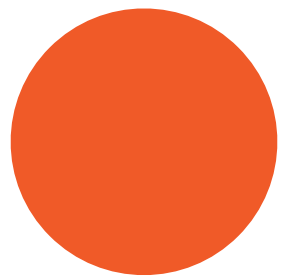
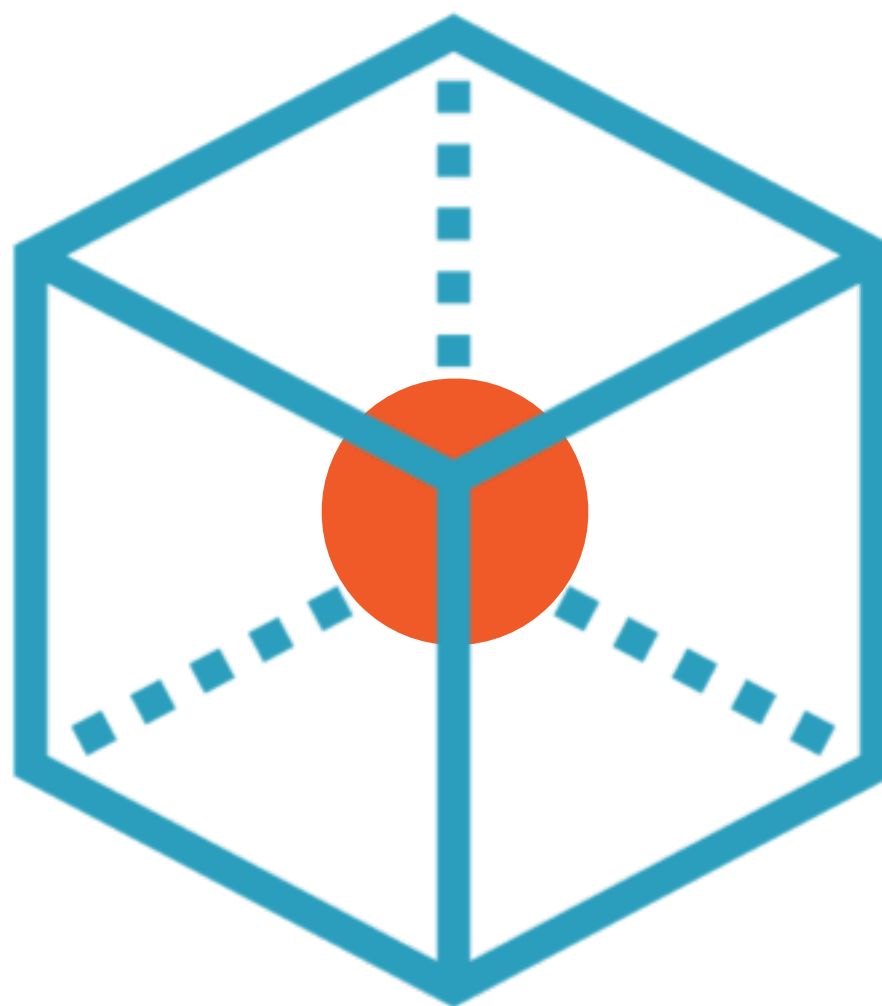


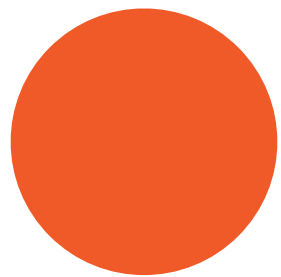
The Decorator

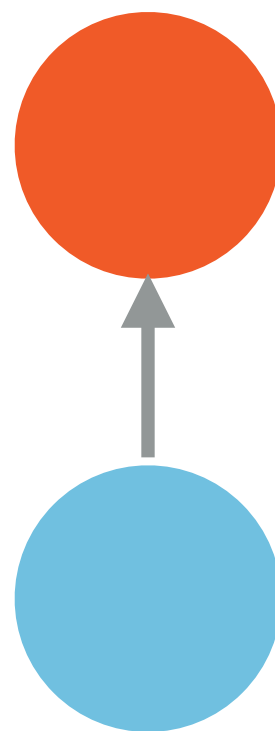
Decorator

Adds new behavior to objects dynamically.





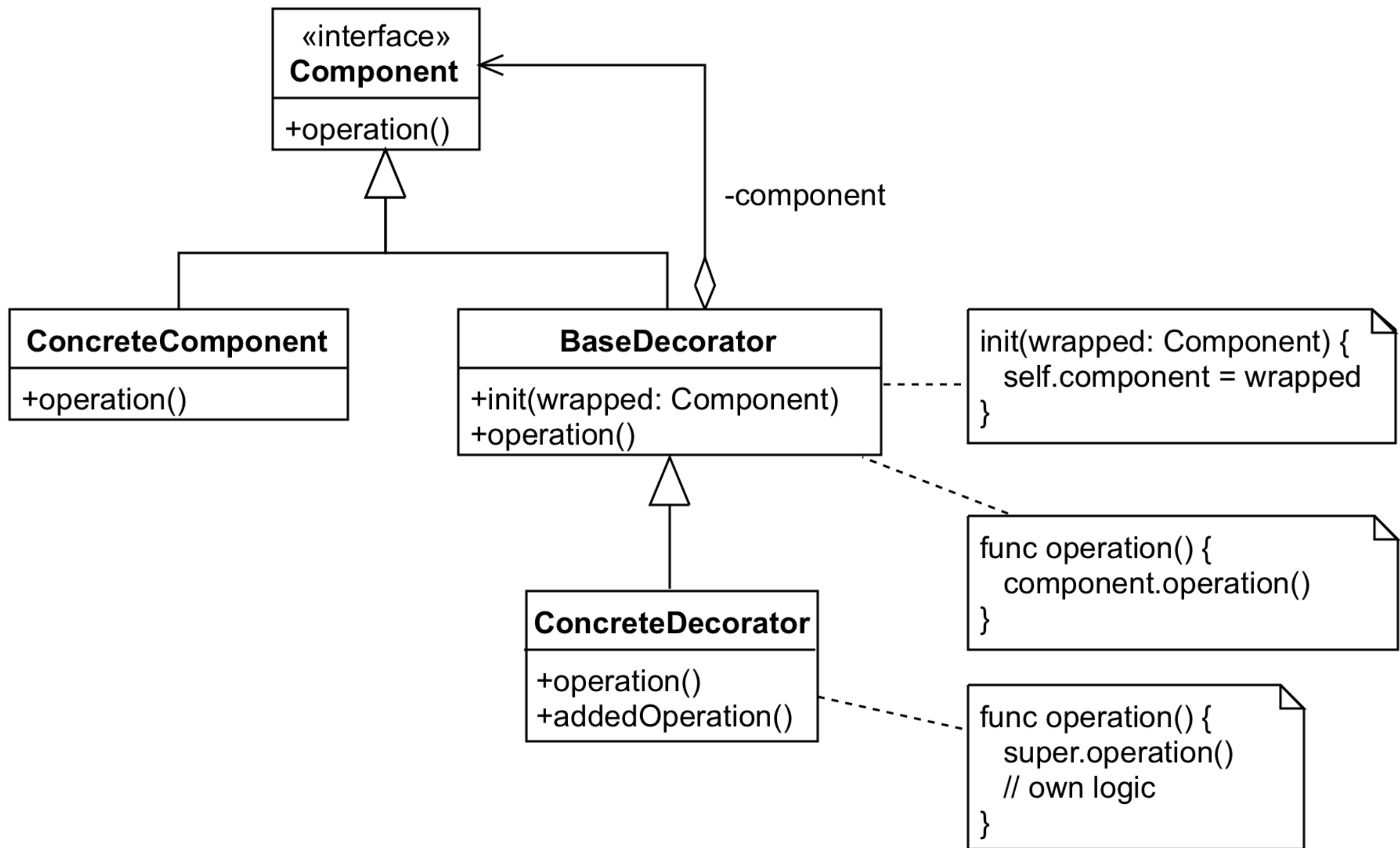




Inheritance pitfalls

Doesn't work with value types

Lack of dynamic behavior



Demo

CameraShop





Mirrorless



Mirrorless
Compact



Mirrorless

Compact

DSLR

Camera
upgrades

View finder

Video recording

LCD

Mirrorless

View finder

Mirrorless

View finder

Video recording

Mirrorless

View finder

Video recording

LCD

Mirrorless

View finder

Video recording

LCD

View finder and video recording

Mirrorless

View finder

Video recording

LCD

View finder and video recording

Video recording and LCD

Mirrorless

View finder

Video recording

LCD

View finder and video recording

Video recording and LCD

View finder and LCD

Mirrorless

View finder

Video recording

LCD

View finder and video recording

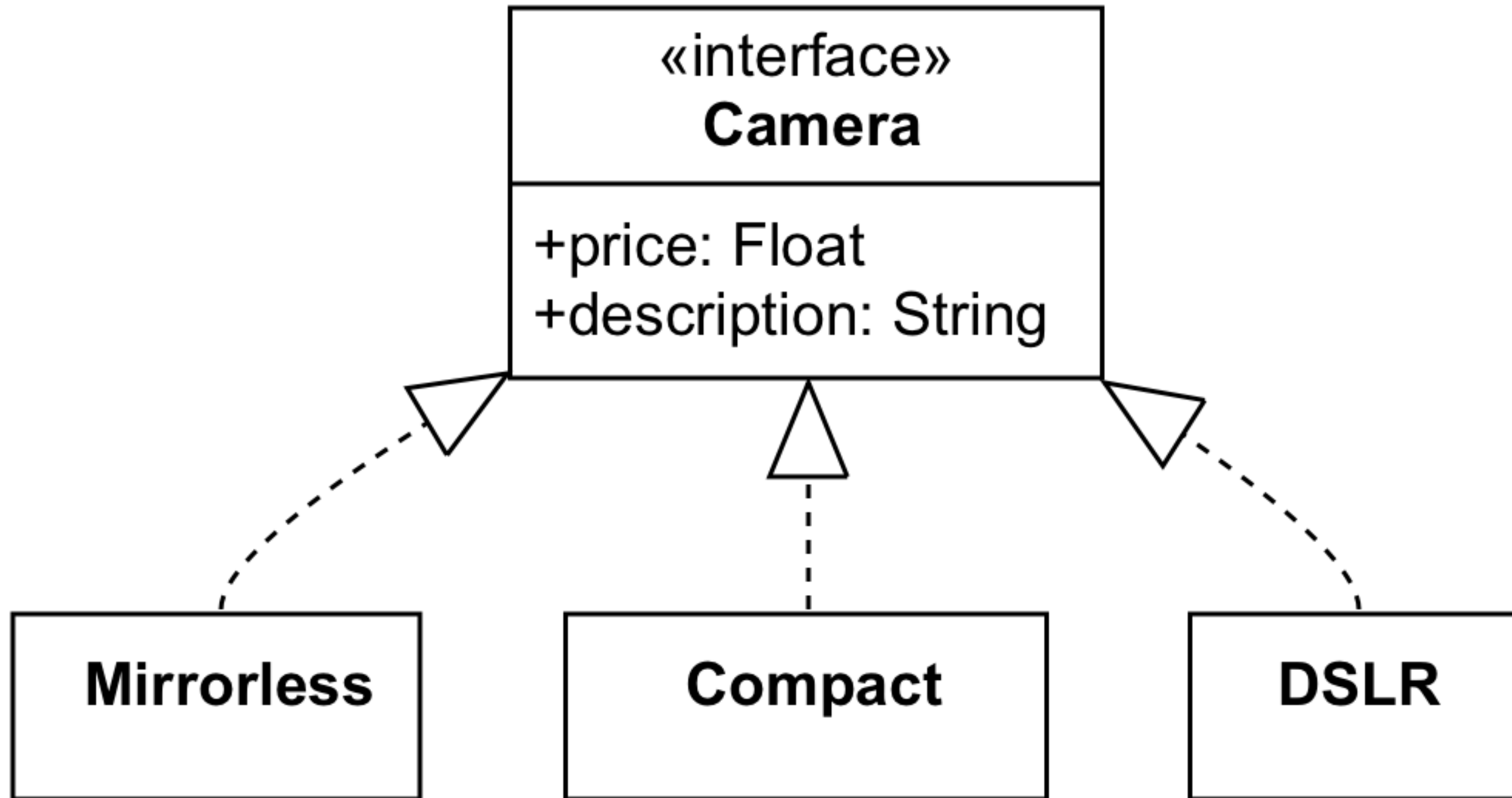
Video recording and LCD

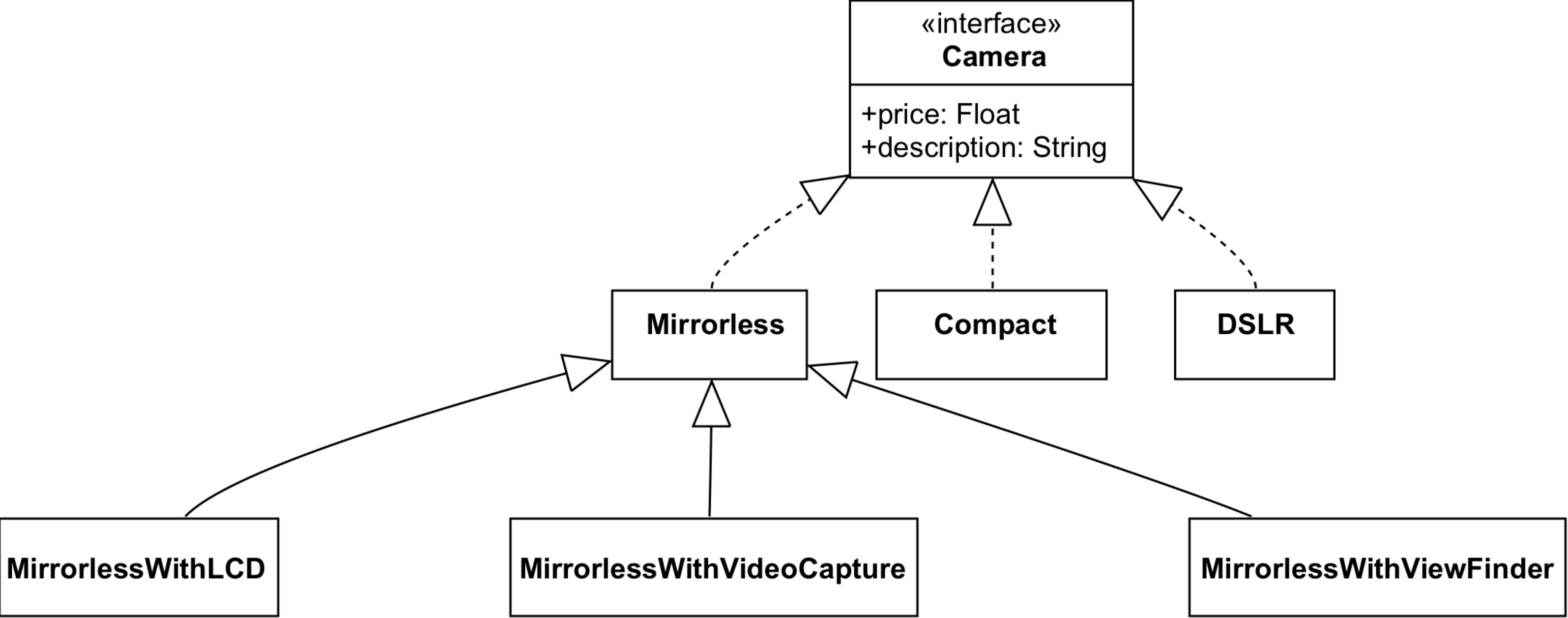
View finder and LCD

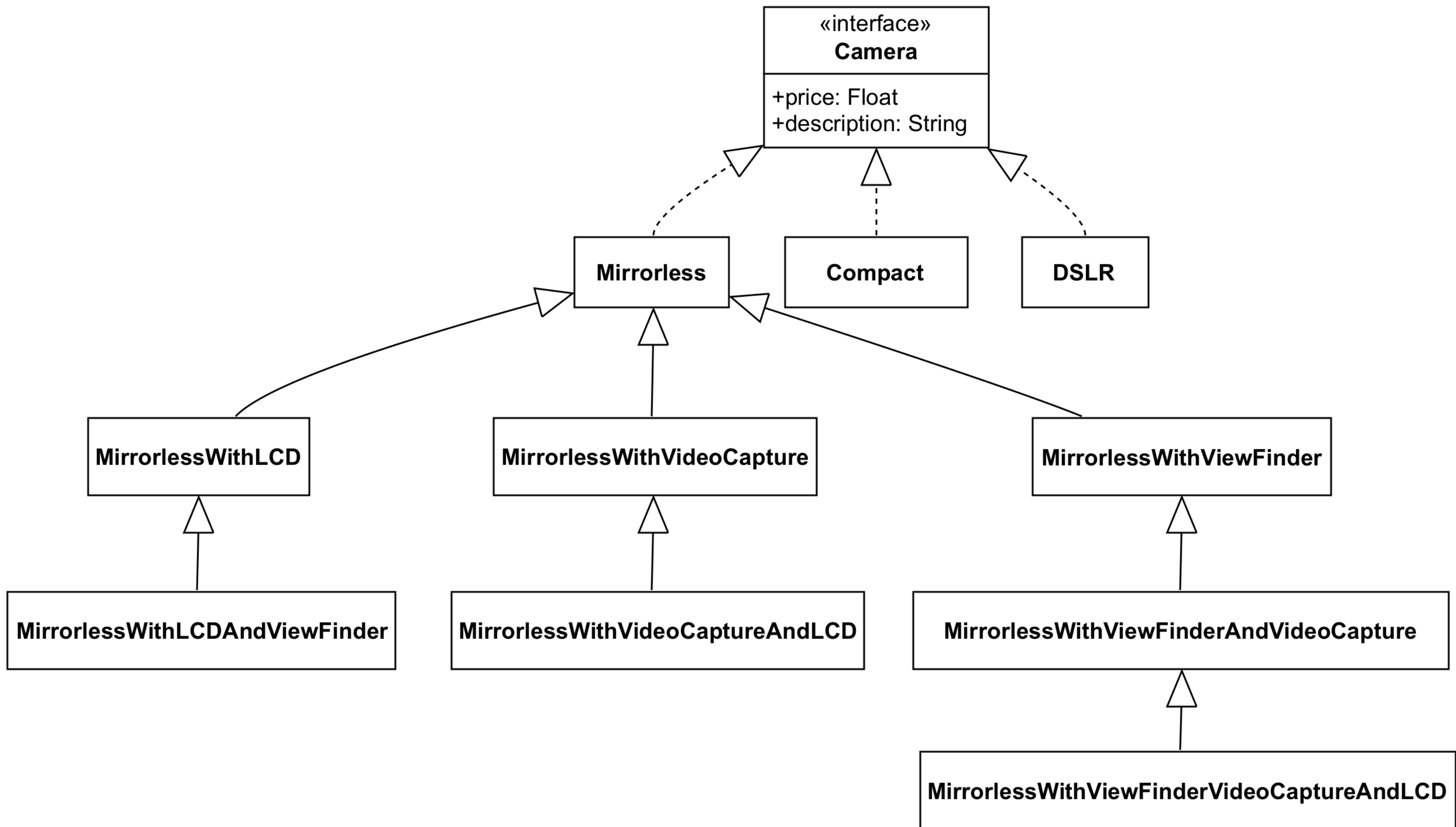
View finder, video recording and LCD

«interface»
Camera

+price: Float
+description: String

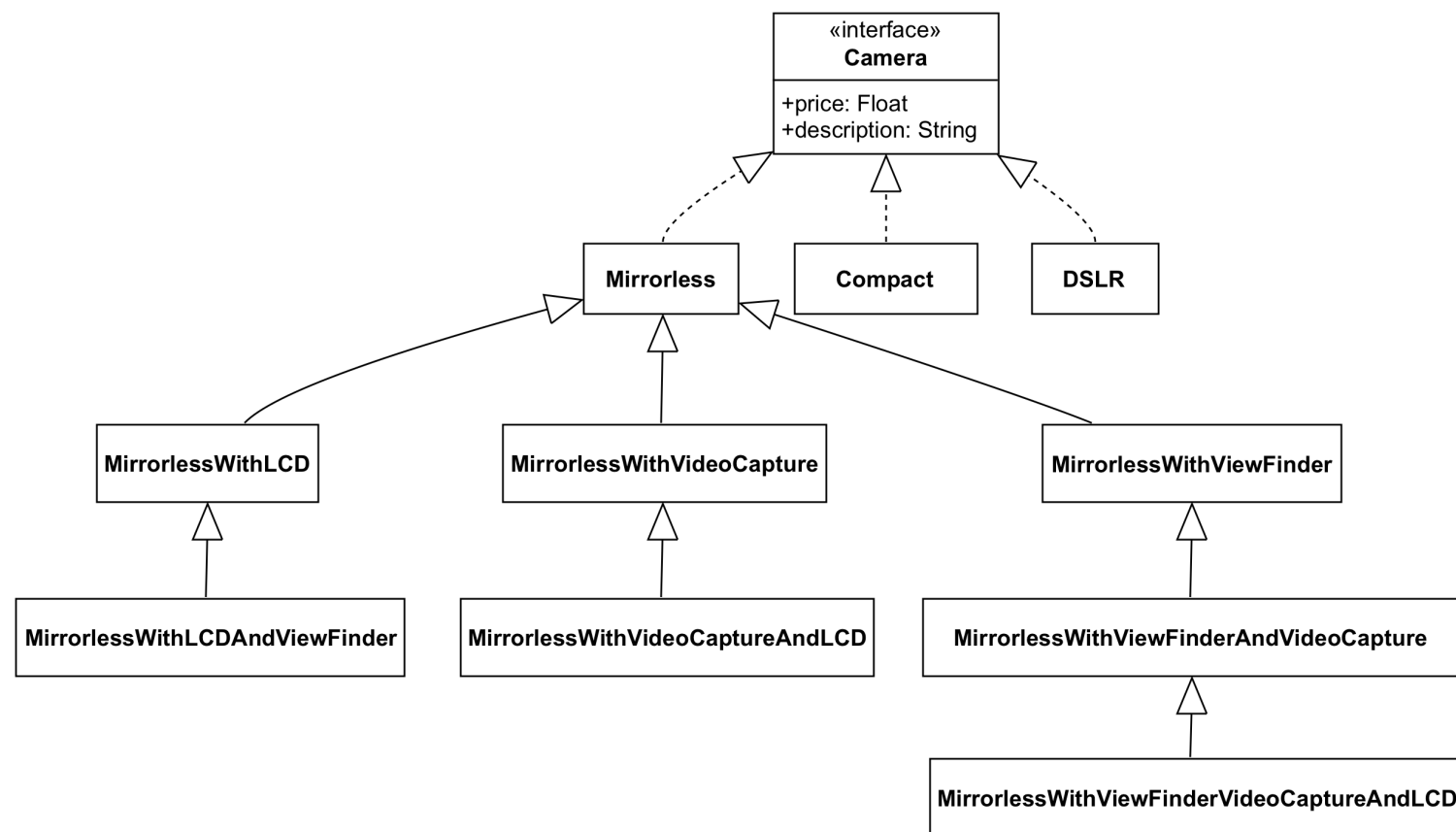




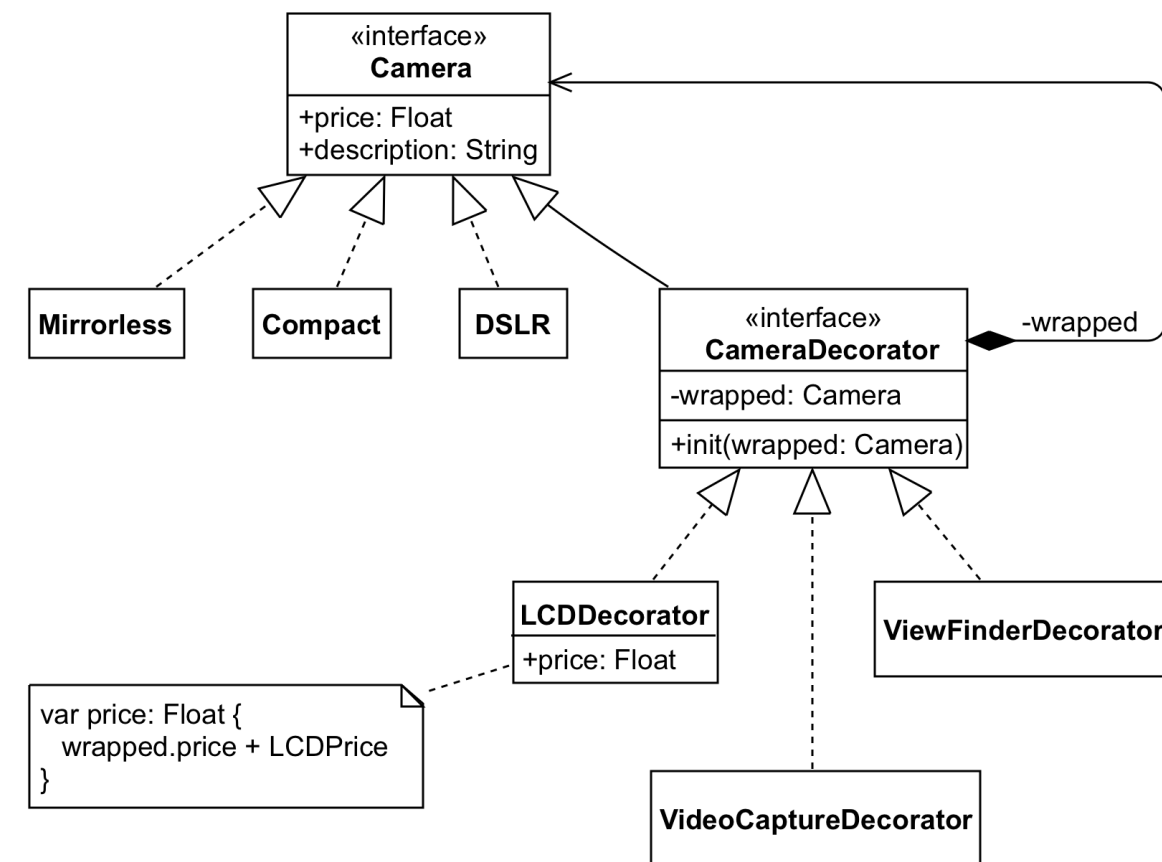


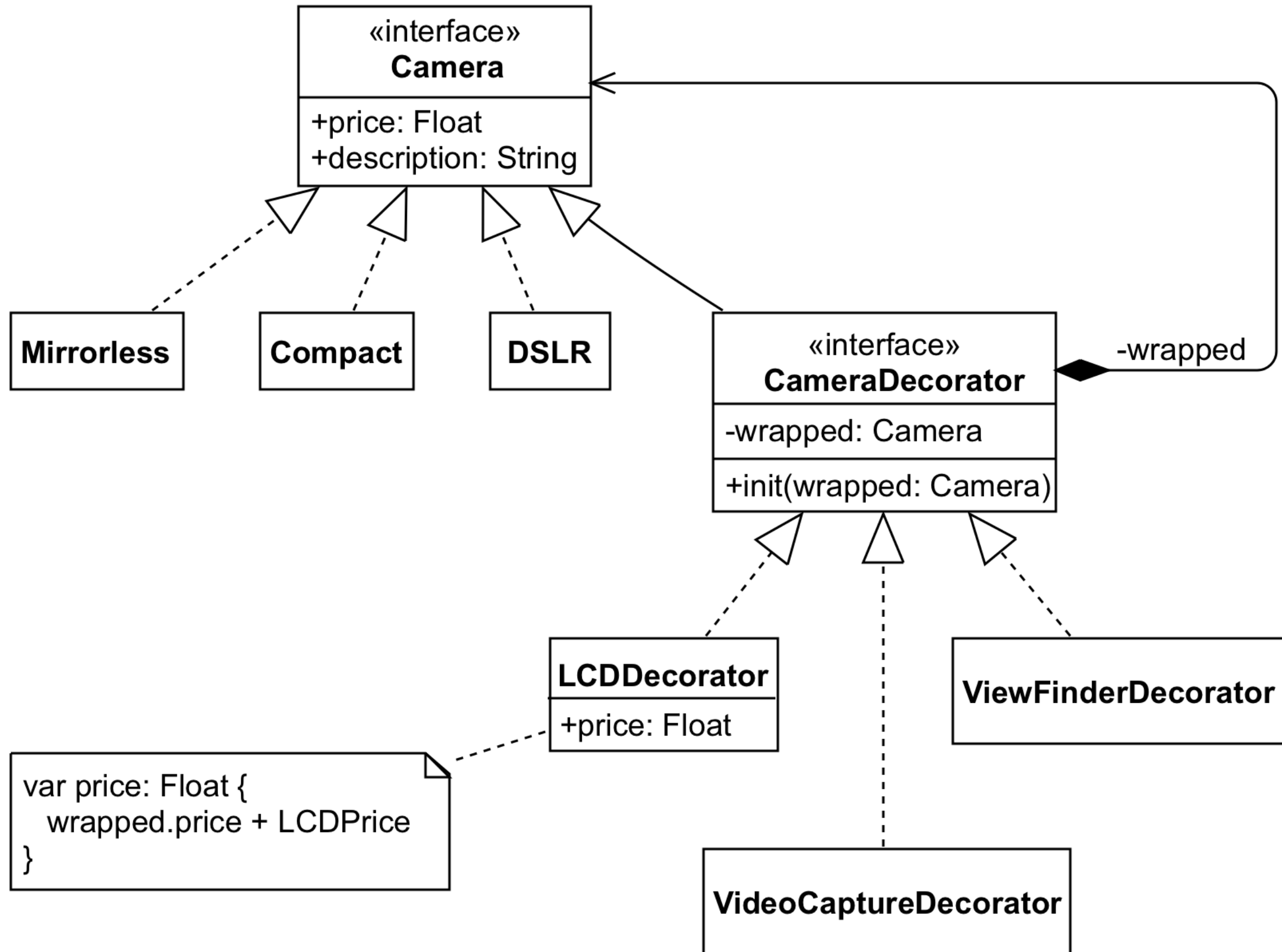
Refactoring

Inheritance



Decorator





Summary

The Decorator Pattern:

- Attaches new responsibilities to objects
- Relies on object composition
- Can add multiple behaviors through recursive composition

Pitfalls

- Proliferation of decorators