

UML Class Diagrams for High Level Design

Created by Derek Thrasher, last modified by Peter MacLellan on Oct 18, 2017

Arrows Matter!

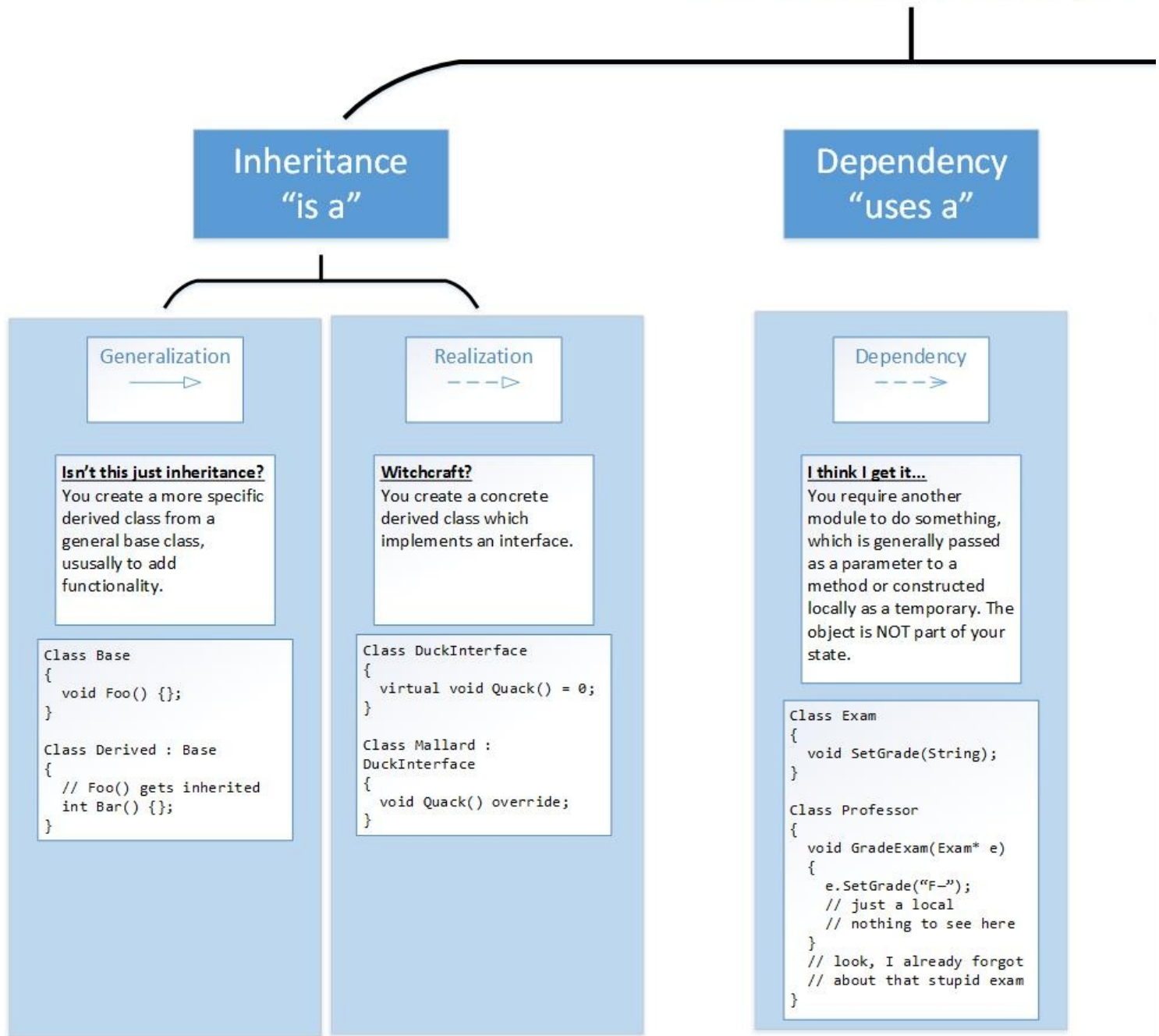
Precise class diagrams help improve the accuracy of our design reviews! For example, if a reviewer sees an aggregation arrow, they know to ask where you intend the reference to the aggregated object to come from (e.g. reference passed to Initialize() by X module)

Great Composition vs. Aggregation Example from Wikipedia

For example, a **university owns various departments** (e.g., chemistry), and each department has a number of professors. If the university closes, the departments will no longer exist, but the **professors in those departments will continue to exist**. Therefore, a **University can be seen as a composition of departments**, whereas **departments have an aggregation of professors**. In addition, a Professor could work in more than one department, but a department could not be part of more than one university.

[OOD Class Relationships.vsd](#)

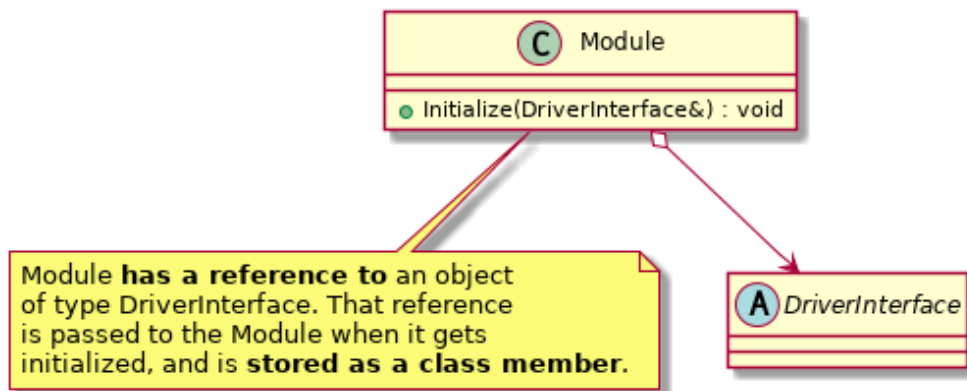
Object-Oriented Class Relationships



Containment Relationships

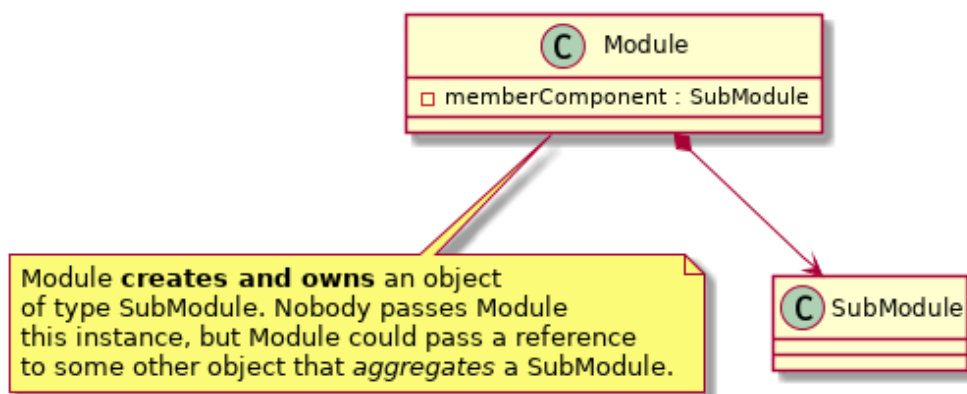
Aggregation

In PlantUML, this is written as "o->" (lowercase O for hollow diamond)



Composition

In PlantUML, this is written as "*"-">" (asterisk for filled in diamond)

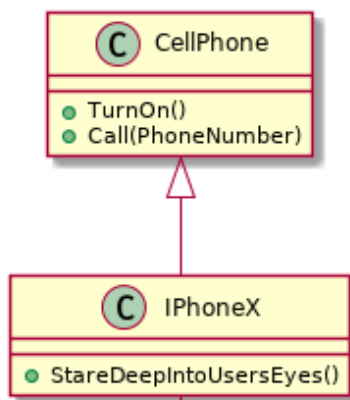


Inheritance Relationships

Generalization/Extension

- ❖ **Favor Realization or better yet Containment**
Whenever possible, we should not be inheriting from non-abstract base classes. This can lead to messy nested inheritance trees if we're not careful, and ain't nobody got time for that.

In PlantUML, this is written as "-|>" (hyphen pipe arrow)

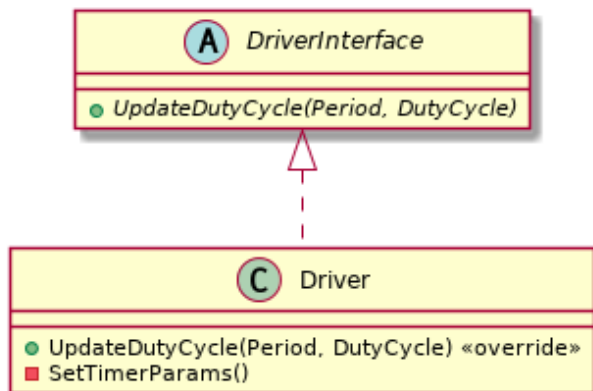


An iPhone X **is a** CellPhone, so it **inherits** the ability to TurnOn and Call, but also **extends** the functionality by allowing face identification by exposing the StareDeepIntoUsersEyes function.

Realization/Implementation

- ❶ The most common use of realization is for run-time polymorphism, which is when a client interacts with a concrete object through a set of functions defined in one of the abstract interfaces it implements.

In PlantUML, this is written as ".|>" (dot pipe arrow)



This Driver **implements** the DriverInterface. Clients can now use a DriverInterface without knowing the specific object that implements it. This **run-time polymorphism**

[class](#) [diagram](#) [uml](#) [design](#) [architecture](#)
[best_practices](#) [software](#)

3 Comments



Dominic Harkness

[@Derek Thrasher](#) , is this something that would be useful to copy into the Systems Engineering space so Prometheus/Perseus people can use it as a reference?



Derek Thrasher

I'd rather a move to [Software Development All Access](#). We should avoid copying if the information is going to end up staying the same



Dominic Harkness

Perhaps here? [Software Design](#)
