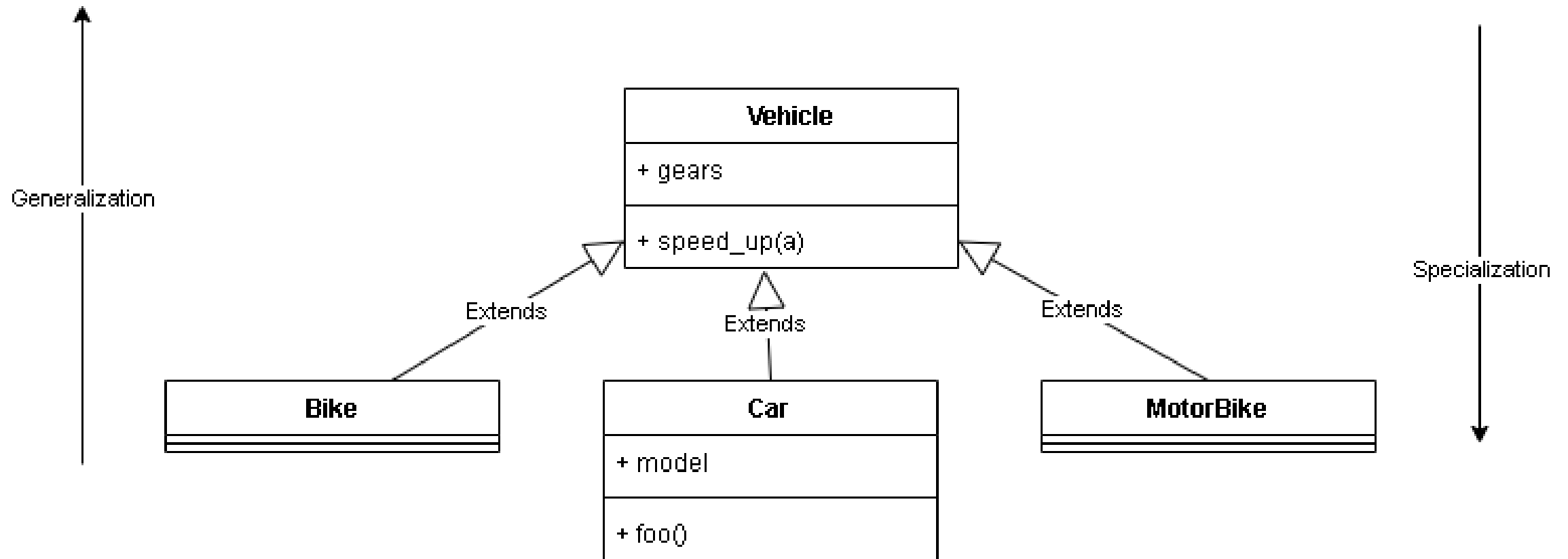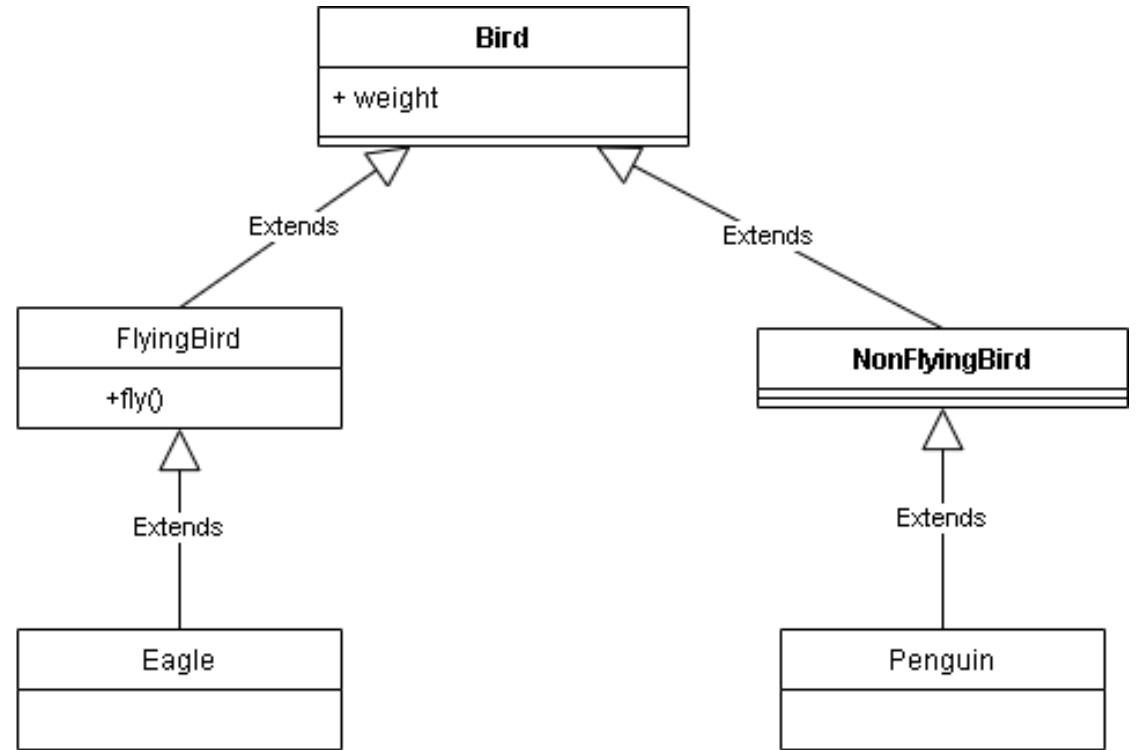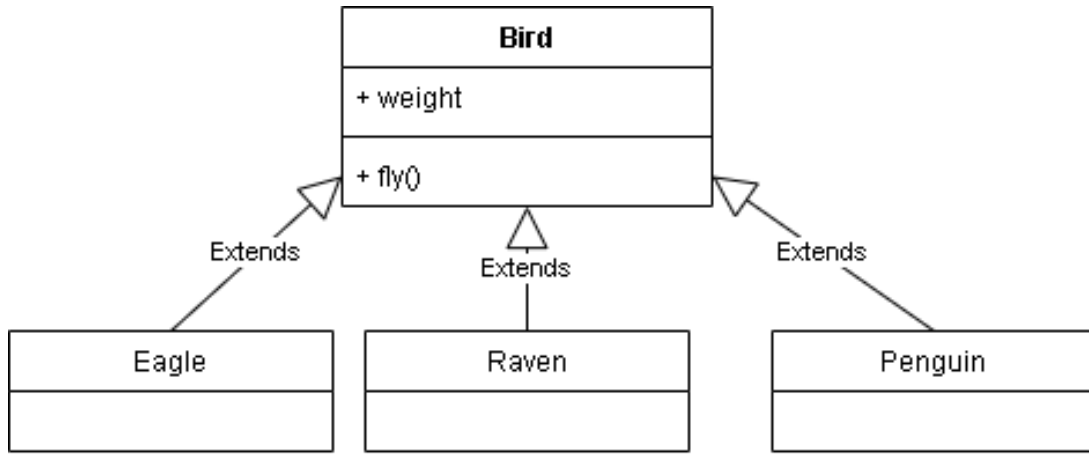# Software Engineering

Notes 27/10/2022
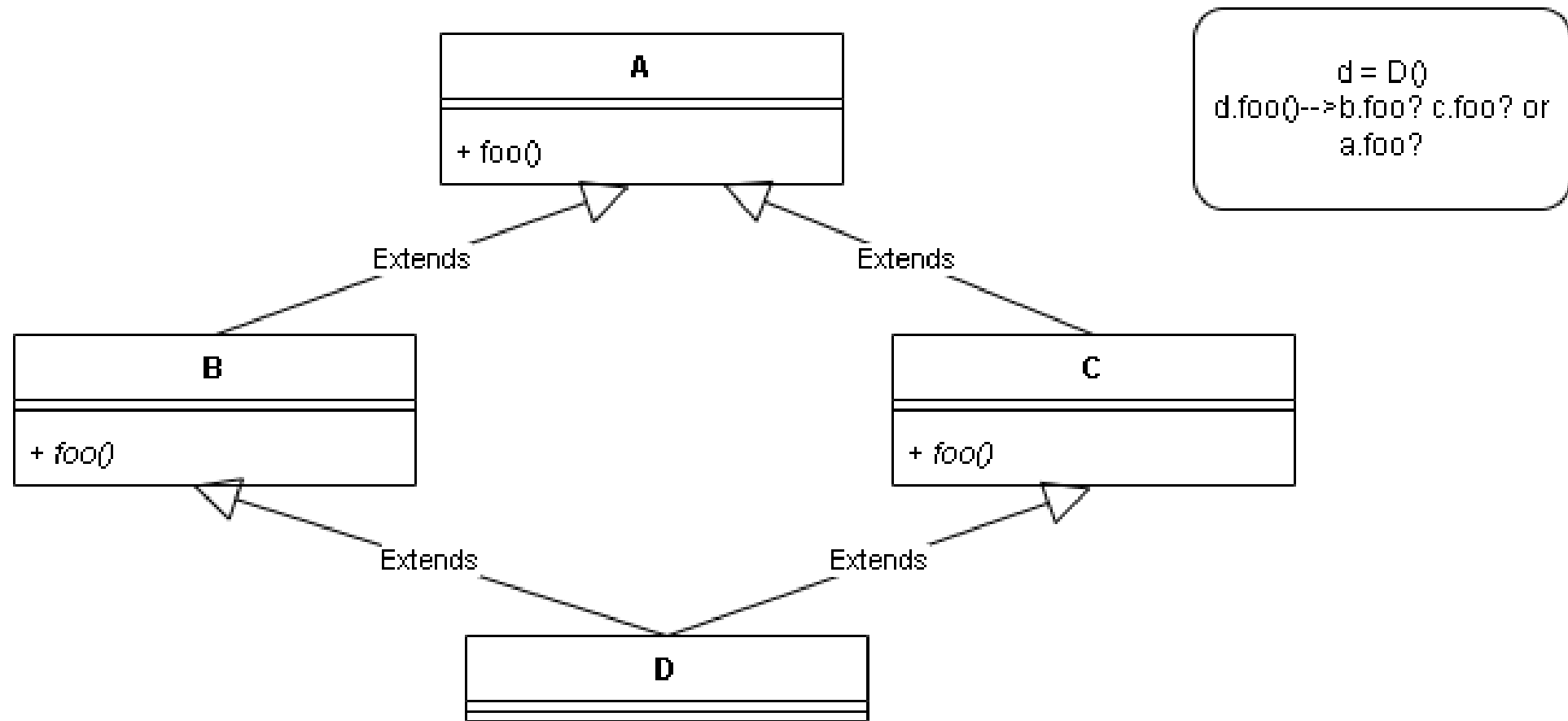
Inheritance generation
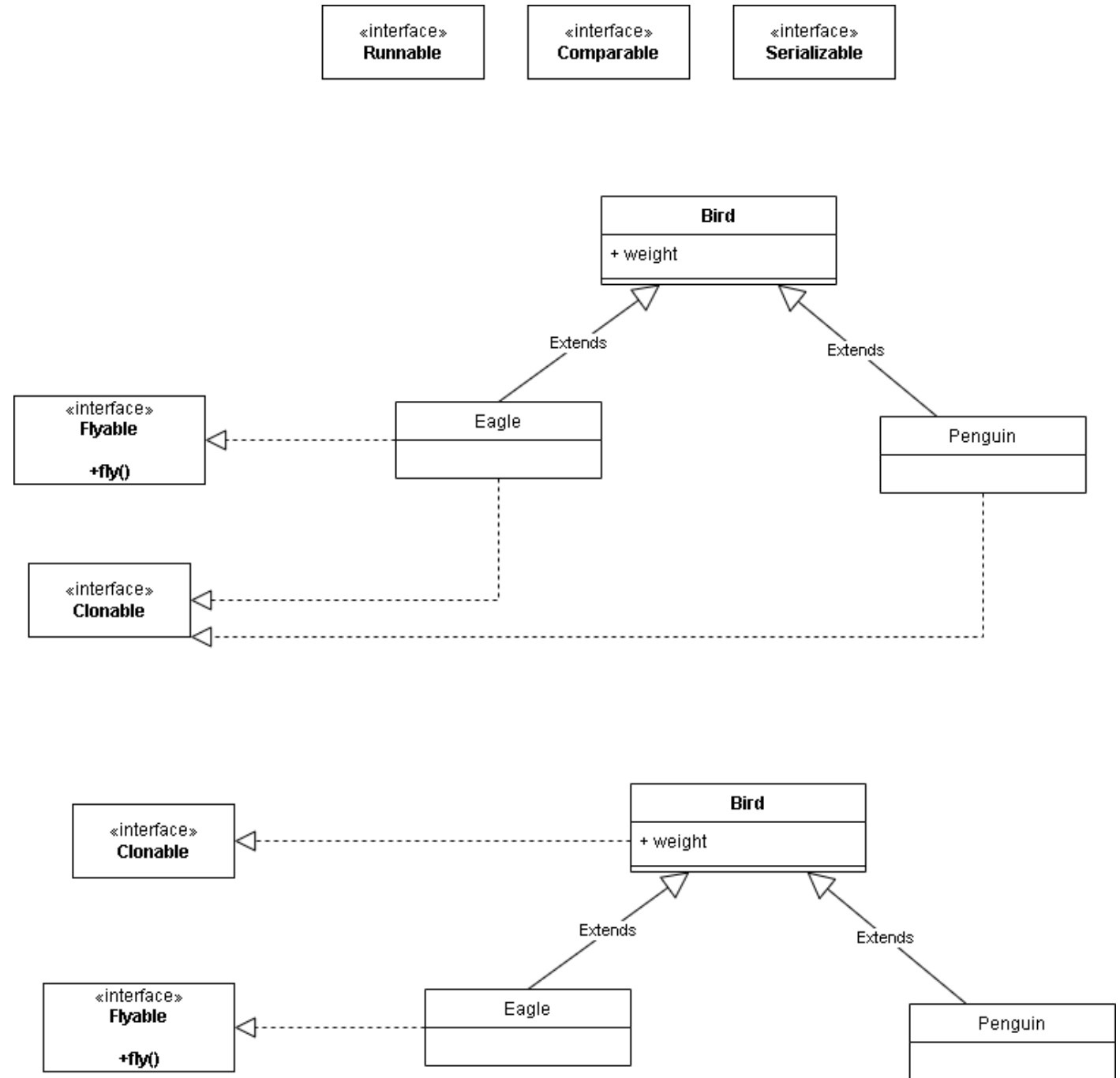
Inheritance: reuse data and/or behavior

Inheritance: diamond problem

Interface

«interface»
Runnable

«interface»
Comparable

«interface»
Serializable

Bird

+ weight

Extends

Extends

«interface»
Flyable

+fly()

Eagle

Penguin

«interface»
Clonable

Bird

+ weight

«interface»
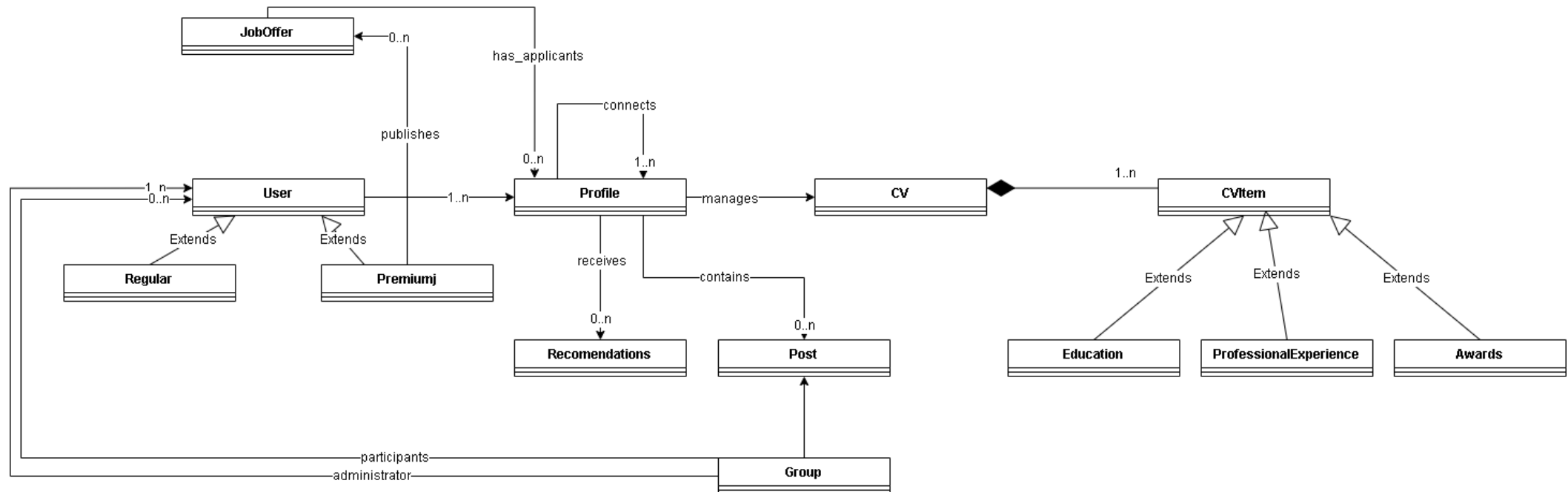**Clonable**

Extends

Extends

«interface»
Flyable

+fly()

Eagle

Penguin

Linkedin: example

SOLID PRINCIPLES

The first five principles are principles of *class design*. They are:

SRP	The Single Responsibility Principle"	A class should have one, and only one, reason to change."
OCP	The Open Closed Principle	*You should be able to extend a classes behavior, without modifying it.*
LSP	The Liskov Substitution Principle	*Derived classes must be substitutable for their base classes.*
ISP	The Interface Segregation Principle	*Make fine grained interfaces that are client specific.*
DIP	The Dependency Inversion Principle	*Depend on abstractions, not on concretions.*

http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod