

What is a UML class diagram?

- A UML class diagram is a picture of the classes in an OO system
 - ☐ their fields and methods
 - ☐ connections between the classes that interact or inherit from each other
- Not represented in a UML class diagram:
 - ☐ details of how the classes interact with each other
 - ☐ algorithmic details; how a particular behavior is implemented

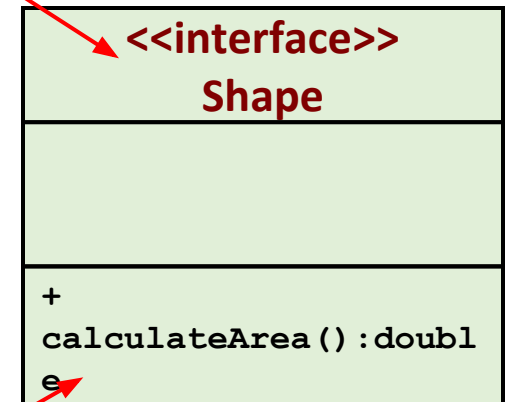
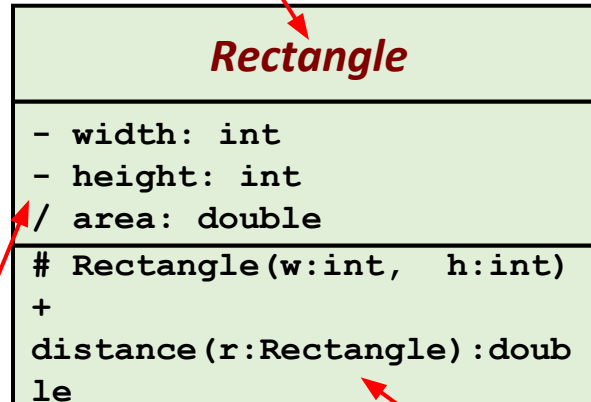
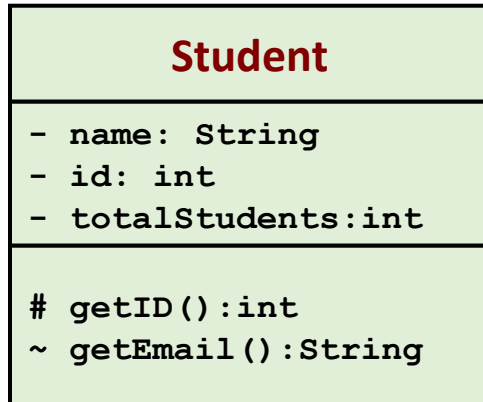
Benefits of Class Diagram

- Class Diagram Illustrates data models for even very complex information systems
- It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time
- It helps for better understanding of general schematics of an application.
- Allows drawing detailed charts which highlights code required to be programmed
- Helpful for developers and other stakeholders.

Diagram of a single class

Class name on top

- write «interface» on top of interfaces' names
- use *italics* for an abstract class name



Attributes (optional)

- In the middle

Operations/ methods (optional)

- may omit trivial (get/set) methods
- but don't omit any methods from an interface!
- should not include inherited methods

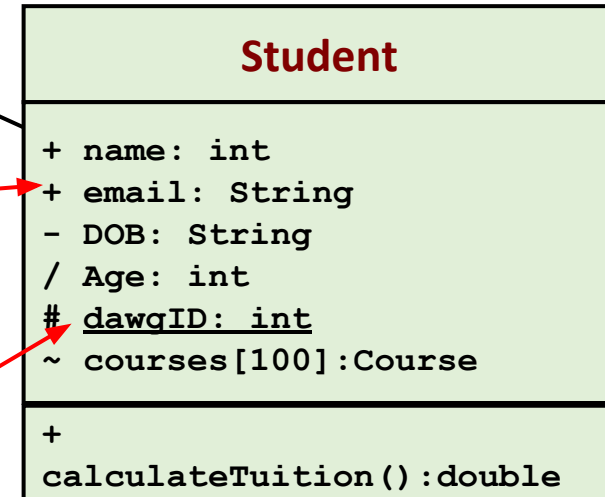
Class attributes (fields, instance variables)

visibility name : type [count] = default_value

- Visibility

- + public
- # protected
- private
- ~ package (default)
- / derived

- underline static attributes



Class operations / methods

visibility name(parameters) : return_type

- Visibility
 - + public
 - # protected
 - private
 - ~ package (default)
- underline static methods
- Parameters listed as name:type
- Omit return_type on constructors and when return type is void

Student
+ name: String + email: String - DOB: String / Age: int # <u>dawgID: int</u> ~ courses[100]:Course
+Student(n:String,dob:String) + getTotalCredits():Course # calculateTuition():double + <u>calculateGPA(crs:Course[]):float</u>

Relationships between class

- **Generalization:** an inheritance relationship

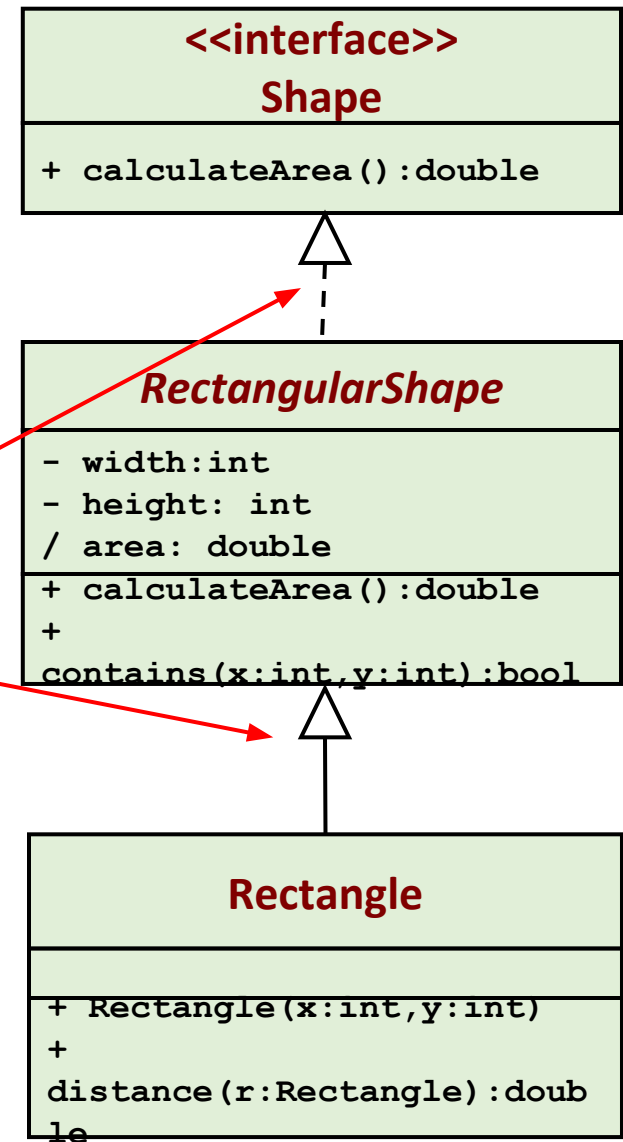
- ☐ inheritance between classes
- ☐ interface implementation

- **Association:** a usage relationship

- ☐ dependency
- ☐ aggregation
- ☐ composition

Generalization relationships

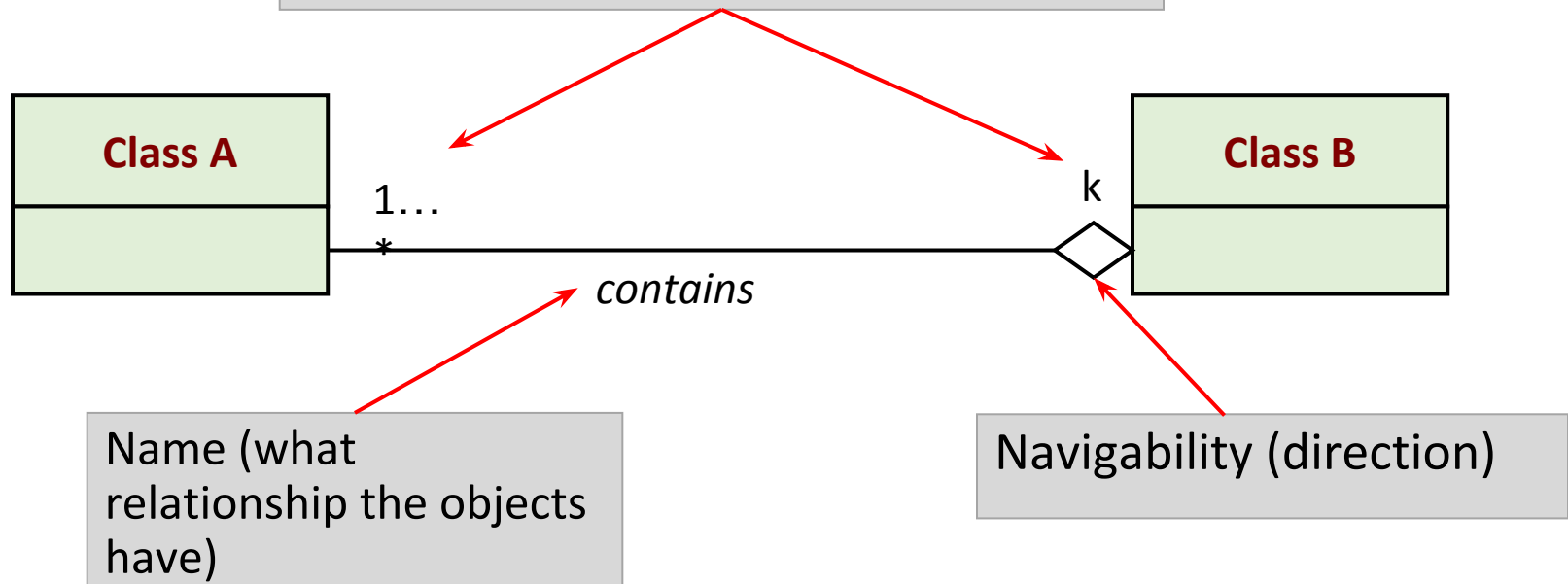
- Hierarchies drawn top-down
- Arrows point upward to parent
- Line/arrow styles indicate if parent is a(n):
 - ☐ **class**: solid line, black arrow
 - ☐ **interface**: dashed line, white arrow
 - ☐ **abstract class**: solid line, white arrow



Association (usage) relationships

Multiplicity (how many are used)

- * (zero or more)
- 1 (exactly one)
- 2..4 (between 2 and 4, inclusive)
- 3..* (3 or more, * may be omitted)



Association multiplicities

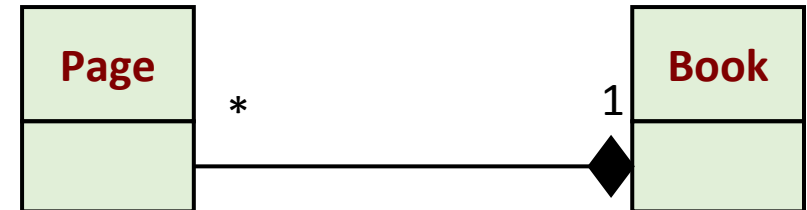
One to one

- Each car has exactly one engine
- Each engine belongs to exactly one car



One to many

- Each book has many pages
- Each page belongs to exactly one book



Association types

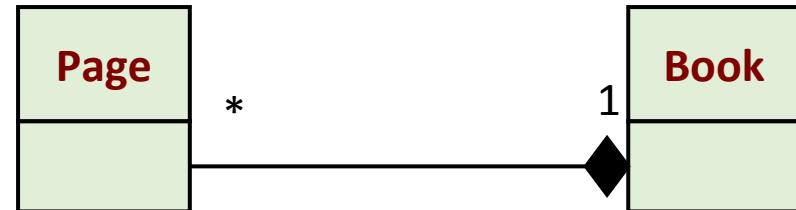
Aggregation: “is part of”

- symbolized by a clear white diamond



Composition: “is entirely made of”

- stronger version of aggregation
- the parts live and die with the whole
- symbolized by a black diamond



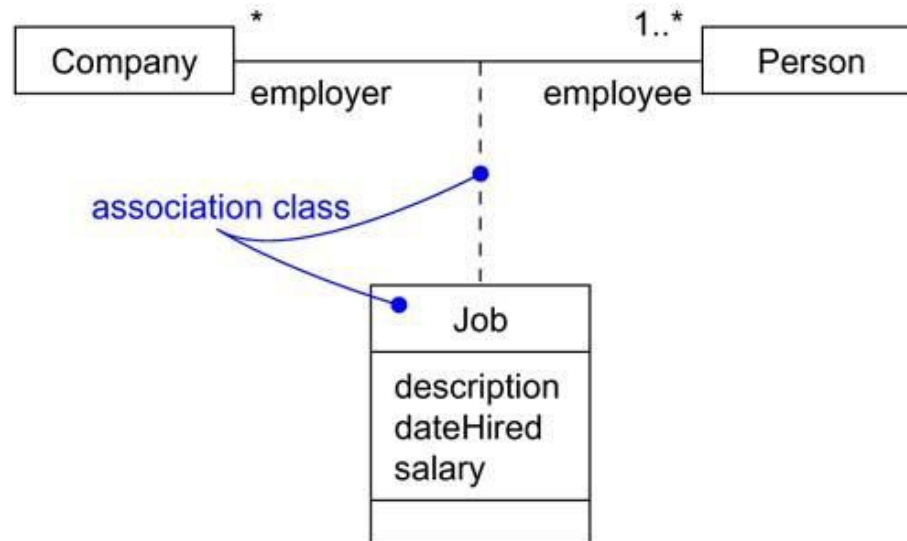
Dependency: “uses temporarily”

- symbolized by dotted line
- often is an implementation detail, not an intrinsic part of the object's state

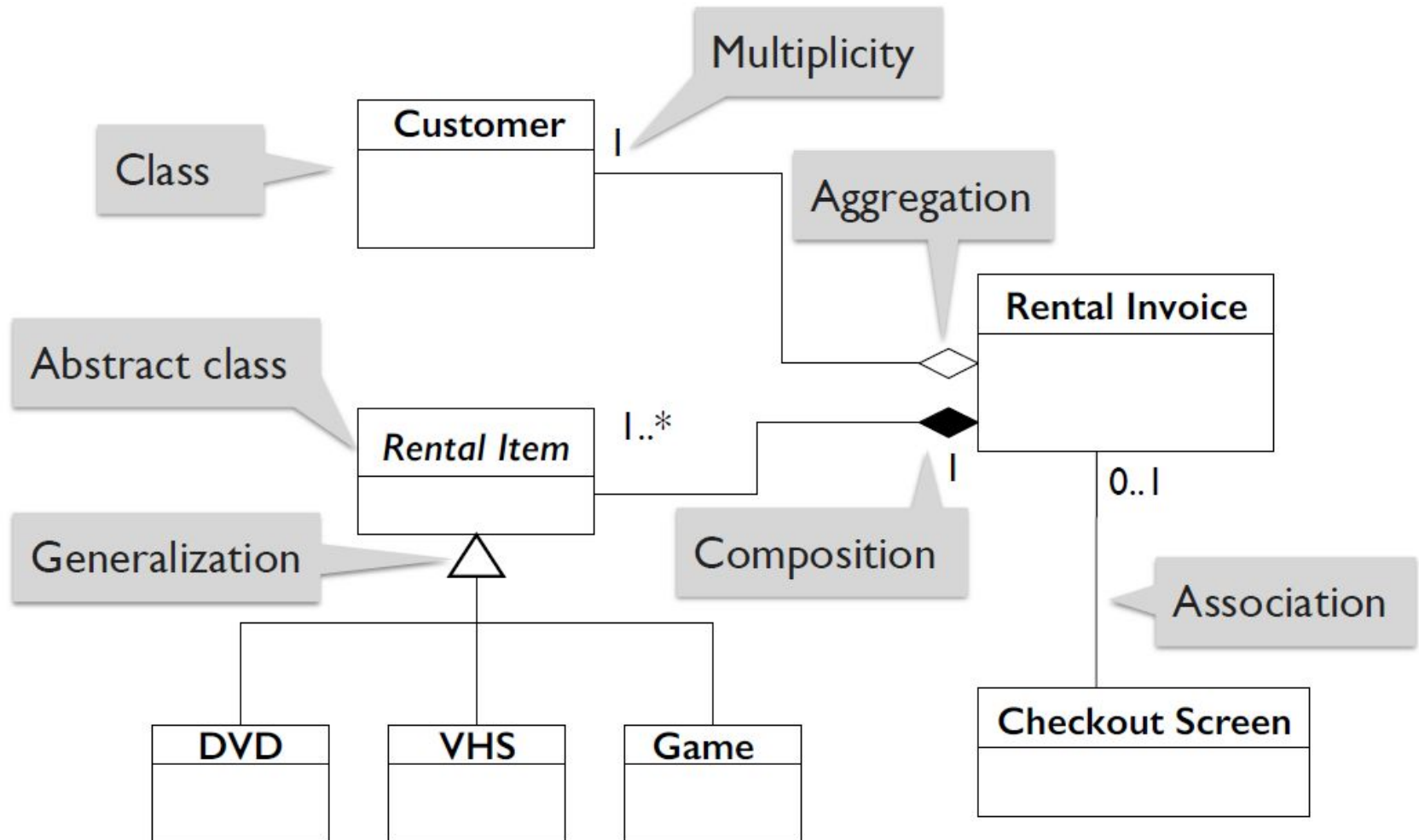


Association Class

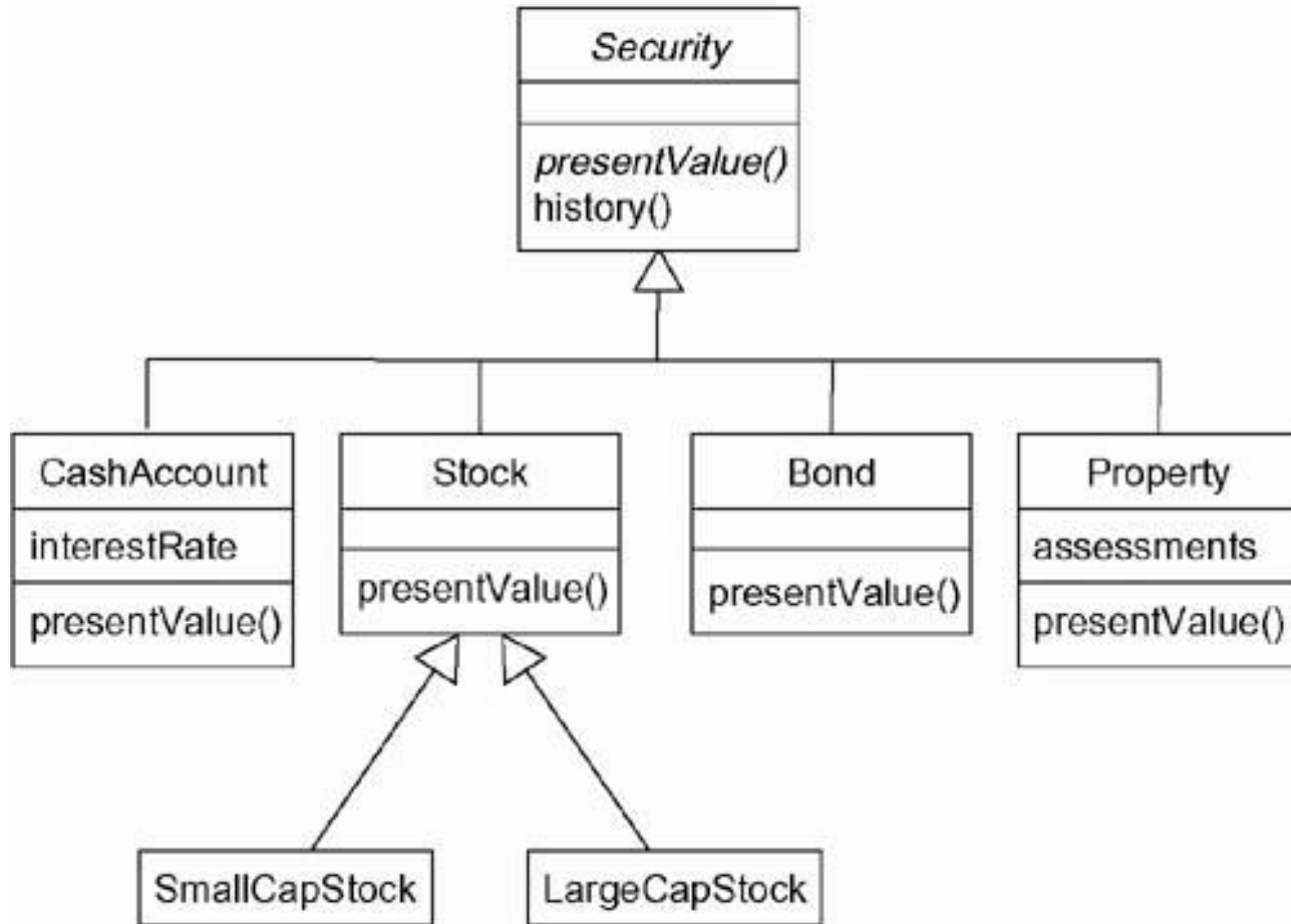
- In an association between two classes, the association itself might have properties.
- For example, in an employer/employee relationship between a Company and a Person, there is a Job that represents the properties of that relationship that apply to exactly one pairing of the Person and Company.
- It wouldn't be appropriate to model this situation with a Company to Job association together with a Job to Person association. That wouldn't tie a specific instance of the Job to the specific pairing of Company and Person.



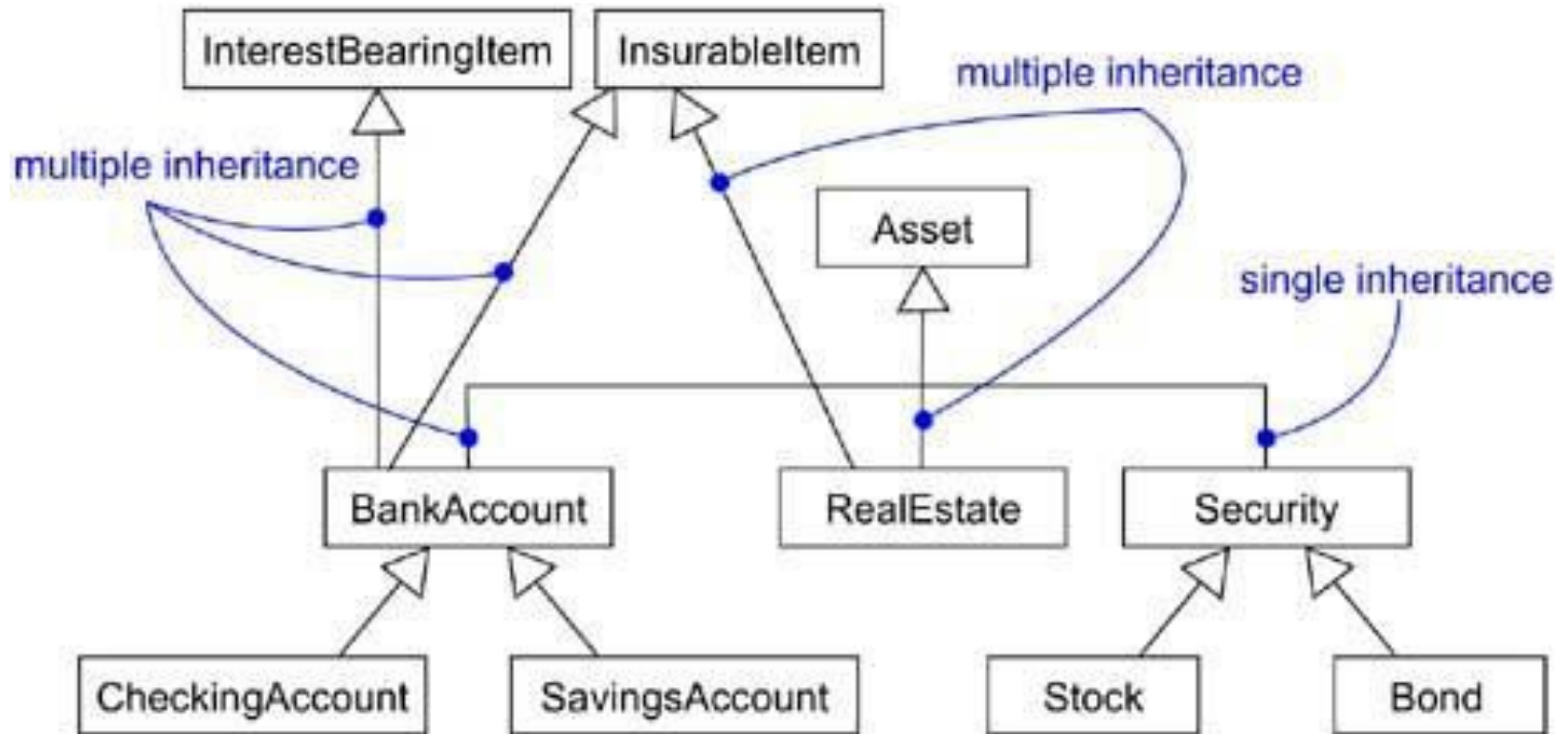
Example: Video rental store

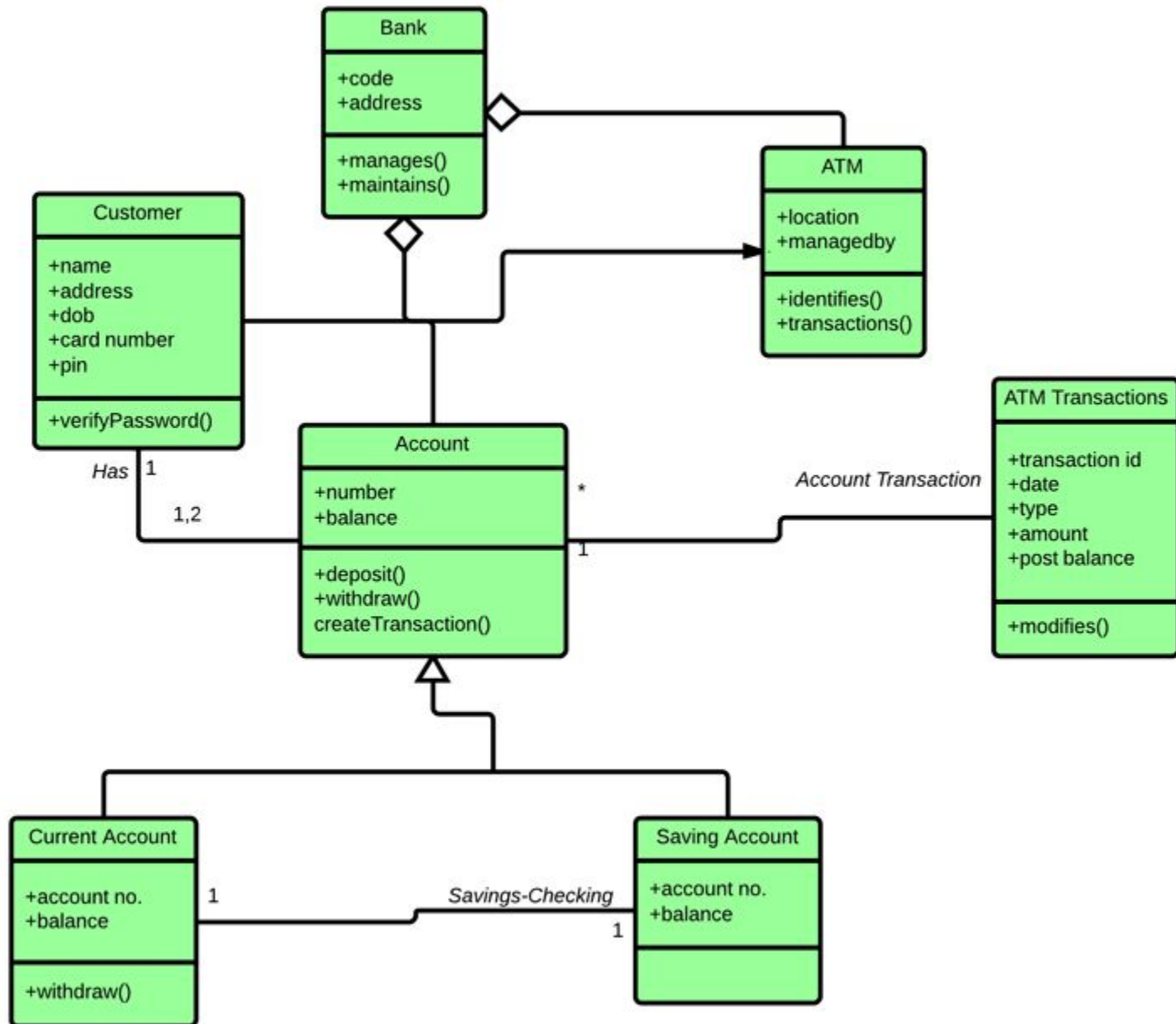


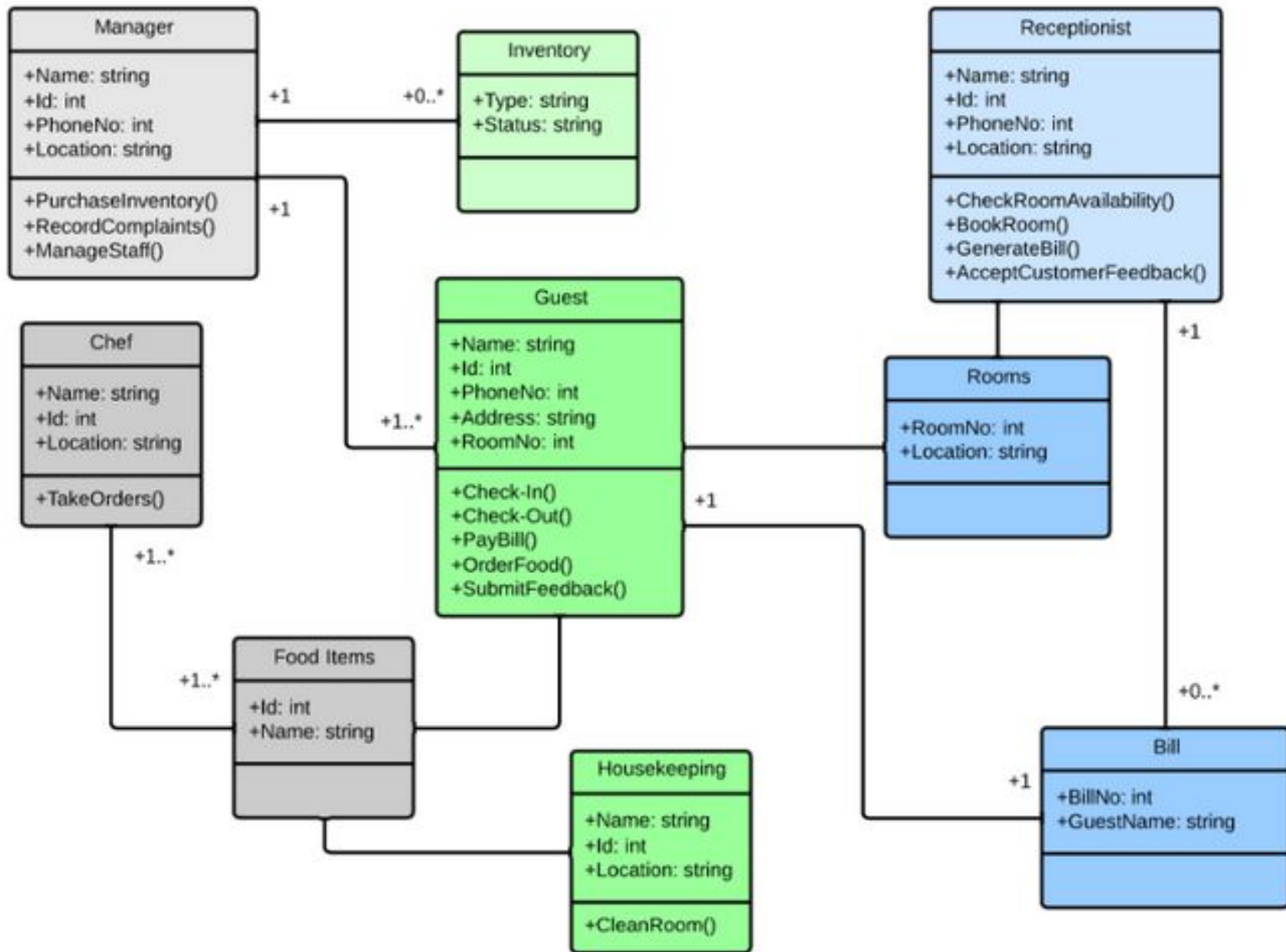
More Example of Inheritance

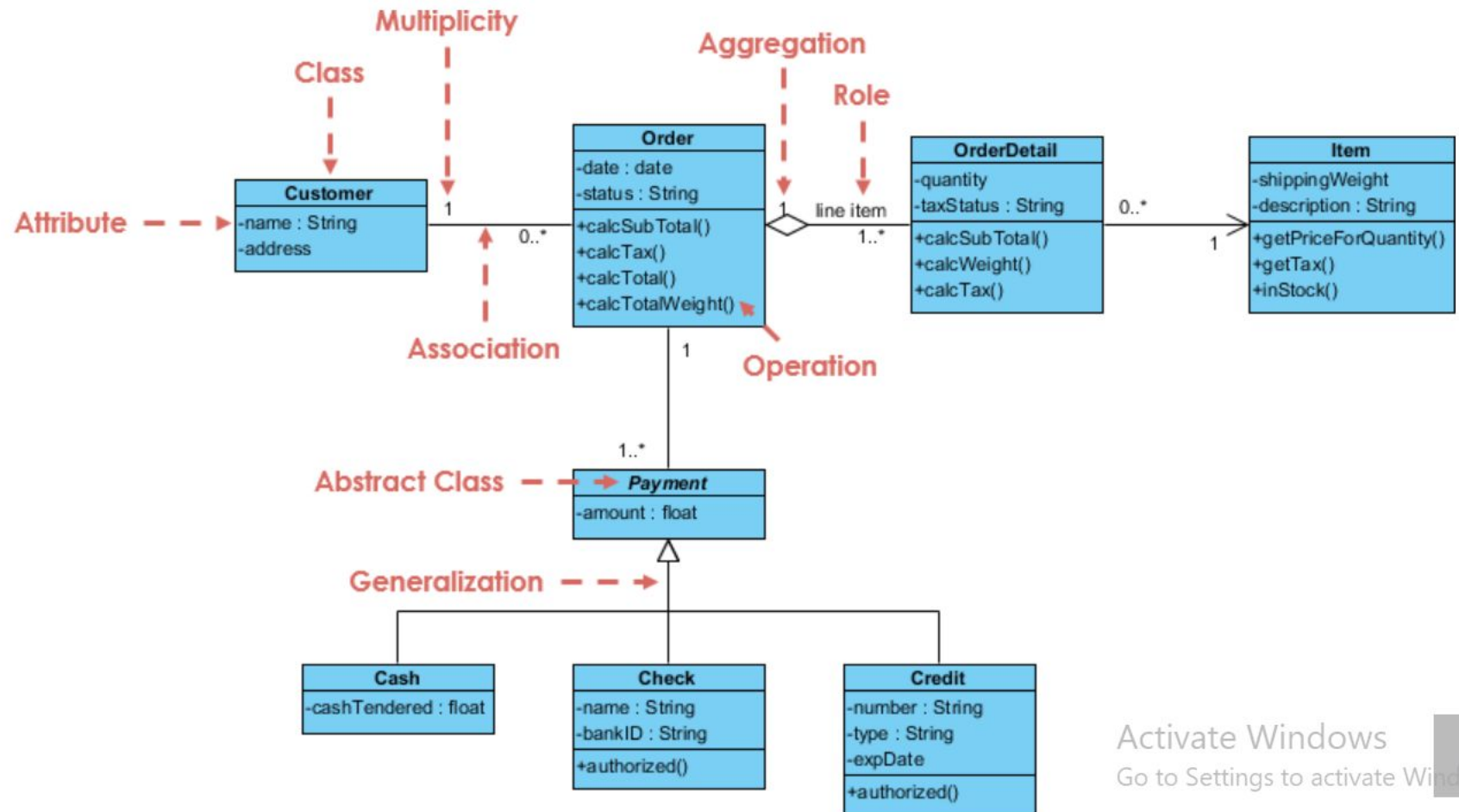


Multiple Inheritance

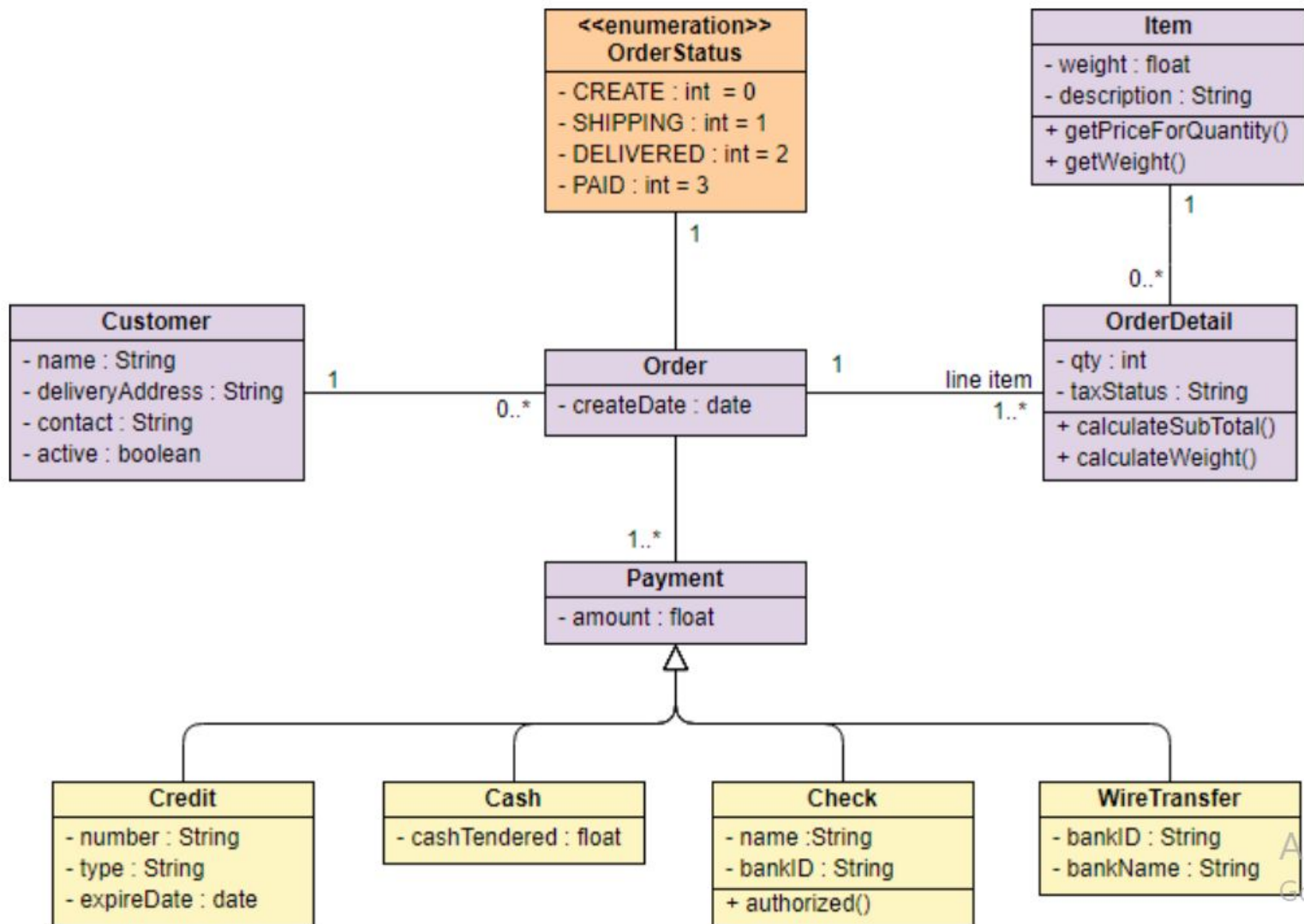


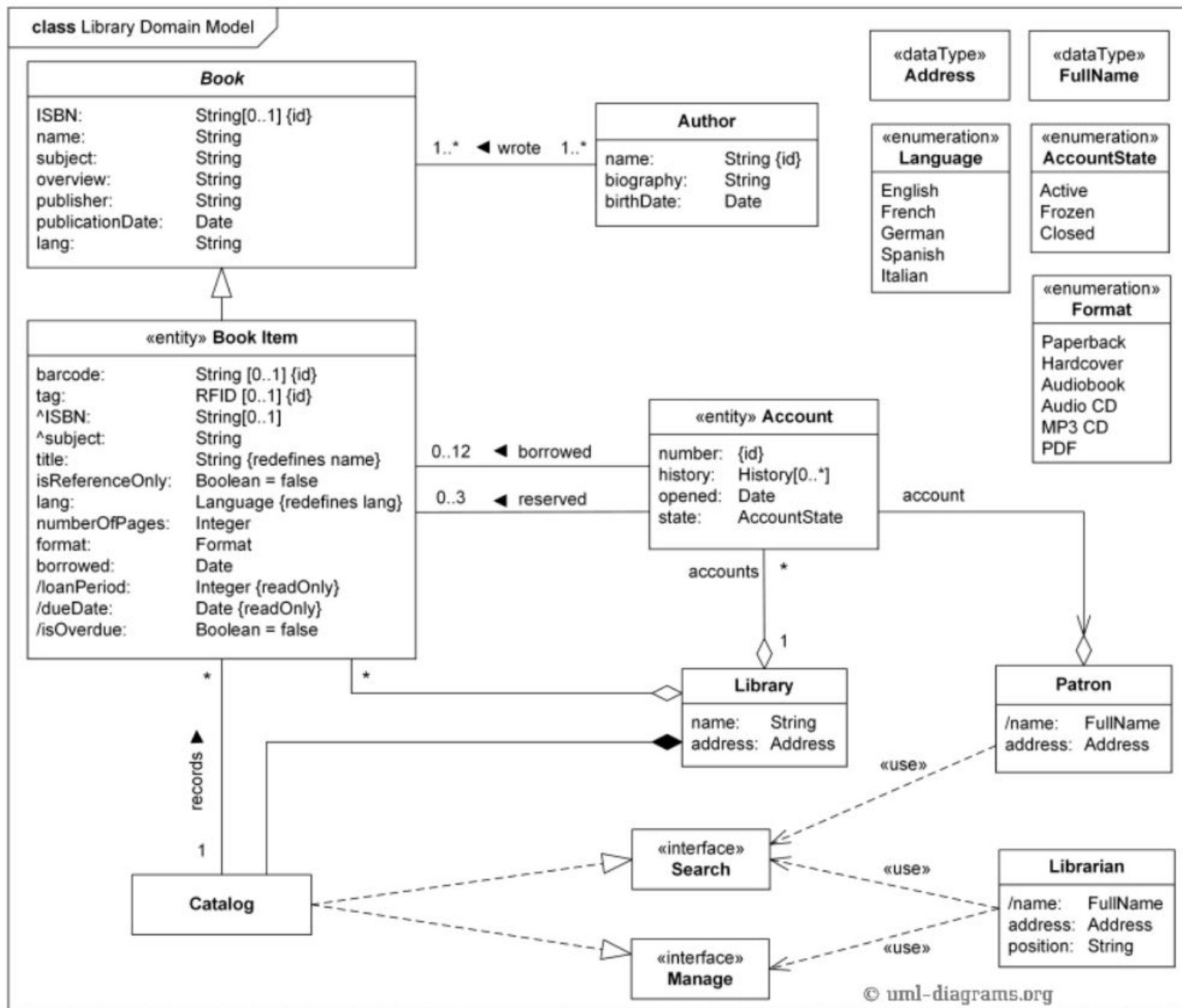


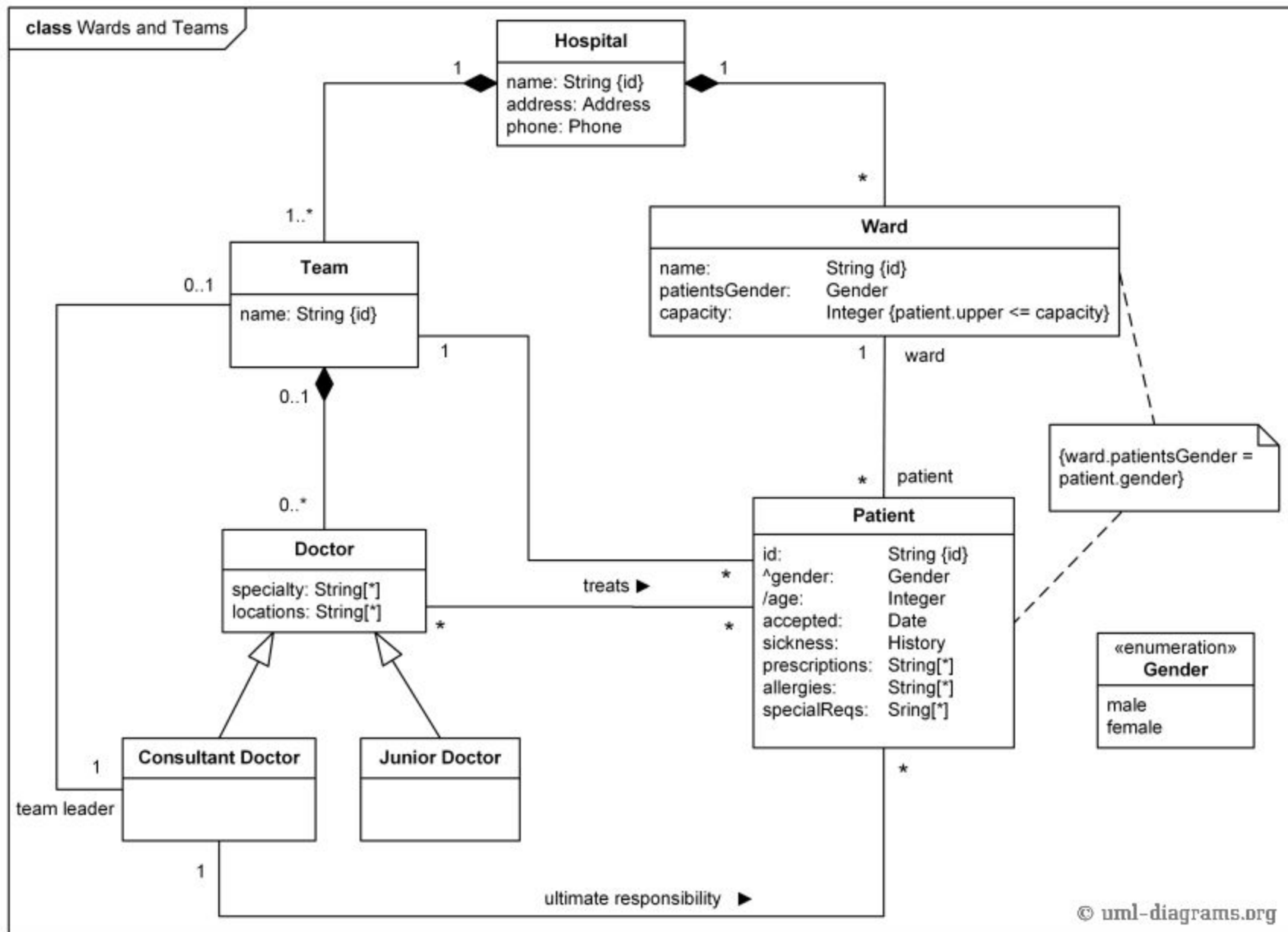




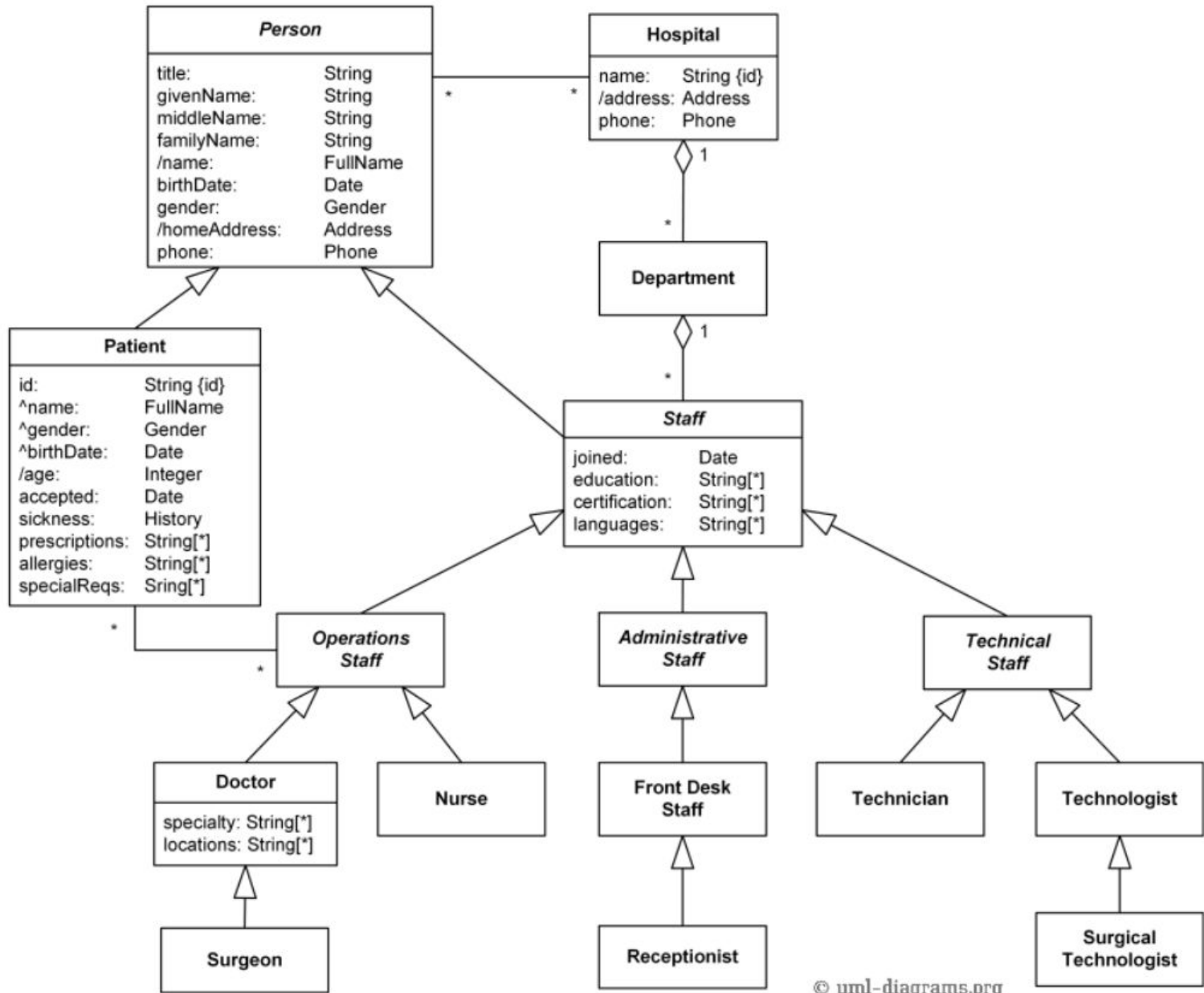
Activate Windows
Go to Settings to activate Windows

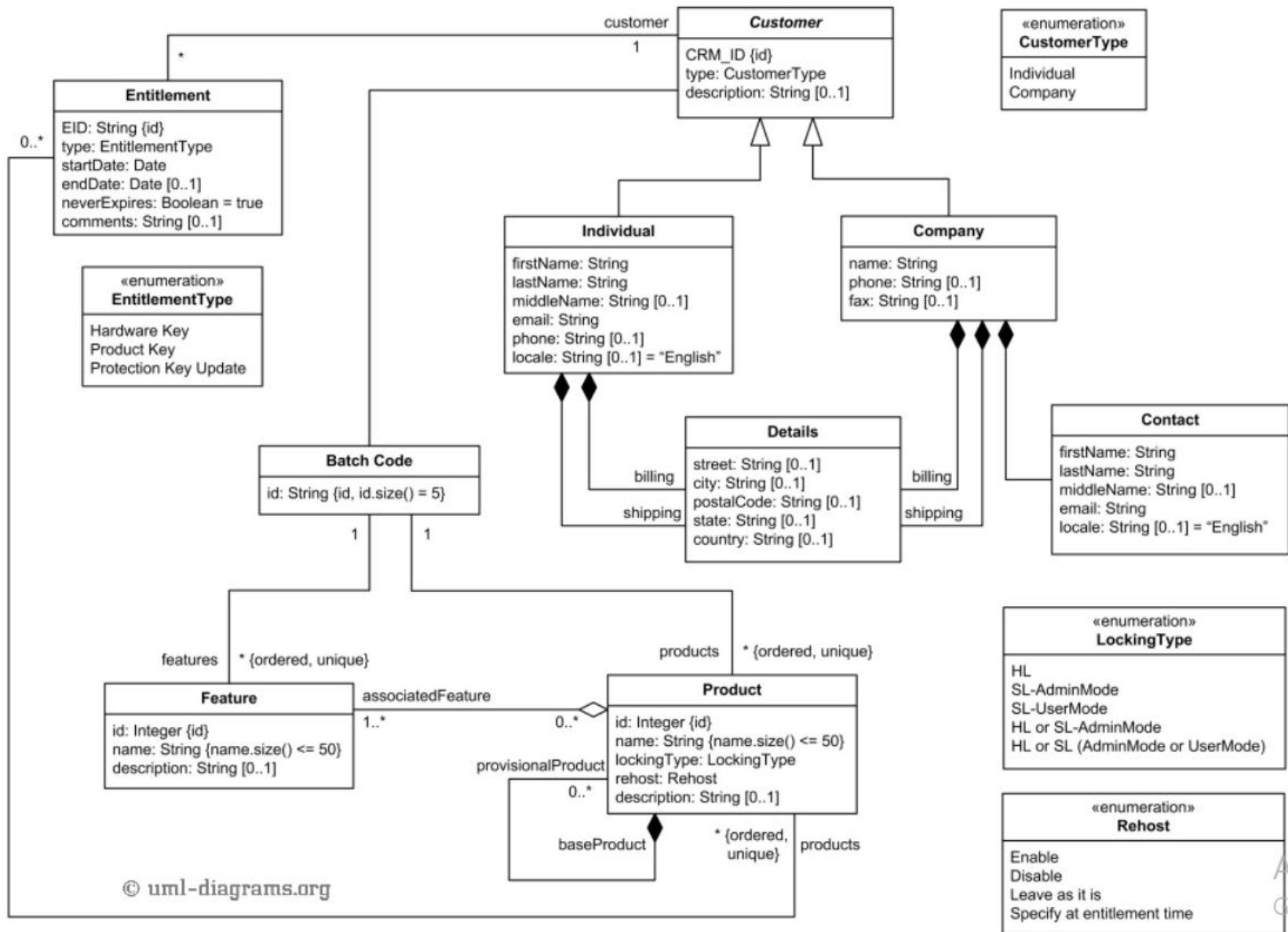


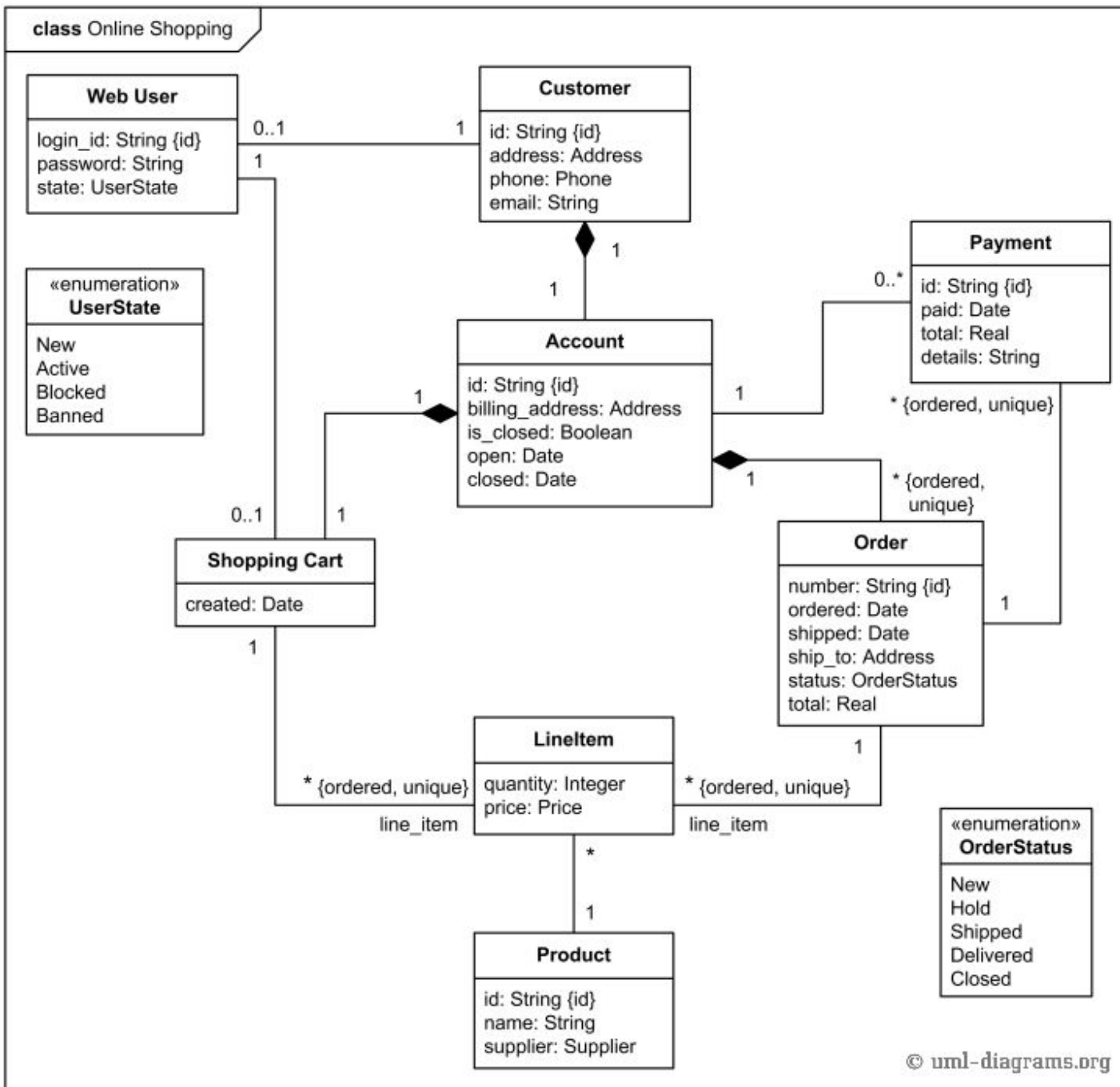


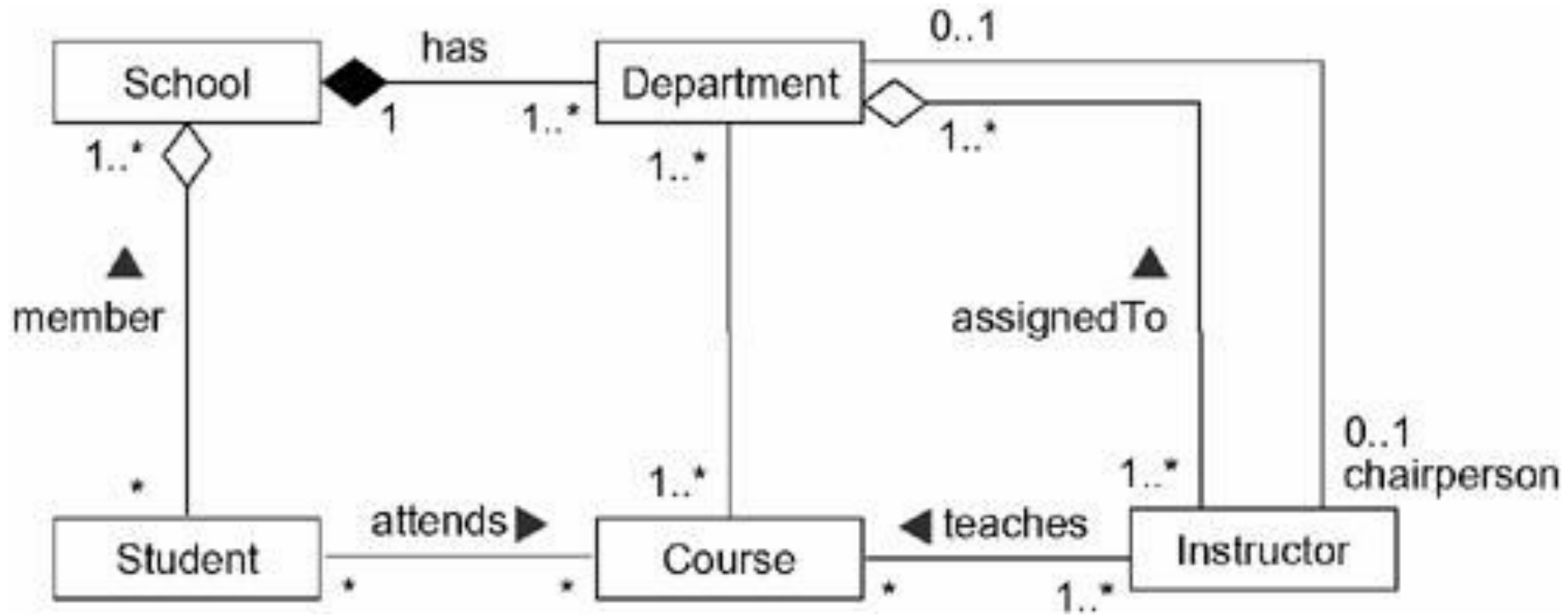


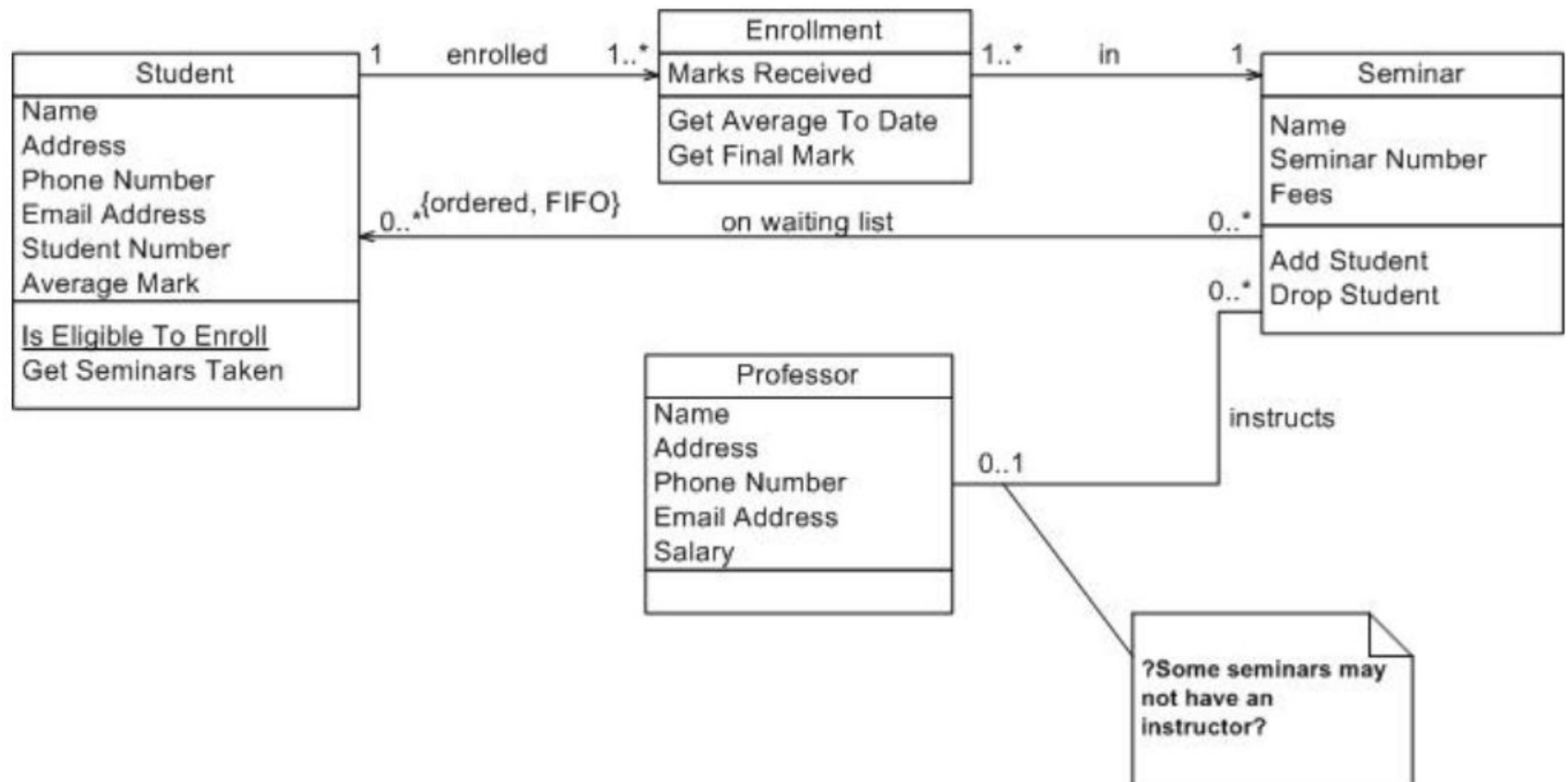
class Organization

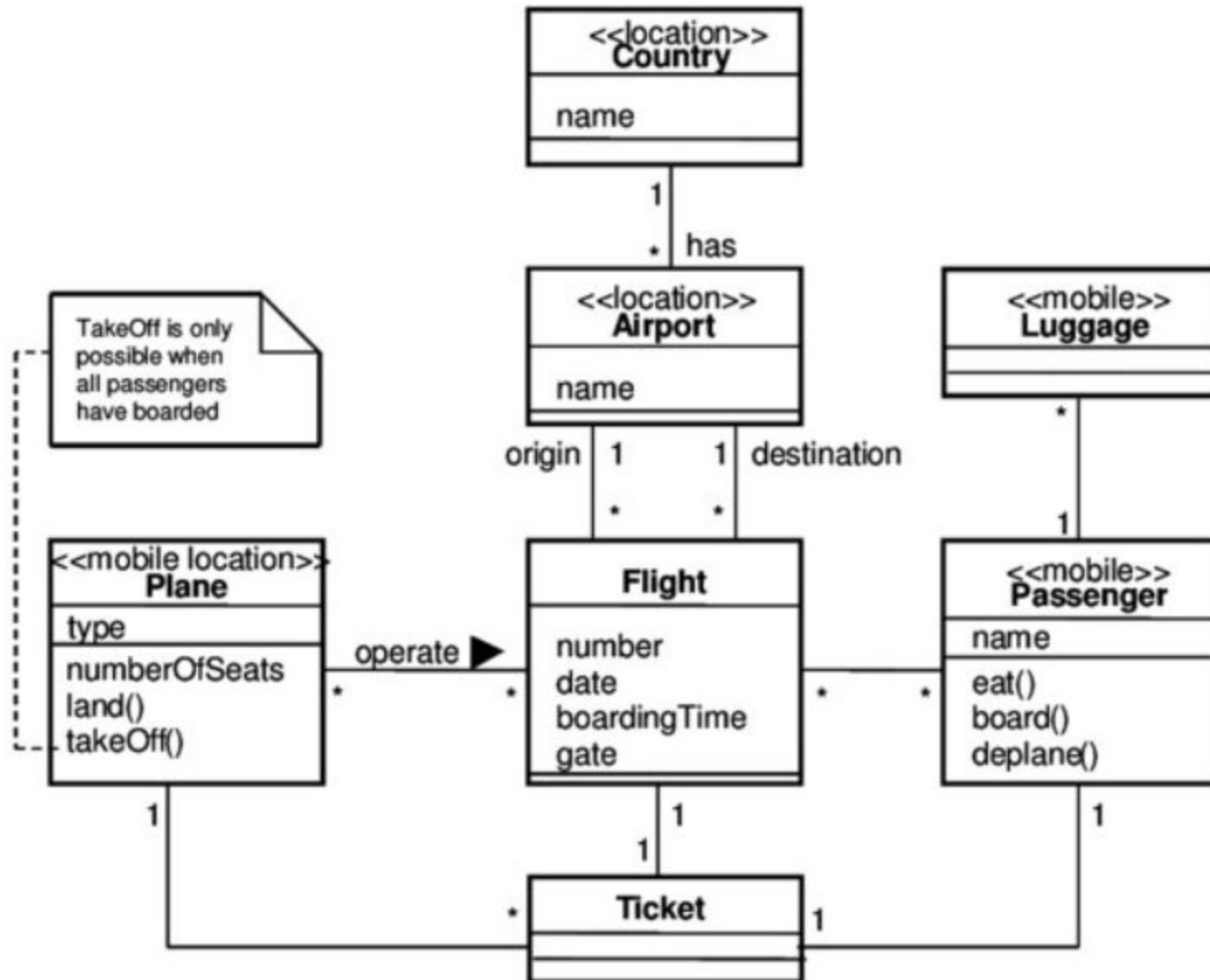


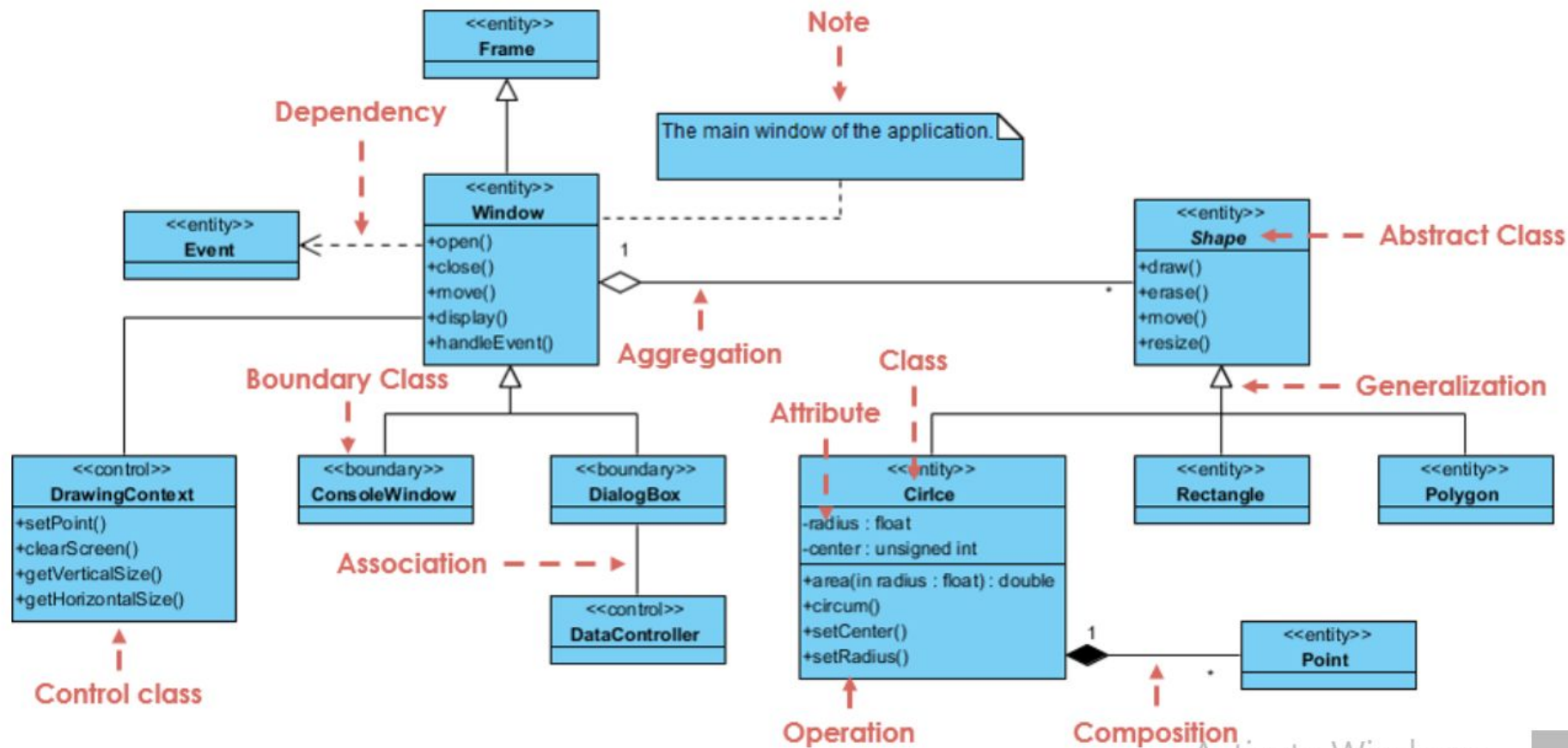












Activate Windows