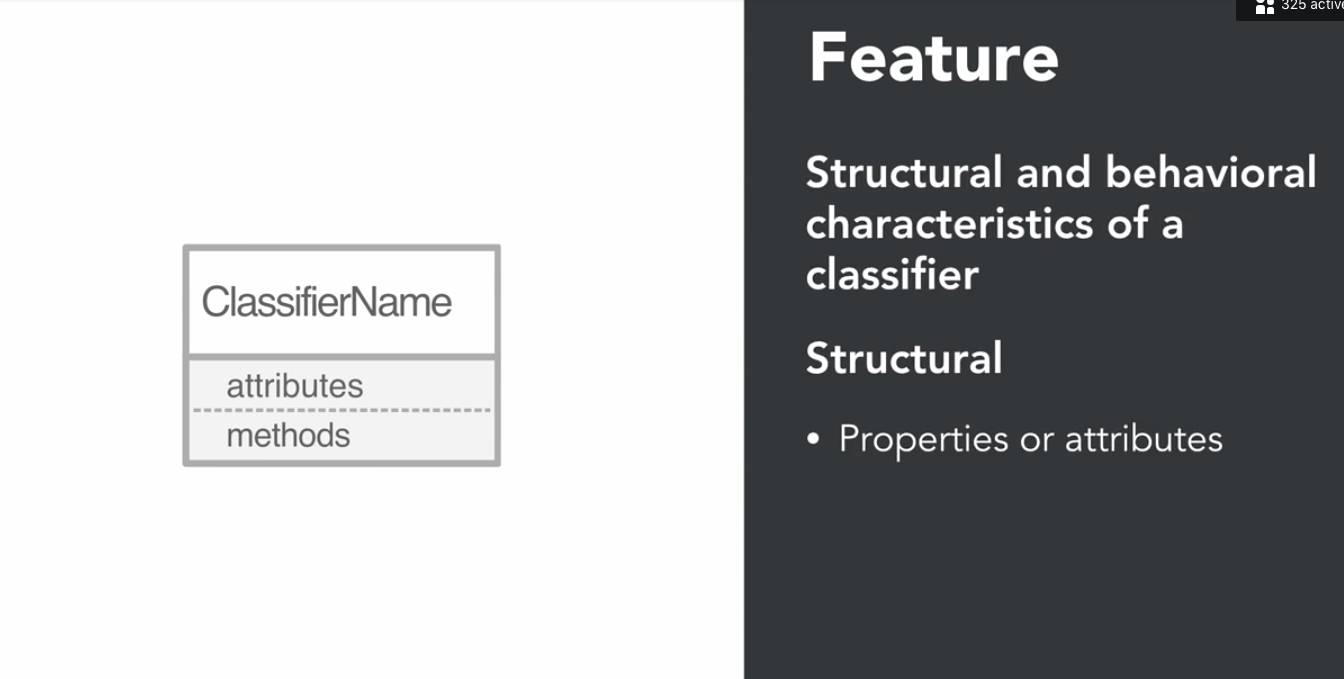
CLASSIFIERS

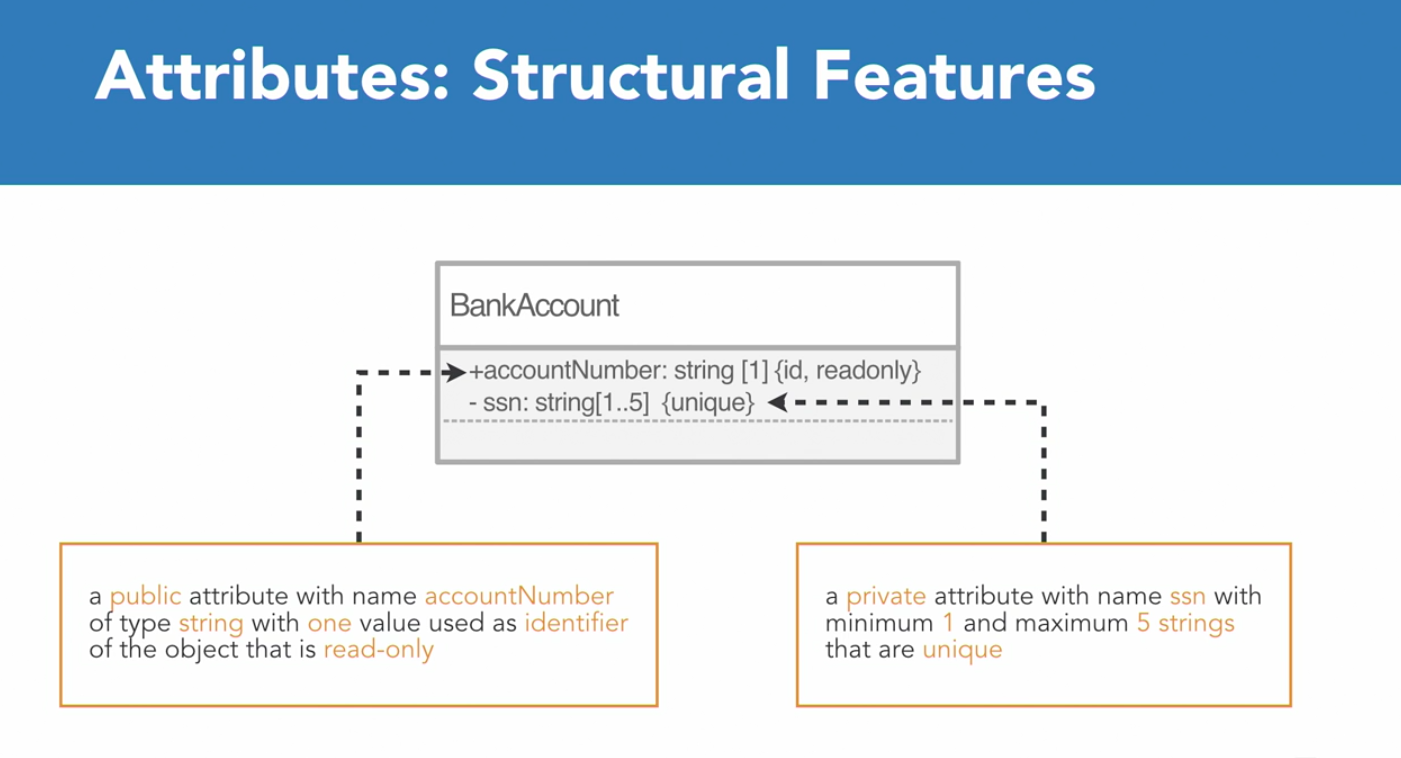


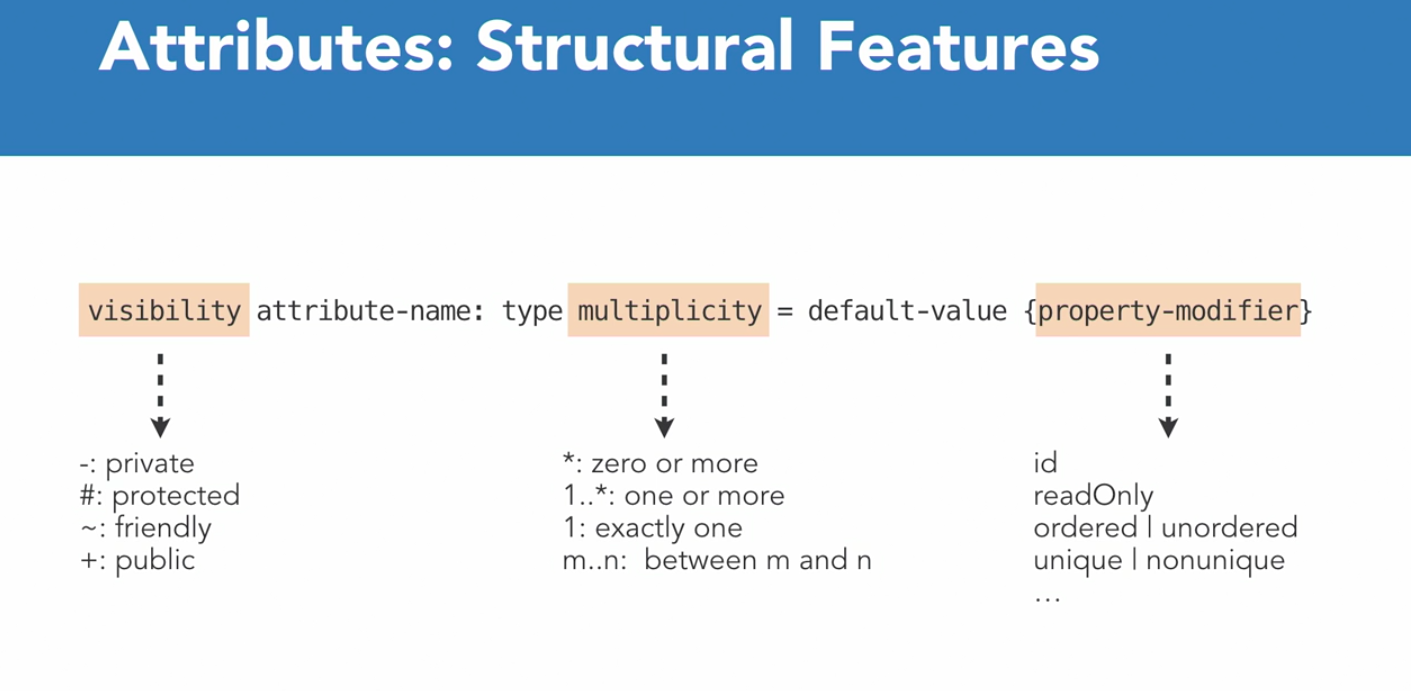




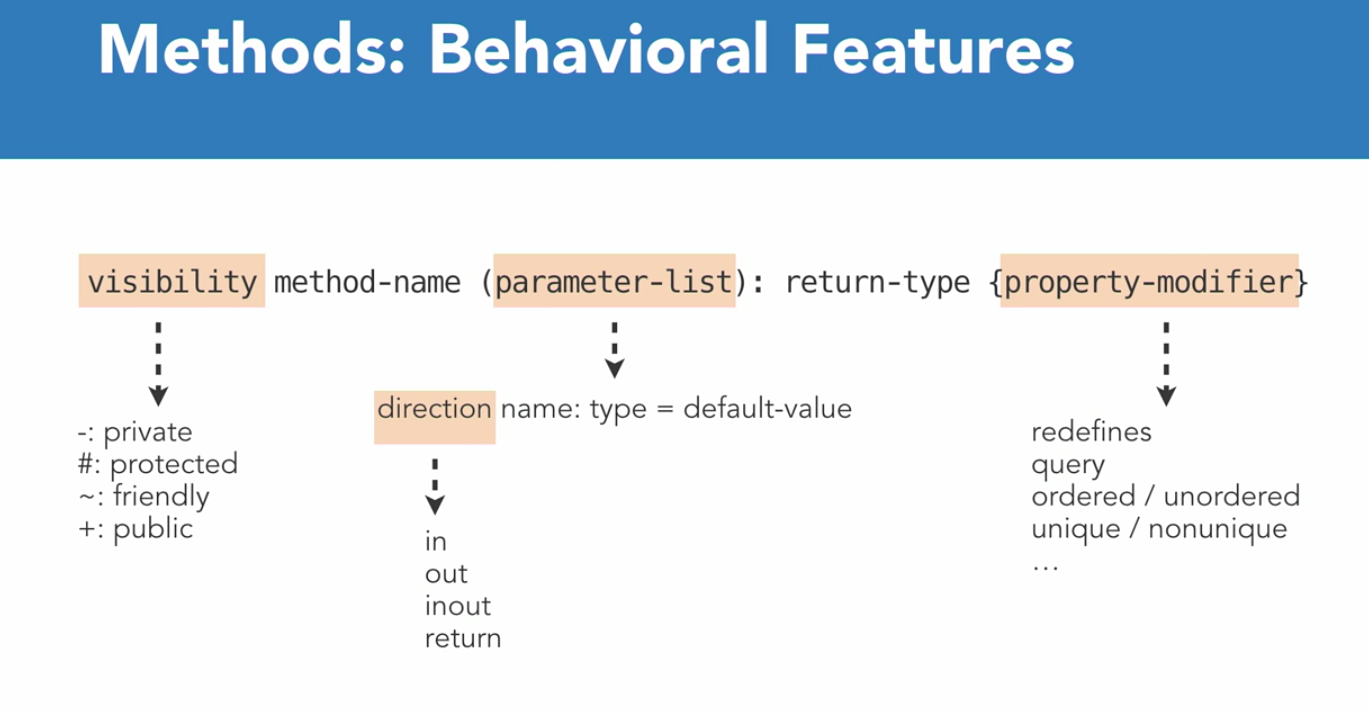
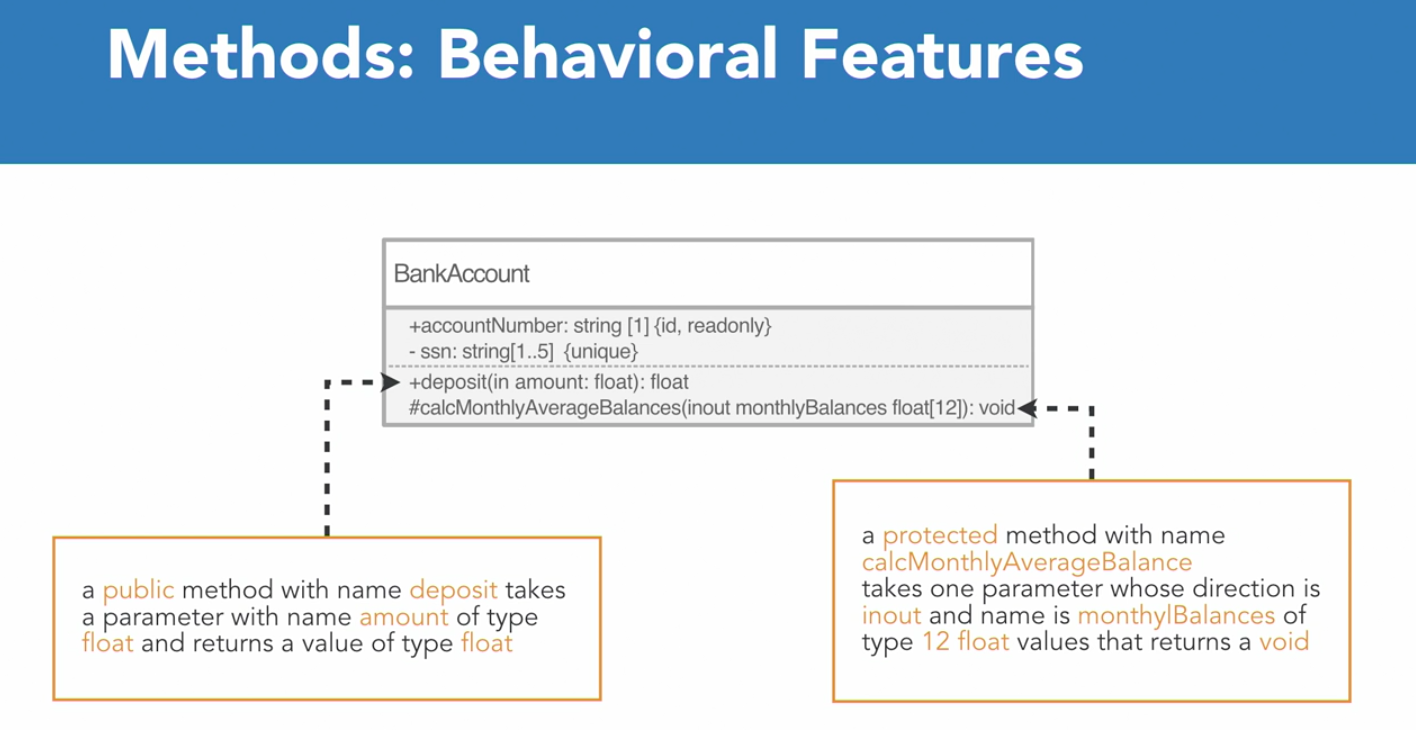
ATTRIBUTES: STRUCTURAL FEATURES



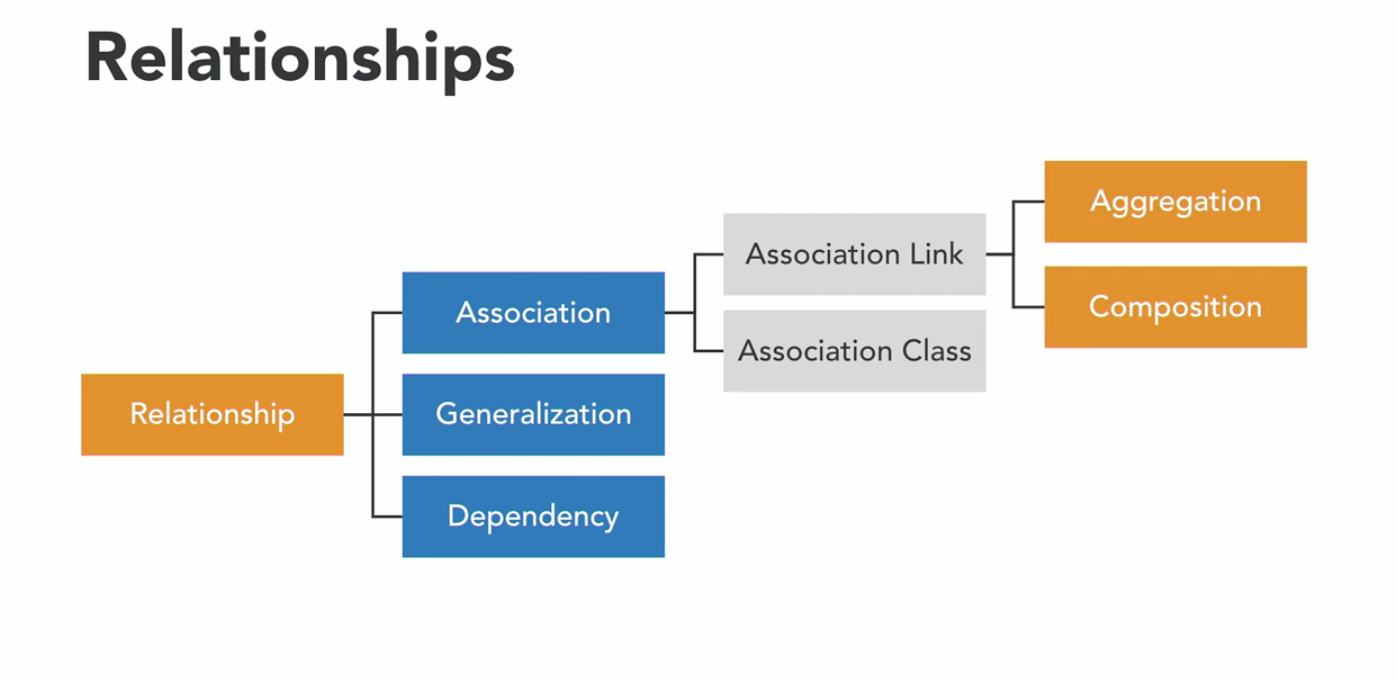


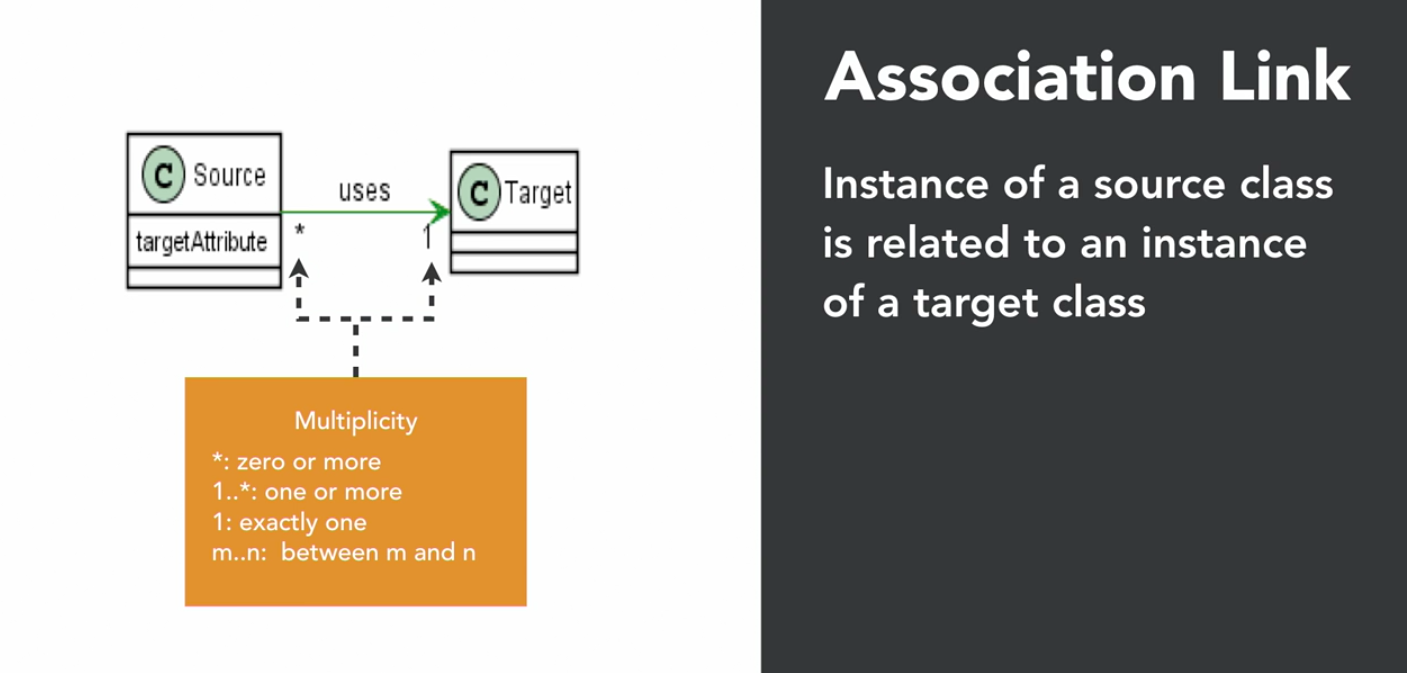


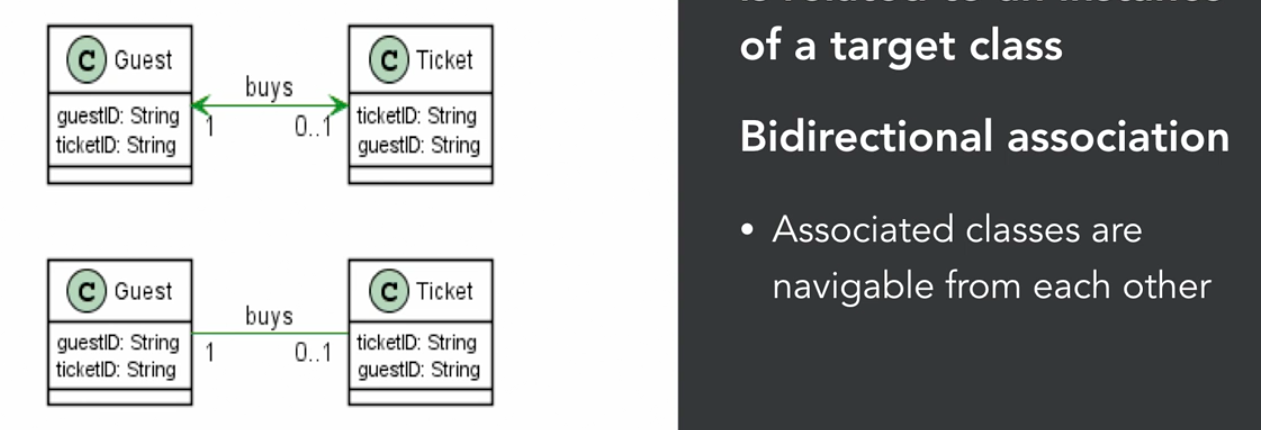
METHODS: BEHAVIORAL FEATURES

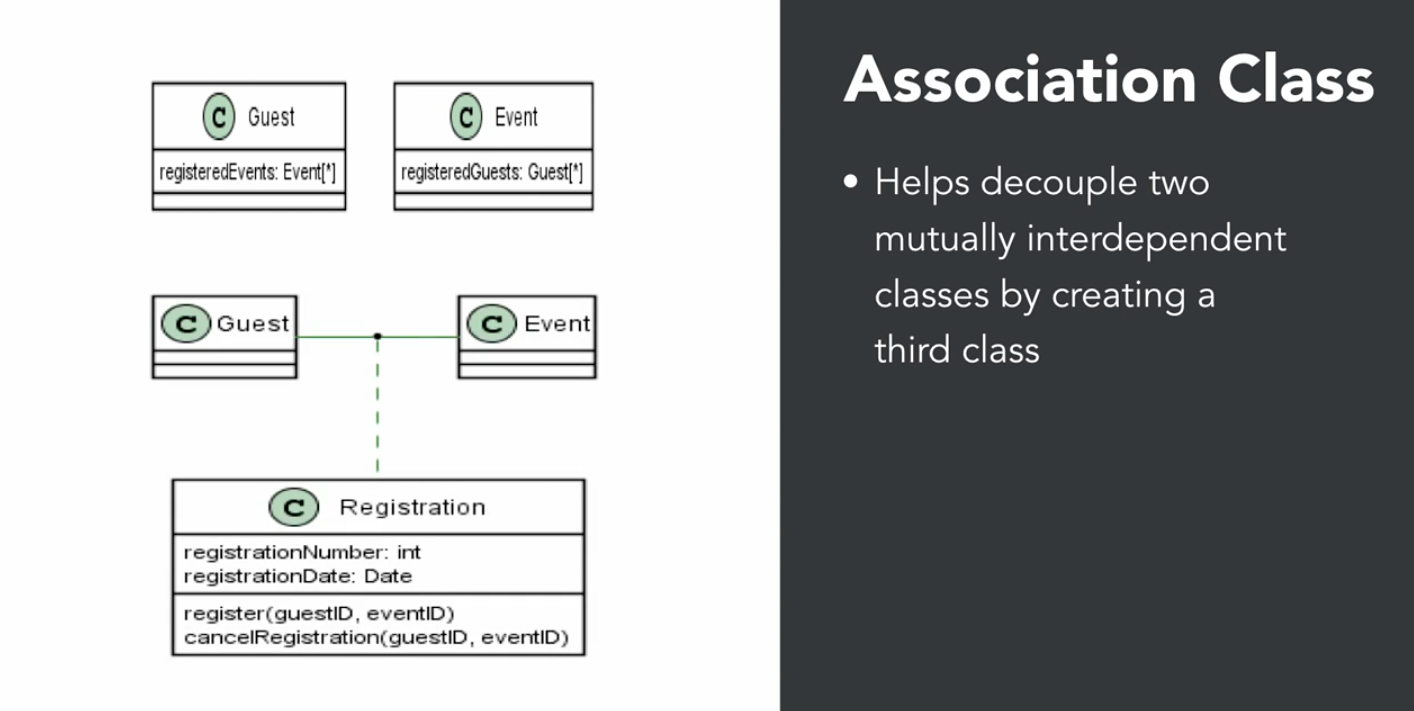


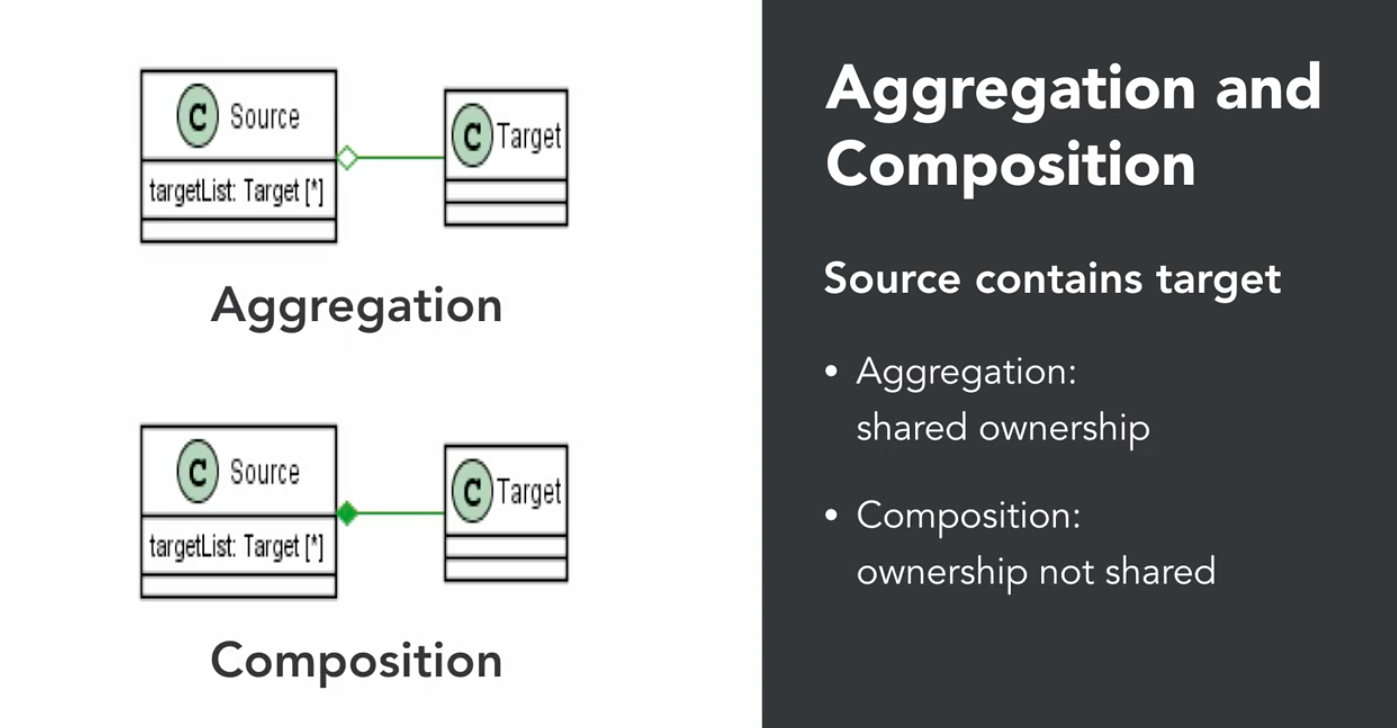
RELATIONSHIPS









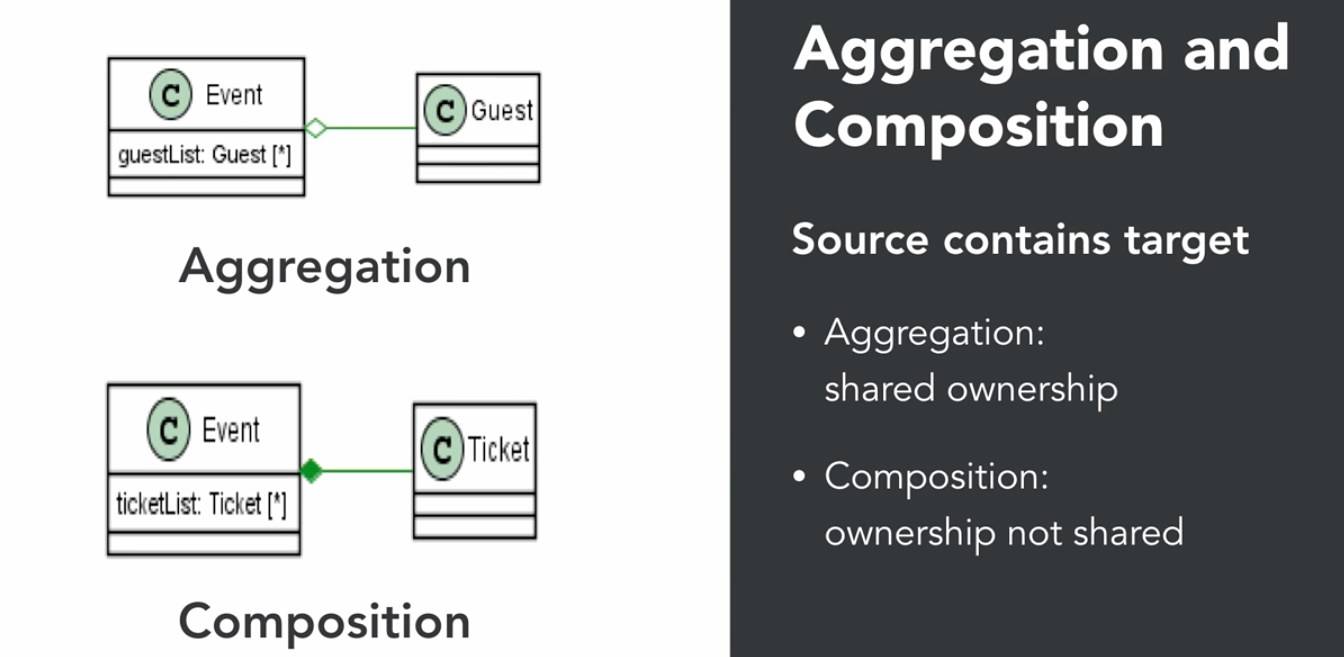


Aggregation

* “Has-a”
* Linked objects are independent of each other. Student has an address
* Child exits in the absence of parent.
* Independent existence possible
* University and Departments
* Library and Books

Composition (Stronger form of Aggregation)

* Human class is **composition** of heart and lungs
* House and Rooms (If house is demolished, the rooms do not exist)
* Computer and CPU

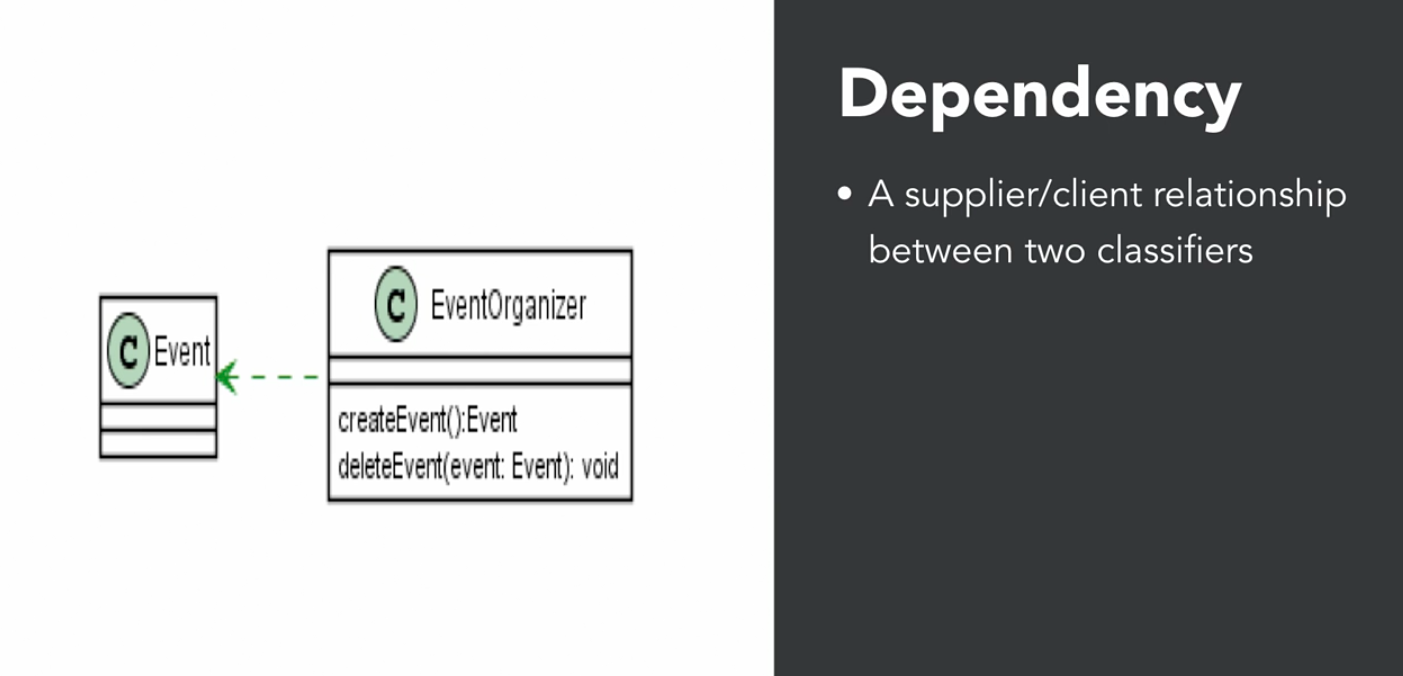
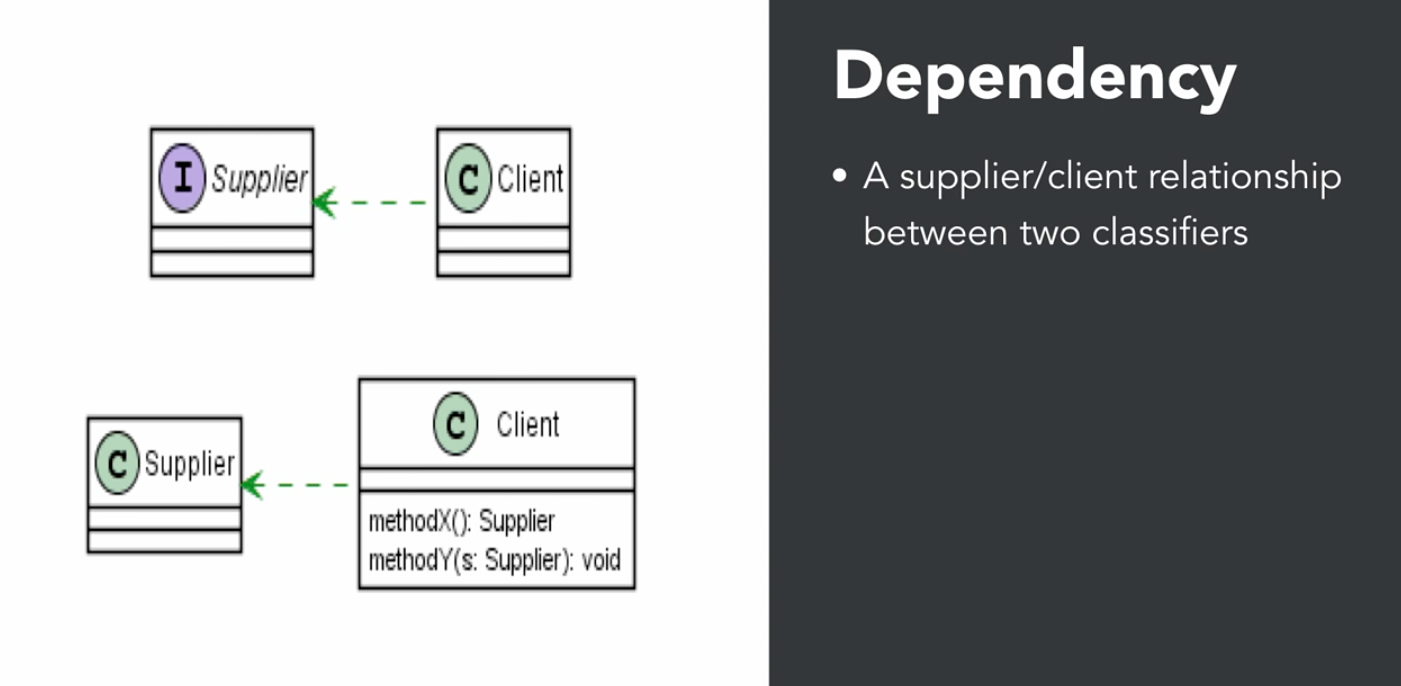


Aggregation

* Event contains target class Guest. But Guest can be a part of more than one Event
* Guest can visit other Event if this Event gets cancelled

Composition

* A ticket belongs to only one Event. If Event gets cancelled, ticket gets destroyed



Dependency because child has Event method….

Generalization (Inheritance)

