

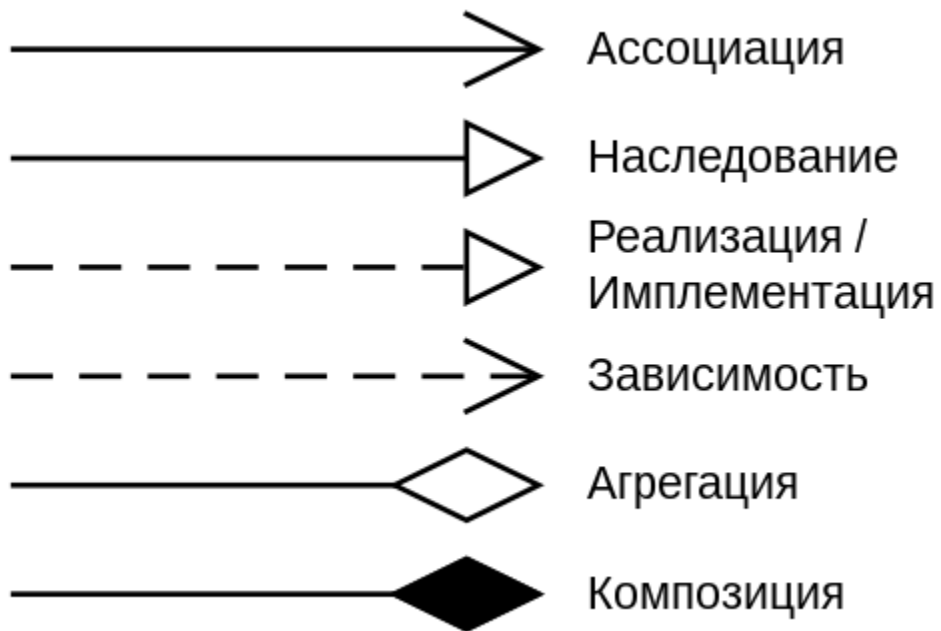


# Объектно-ориентированное программирование

## Лекция 5. Введение в технологию программирования



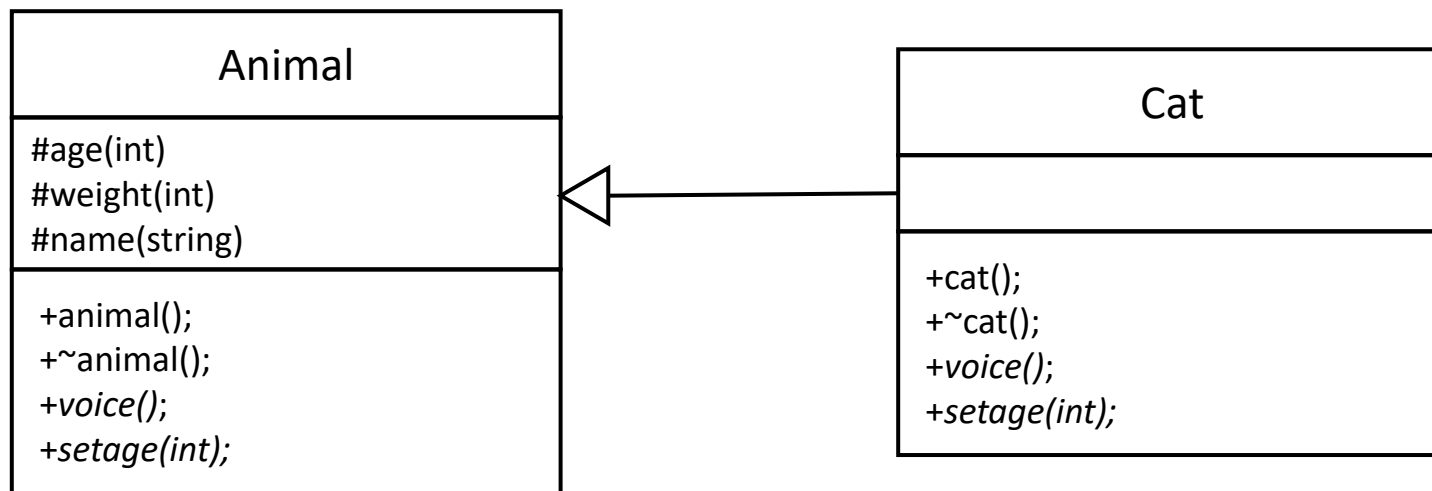
# UML диаграмма классов



+	Публичный (Public)
-	Приватный (Private)
#	Защищённый (Protected)
/	Производный (Derived) (может быть совмещён с другими)
~	Пакет (Package)



# UML диаграмма классов (пример)





# Принципы SOLID

**S**

— **Single Responsibility** (Принцип единственной ответственности)

**O**

— **Open-Closed** (Принцип открытости-закрытости)

**L**

— **Liskov Substitution** (Принцип подстановки Барбары Лисков)

**I**

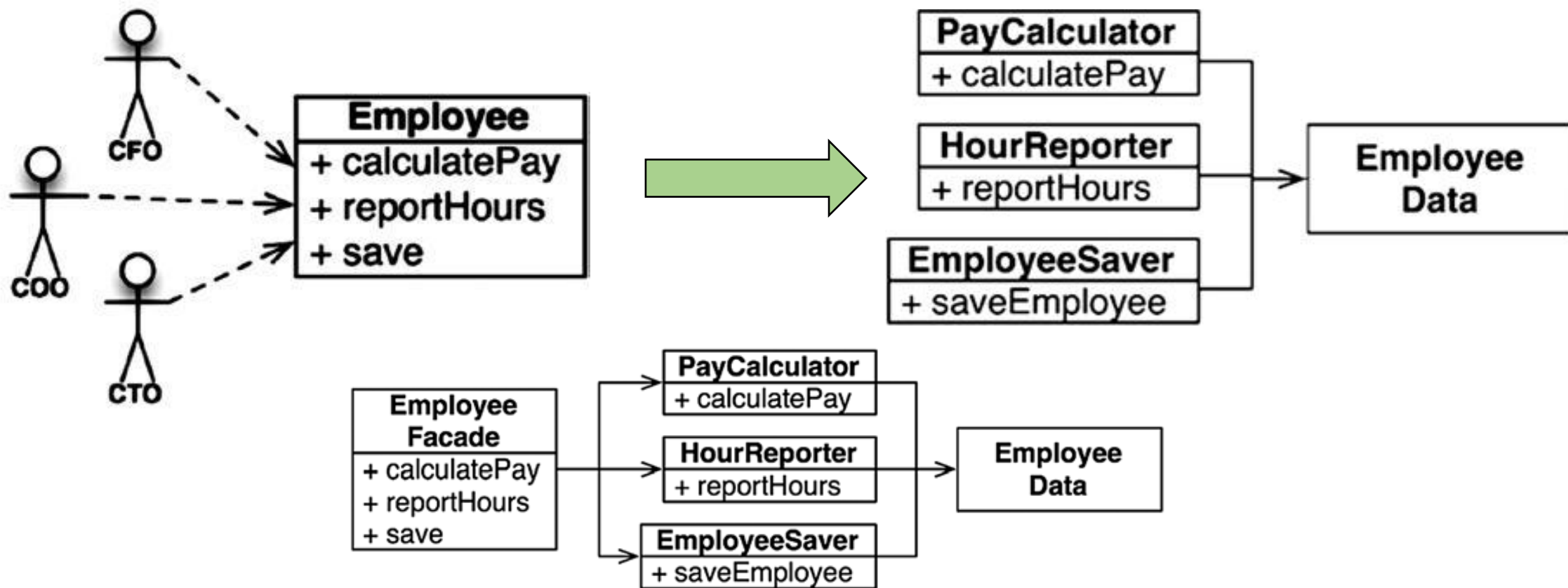
— **Interface Segregation** (Принцип разделения интерфейсов)

**D**

— **Dependency Inversion** (Принцип инверсии зависимостей)



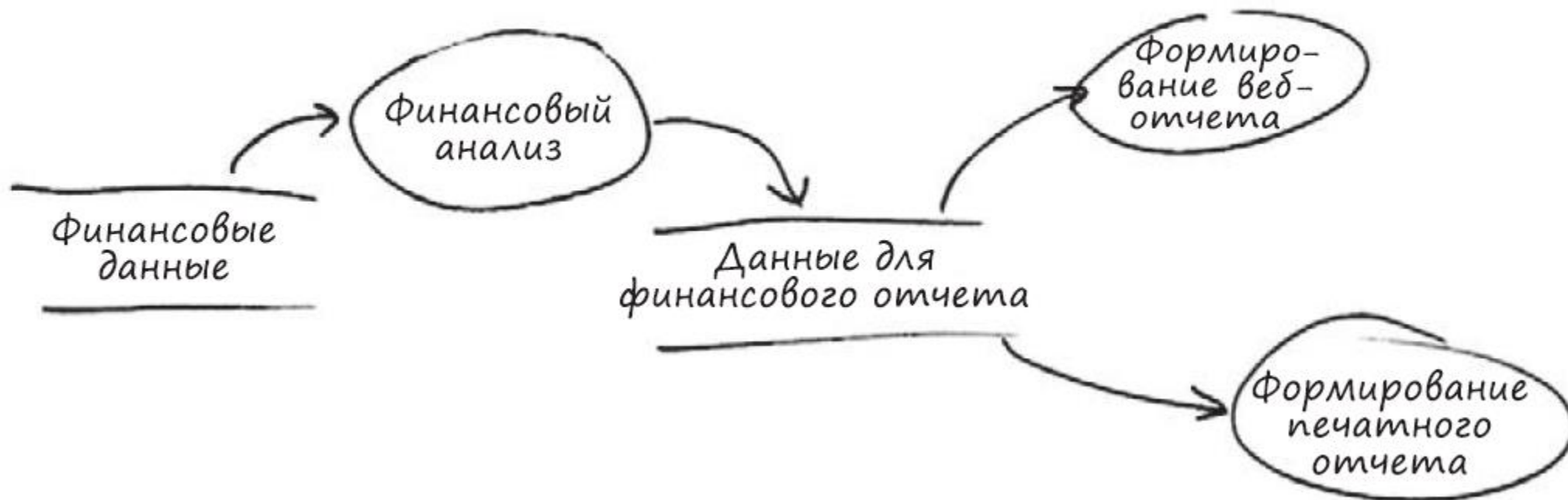
# Принцип единственной ответственности



*Модуль должен иметь одну и только одну причину для изменения*



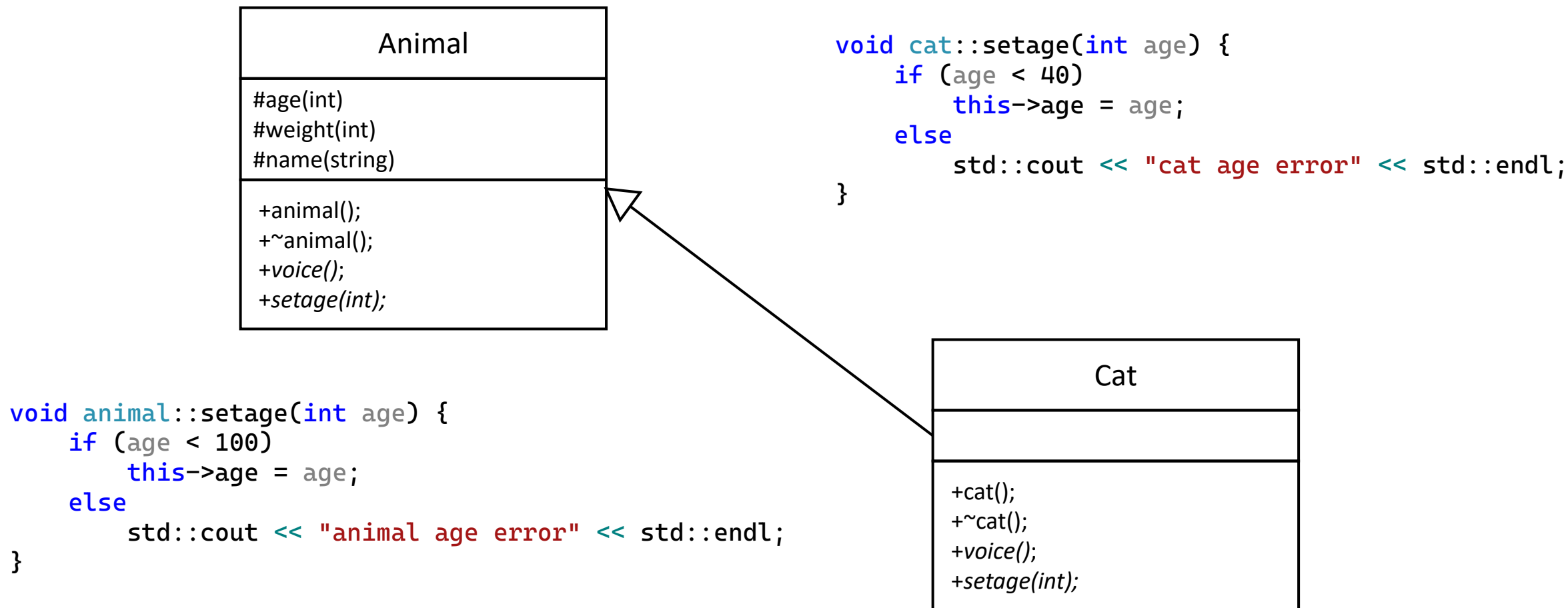
# Принцип открытости/закрытости



*Программные сущности должны быть открыты для расширения и закрыты для изменения*



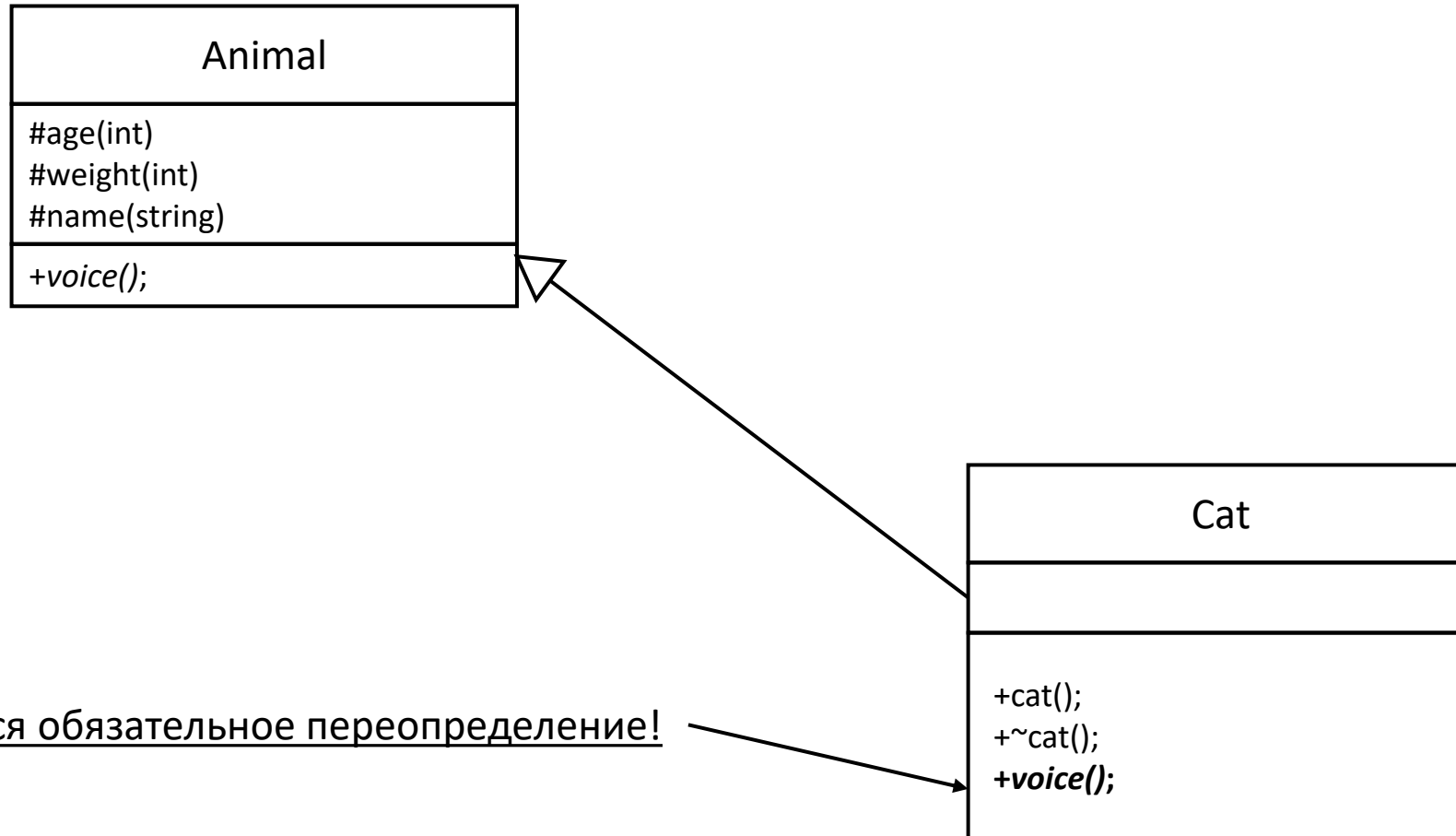
# Принцип подстановки Барбары Лисков



*Подклассы должны дополнять, а не замещать поведение базового класса.*



# Принцип разделения интерфейсов

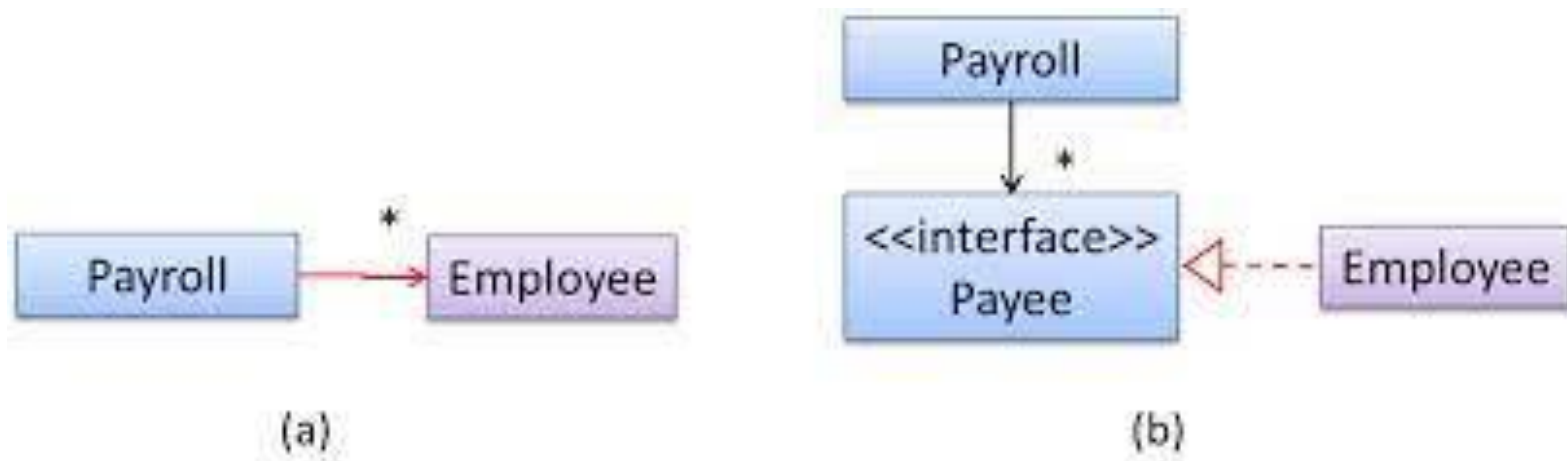


*Клиенты не должны зависеть от методов, которые они не используют.*





# Принцип инверсии зависимостей



Классы верхних уровней не должны зависеть от классов нижних уровней. Оба должны зависеть от абстракций. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.



## Вместо заключения...

