GOF design patterns

Github:

https://bit.ly/3ypgSaQ

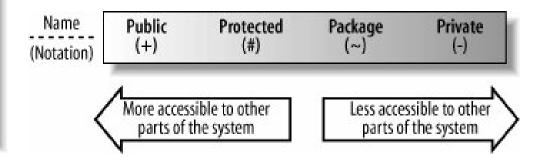


Class Diagram

Class in UML

Guitarist

- instrument : Instrument
- + getInstrument(): Instrument
- + setInstrument(instrument : Instrument) : void
- + play(): void
- + main(args : String[]) : void



ClassName

Attribute Attribute

Operation Operation

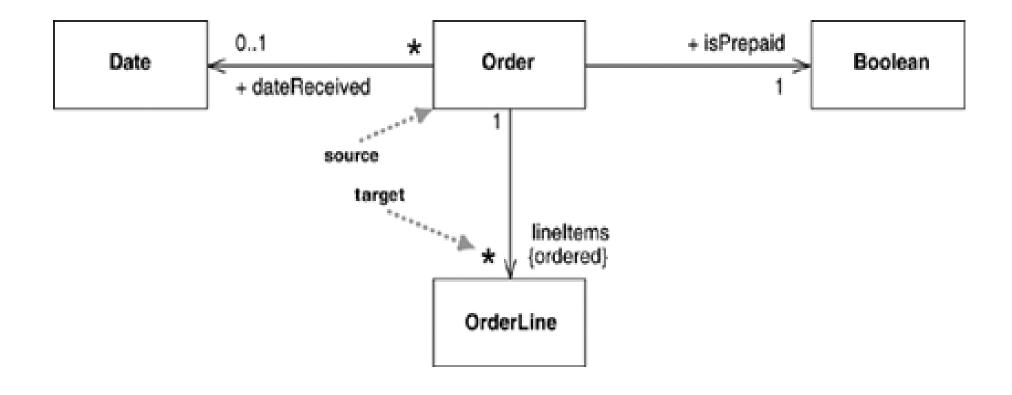
ClassName

Attribute Attribute

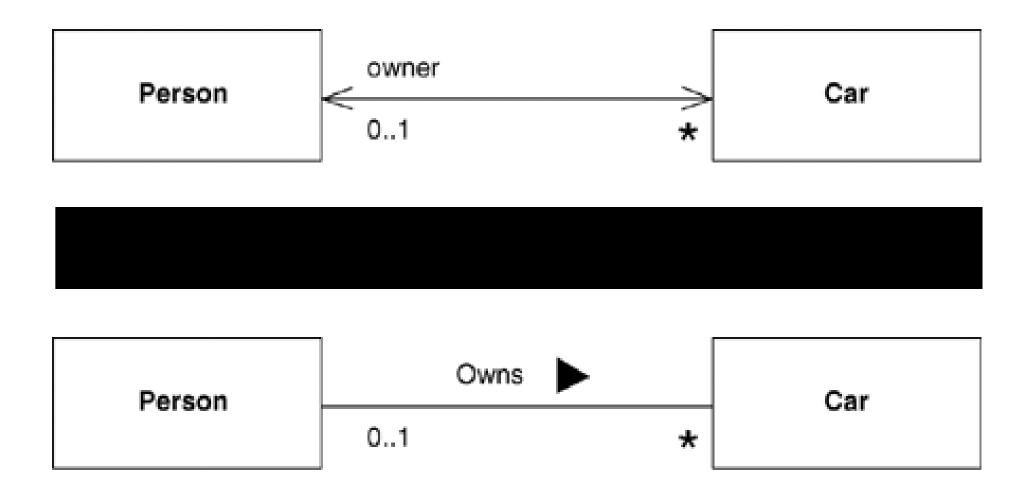
ClassName

Operation Operation ClassName

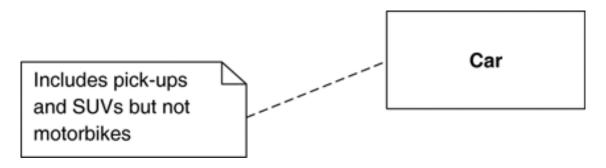
Class Diagram

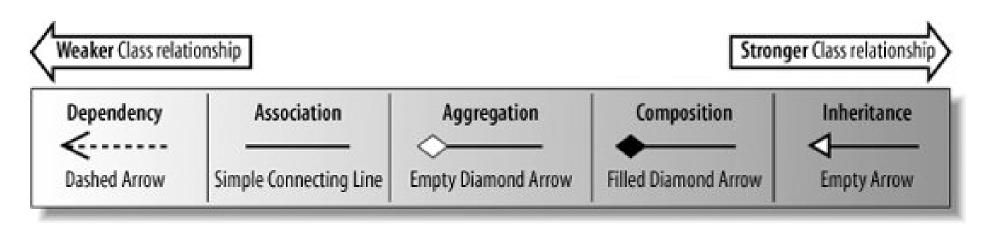


Class Diagram



Comments and Relationships

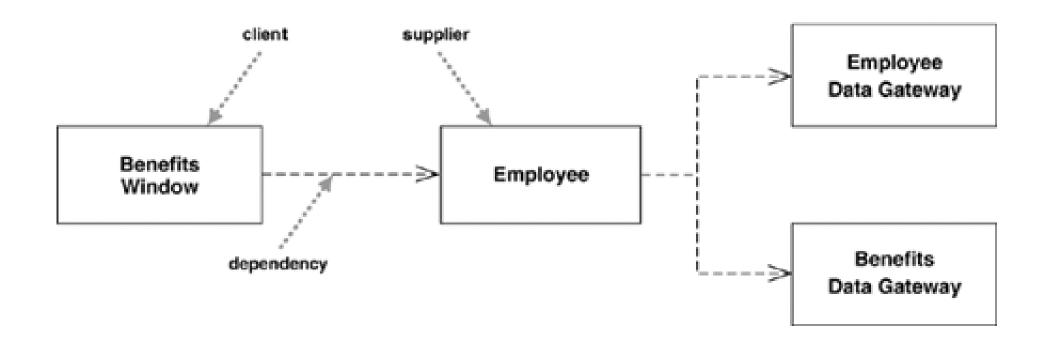


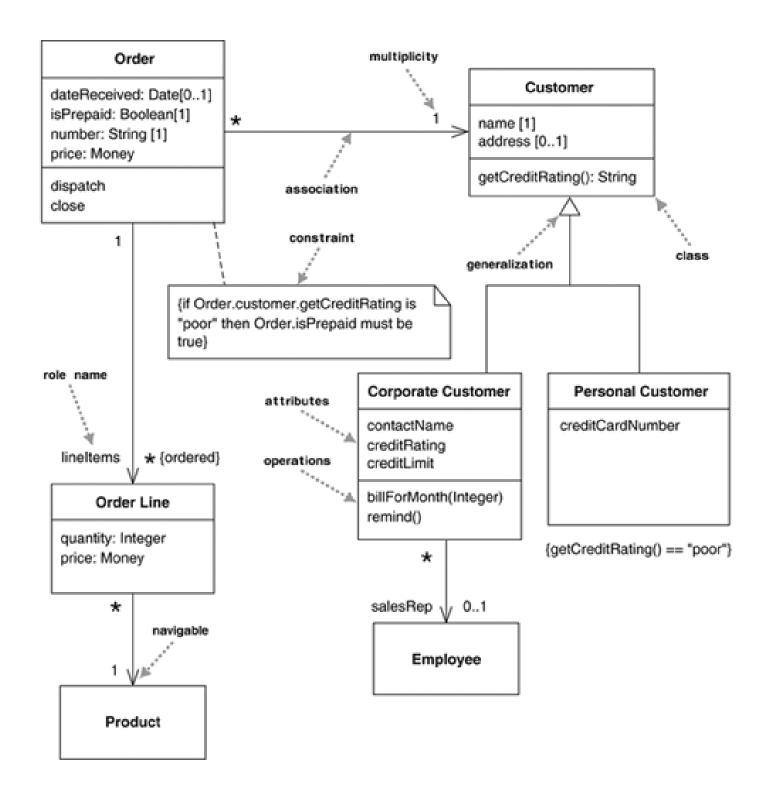


When objects of one class work briefly with objects of another class When objects of one class work with objects of another class for some prolonged amount of time When one class owns but shares a reference to objects of another class

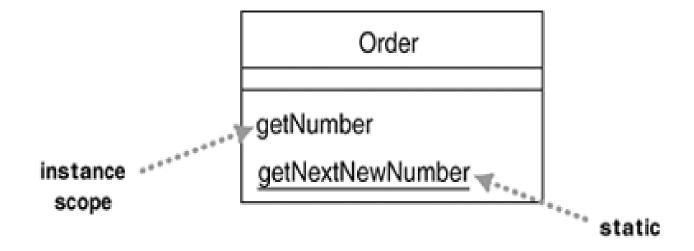
When one class contains objects of another class When one class is a type of another class

Dependency

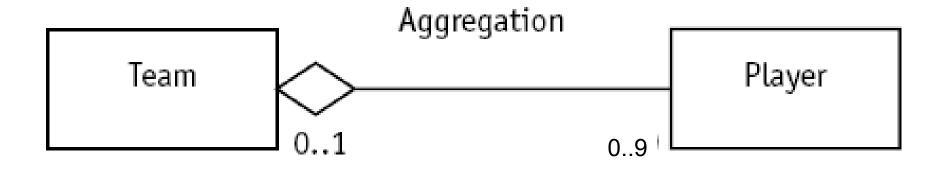


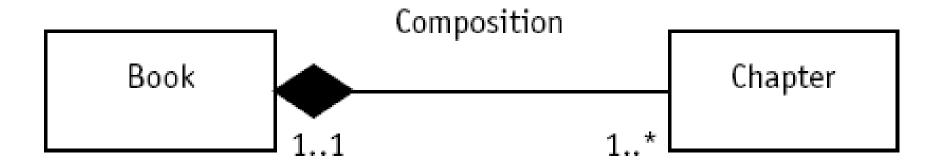


Notation



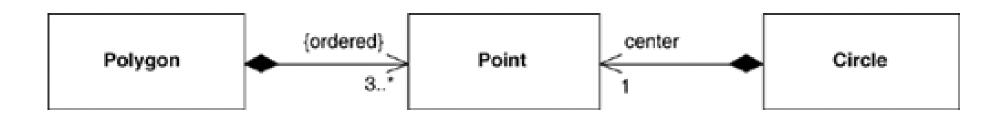
Aggregation & Composition





Aggregation & Composition





Interface notation

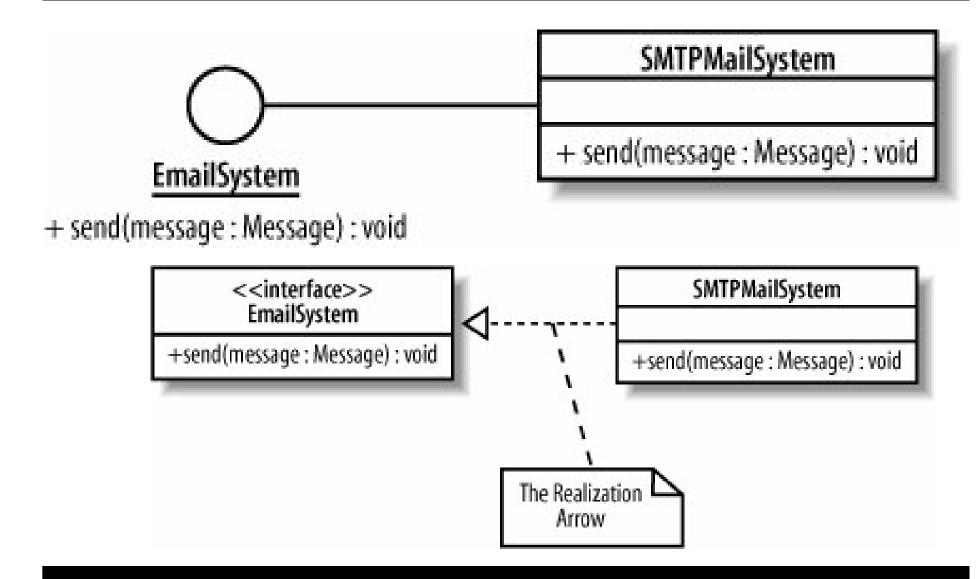
<<interface>>
EmailSystem
+ send(message : Message) : void

Stereotype Notation

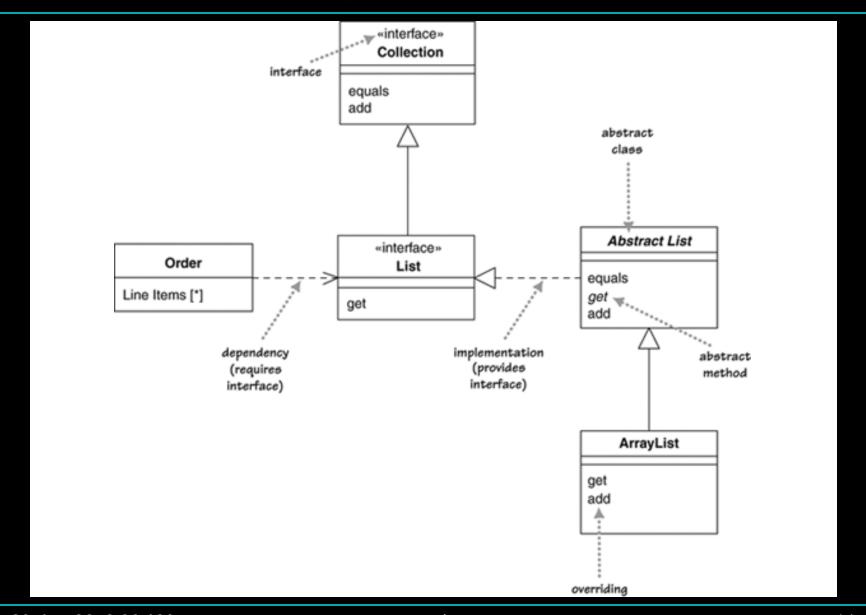
Cr
EmailSystem
+ send(message : Message) : void

"Ball" Notation

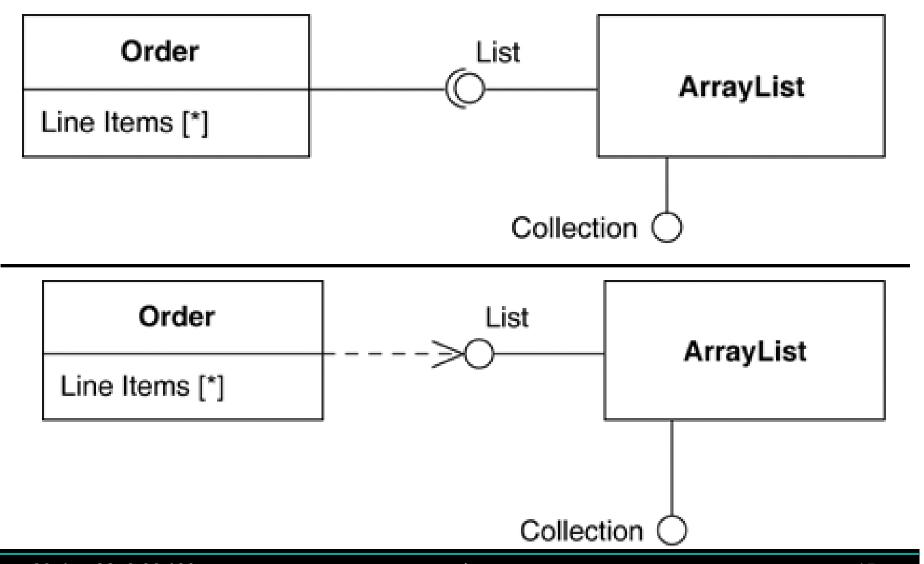
Class implementing Interface



Interface & Abstract classes



Interface



Constraints

Senior Citizen Group

0..1

 $\{person.age > 60\}$

0..*

Person

The End

"Out of intense complexities, intense simplicities emerge."

—Winston Churchill