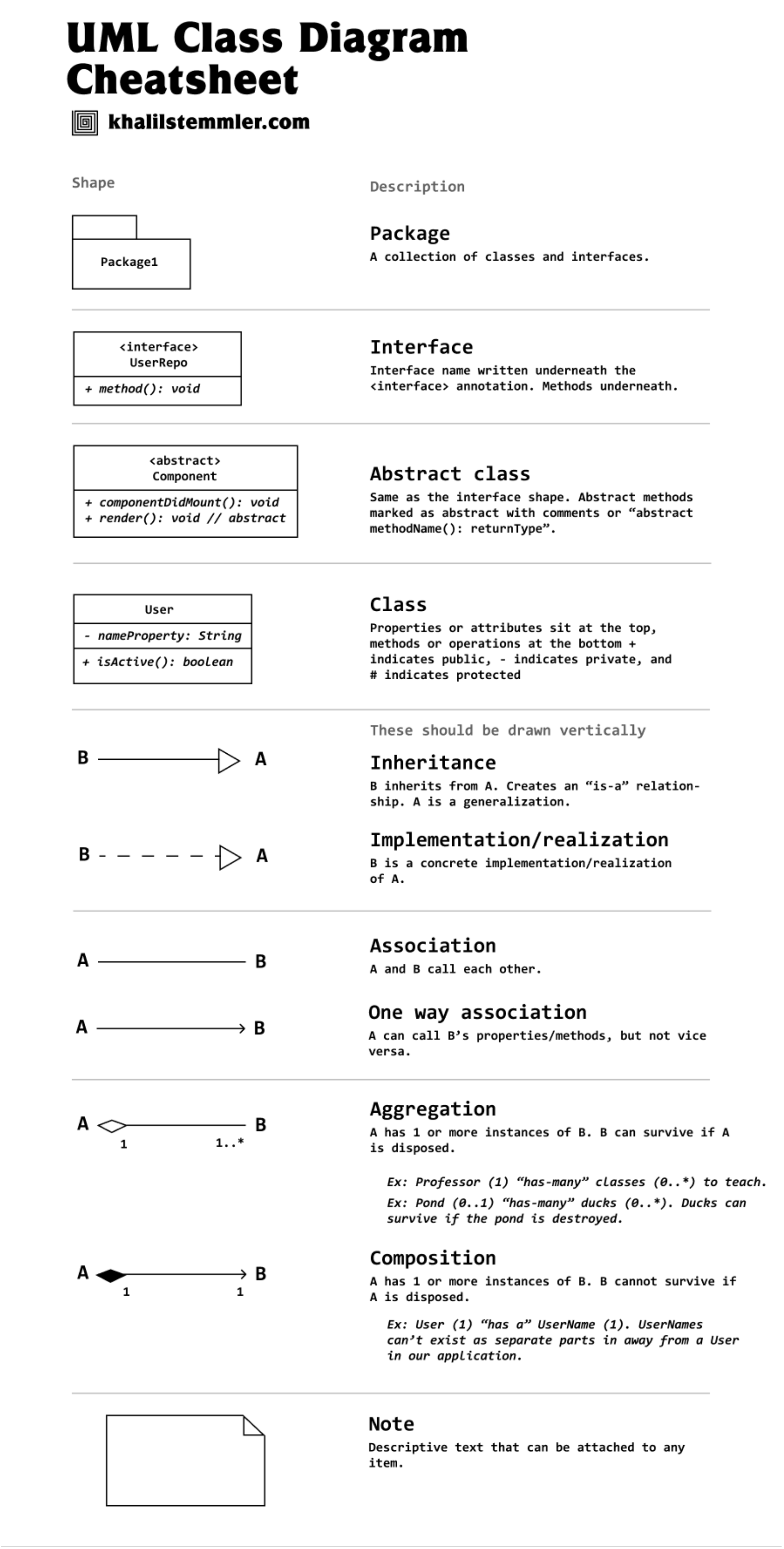

UML Class Diagram Cheatsheet

khalilstemmler.com

Shape

Description

Package1

Package

A collection of classes and interfaces.

<interface>
UserRepo

+ method(): void

Interface

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

<abstract>
Component

+ componentDidMount(): void
+ render(): void // abstract

Abstract class

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

User

- nameProperty: String

+ isActive(): boolean

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

B A

Inheritance

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

B A

Implementation/realization

B is a concrete implementation/realization of A.

A B

Association

A and B call each other.

A B

One way association

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

A B
1 1..*

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.*

A B
1 1

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.