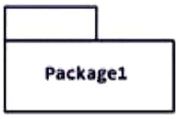
UML Class Diagram Cheatsheet

khalilstemmler.com

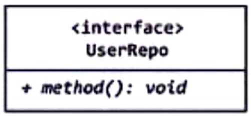
Shape

Description



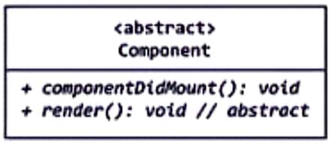
Package

A collection of classes and interfaces.



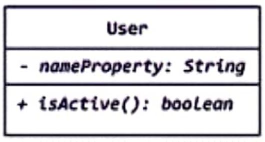
Interface

Interface name written underneath the <interface> annotation. Methods underneath.



Abstract class

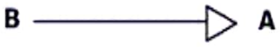
Same as the interface shape. Abstract methods marked as abstract with comments or "abstract methodName(): returnType".



Class

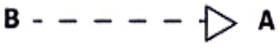
Properties or attributes sit at the top, methods or operations at the bottom + indicates public, - indicates private, and # indicates protected

These should be drawn vertically



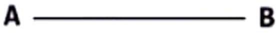
Inheritance

B inherits from A. Creates an "is-a" relationship. A is a generalization.



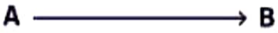
Implementation/realization

B is a concrete implementation/realization of A.



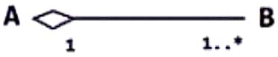
Association

A and B call each other.



One way association

A can call B's properties/methods, but not vice versa.

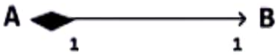


Aggregation

A has 1 or more instances of B. B can survive if A is disposed.

Ex: Professor (1) "has-many" classes (0..) to teach.*

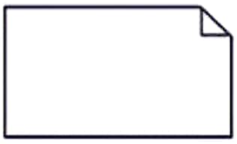
Ex: Pond (0..1) "has-many" ducks (0..). Ducks can survive if the pond is destroyed.*



Composition

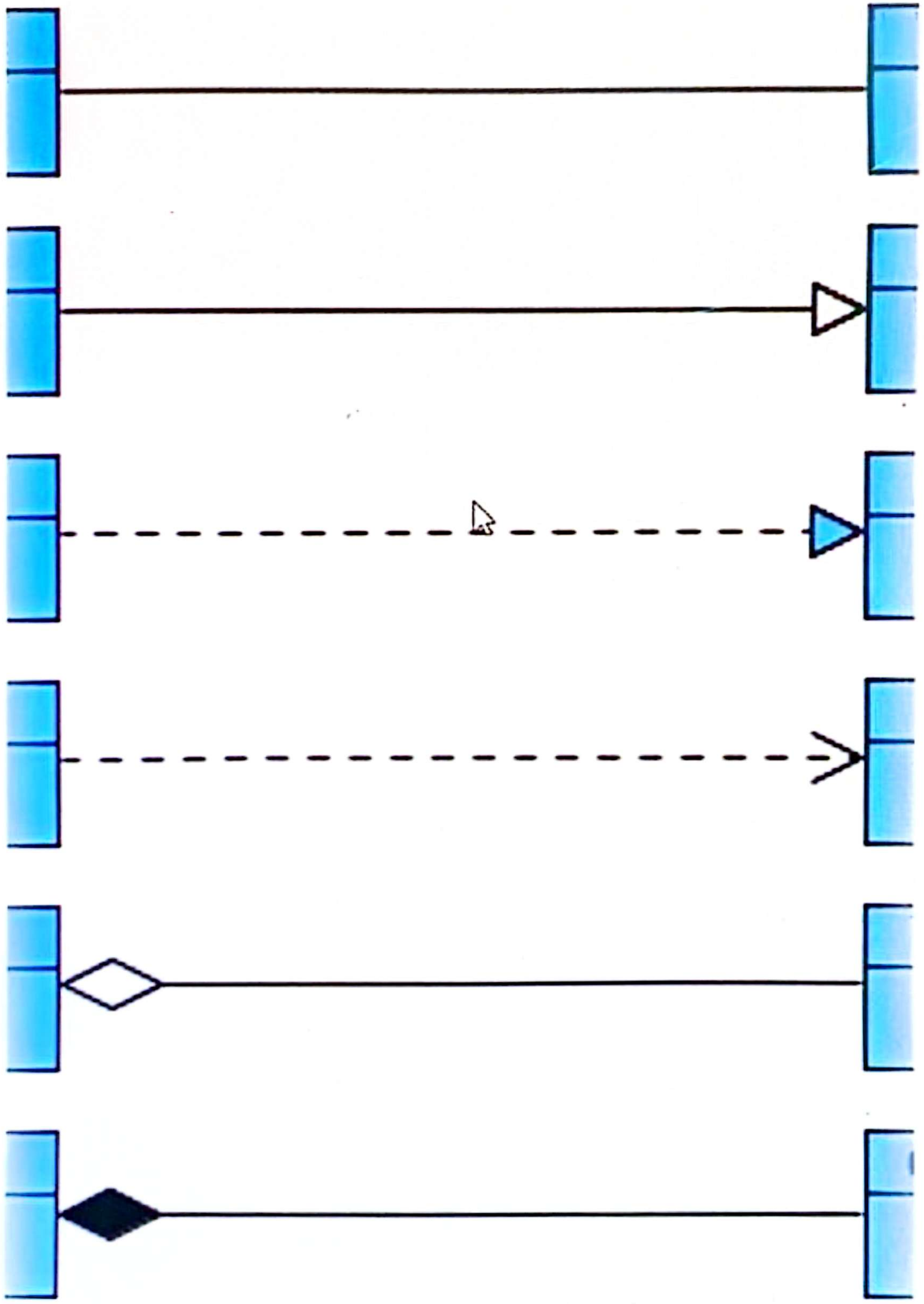
A has 1 or more instances of B. B cannot survive if A is disposed.

Ex: User (1) "has a" UserName (1). UserNames can't exist as separate parts in away from a User in our application.



Note

Descriptive text that can be attached to any item.



Association

Inheritance

Realization

Dependency

Aggregation

Composition