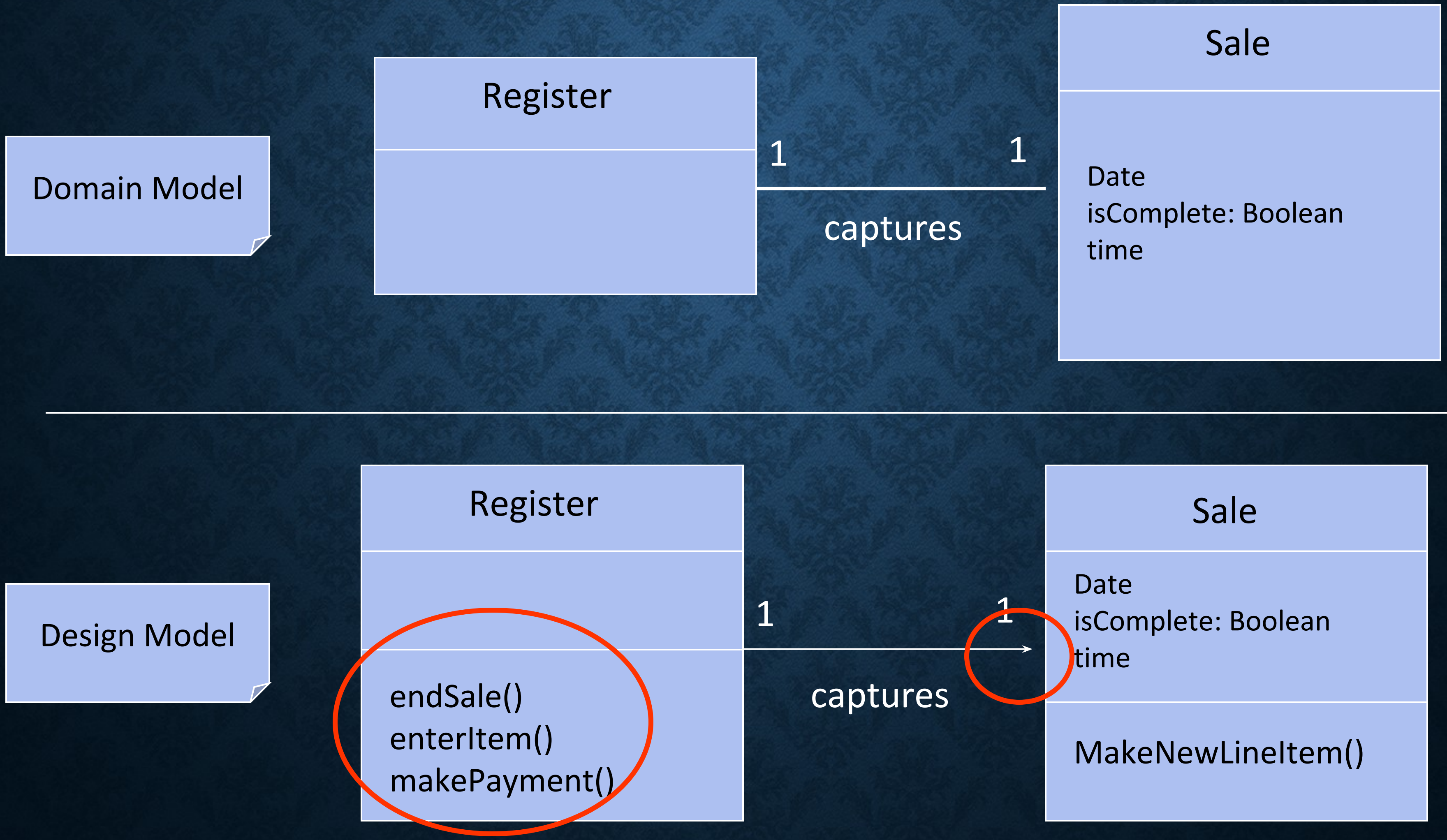
DESIGN CLASS DIAGRAM (DCD)

DESIGN CLASS DIAGRAM (DCD)

- A **Design class diagram (DCD)** illustrates the specifications for software classes and interfaces (for example Java interfaces) in an application.
- It involves
  - Classes, associations and attributes
  - Interfaces with their operations
  - Methods
  - Navigability
  - Dependencies
- Design classes can be identified from interaction diagrams or from Domain Model

DOMAIN MODEL VS DESIGN MODEL CLASSES



Domain Model

Register

1

1

captures

Sale

Date
isComplete: Boolean
time

Design Model

Register

endSale()
enterItem()
makePayment()

1

1

captures

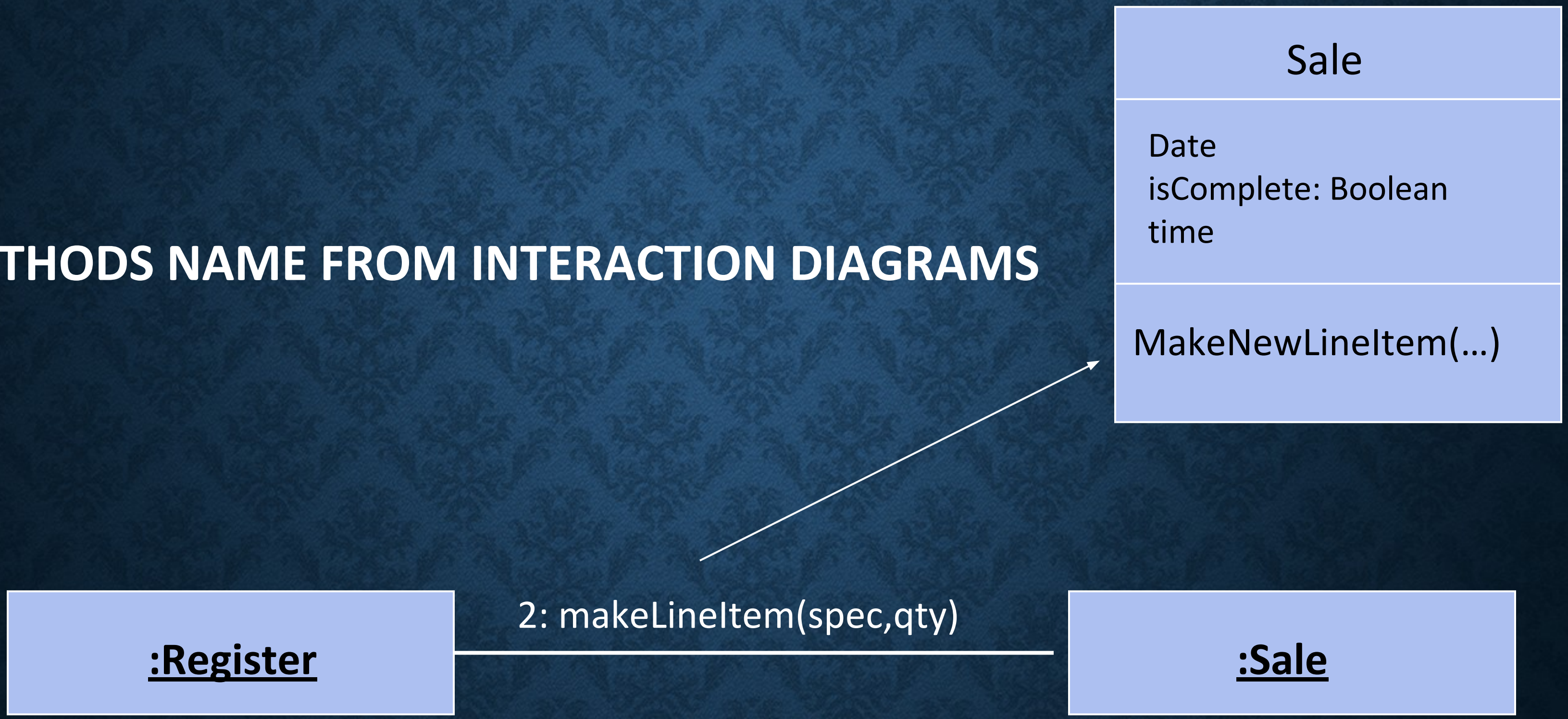
Sale

Date
isComplete: Boolean
time

MakeNewLineItem()

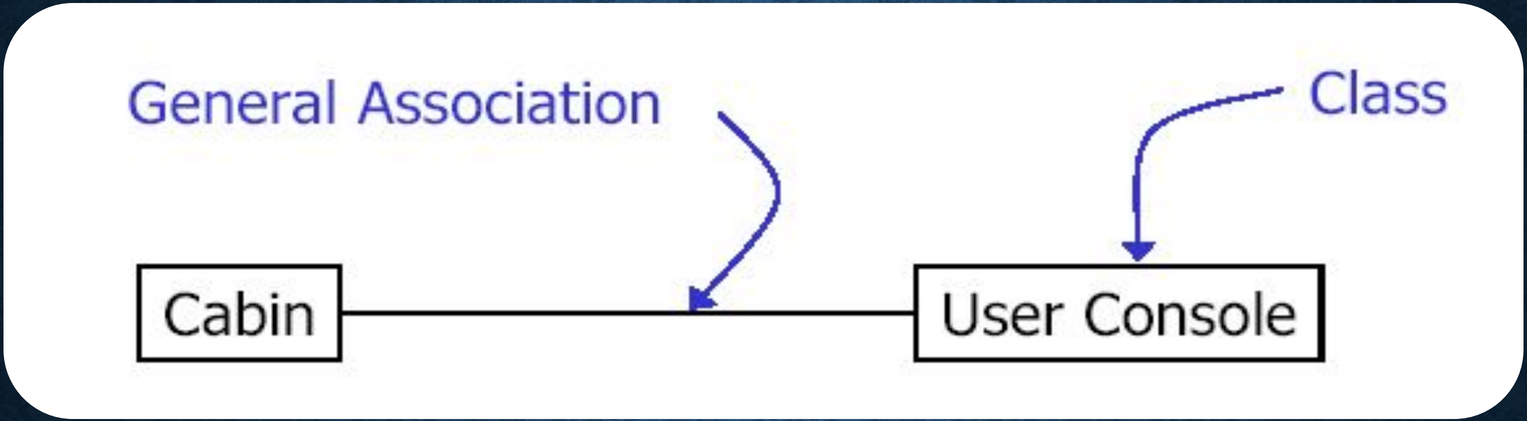
Methods

METHODS NAME FROM INTERACTION DIAGRAMS

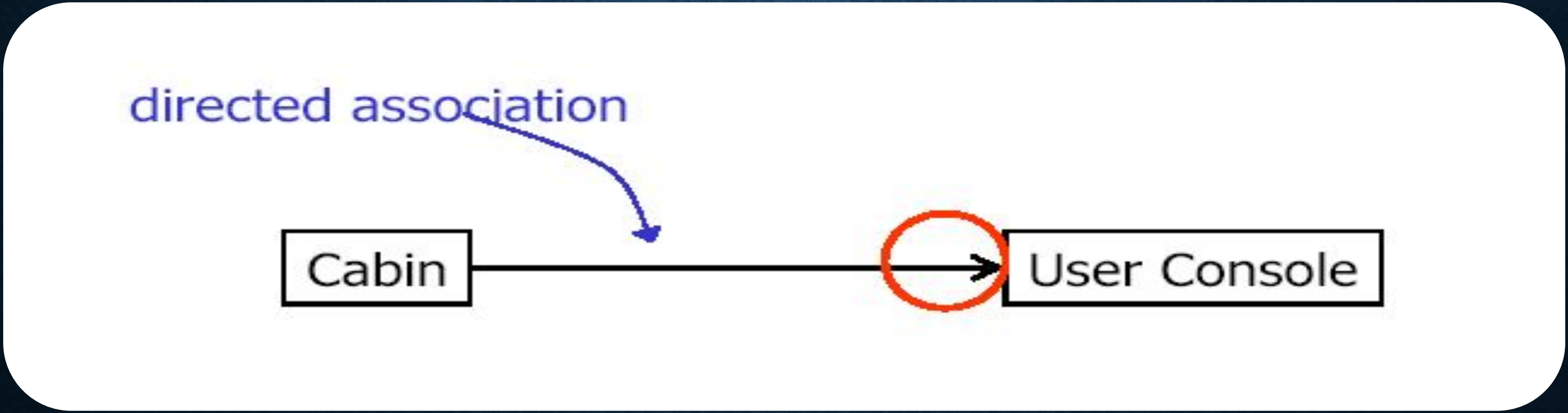


ASSOCIATION AND NAVIGABILITY OF CLASSES

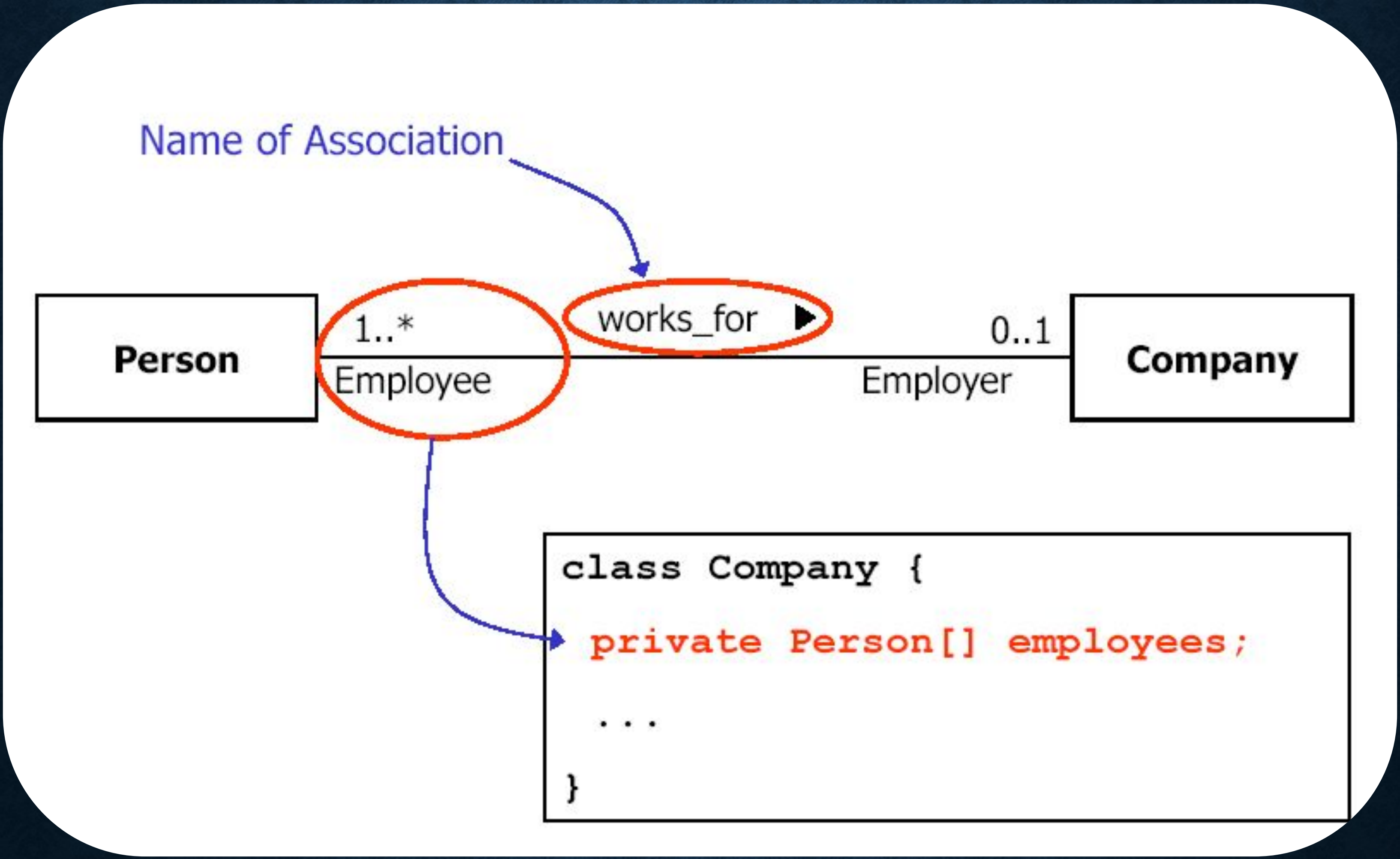
- Associations describe navigation paths through the class/object model



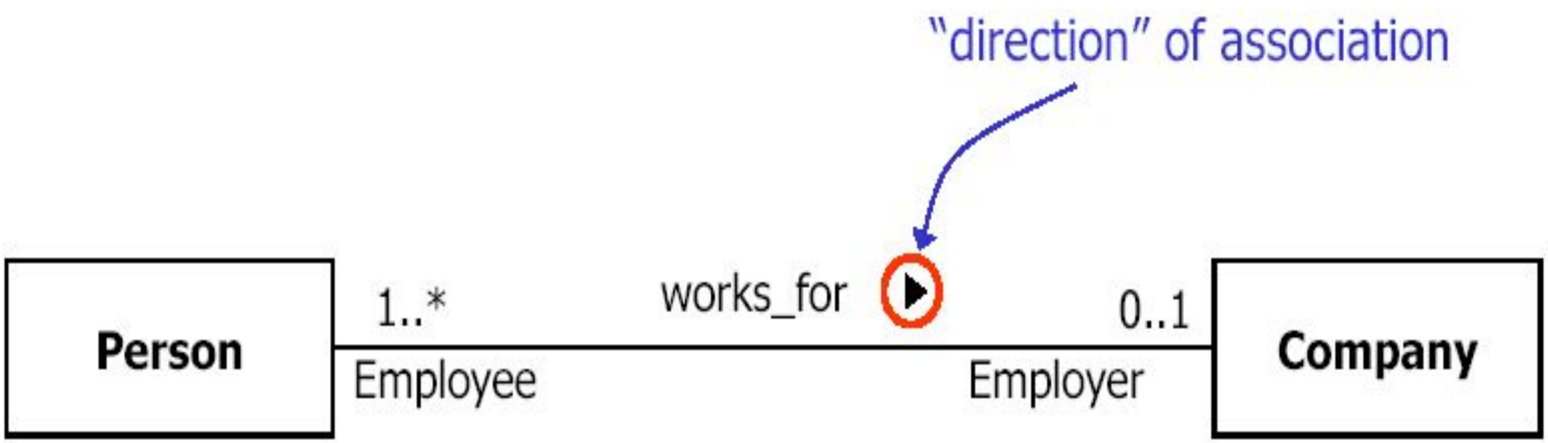
Here: navigation in both directions possible

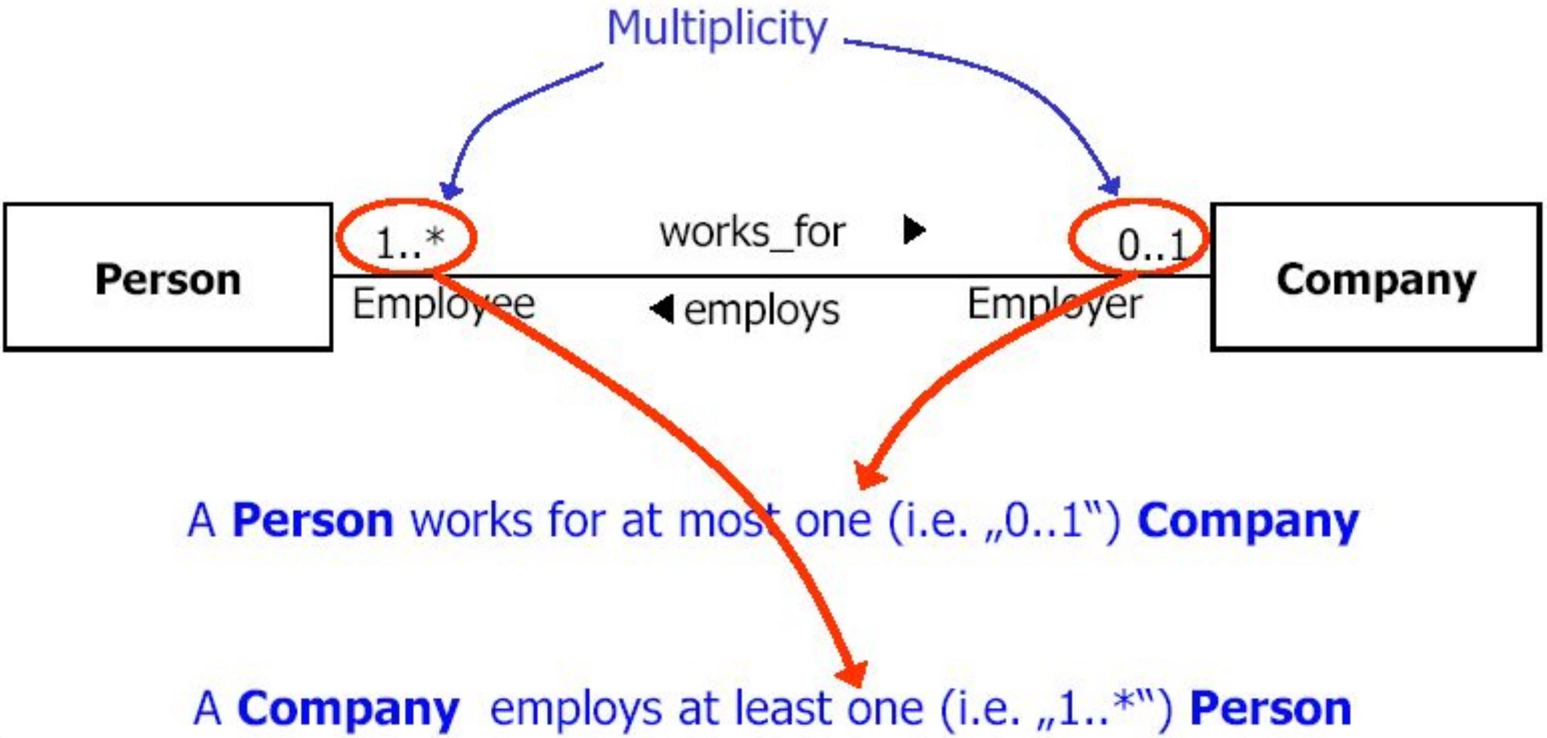


“Cabin” instances can access “User Console” instances, but not the other way around!



```
class Company {
 private Person[] employees;
  ...
}
```


"direction" of association
Person
1..*
works_for ▶
0..1
Company
Employee
Employer

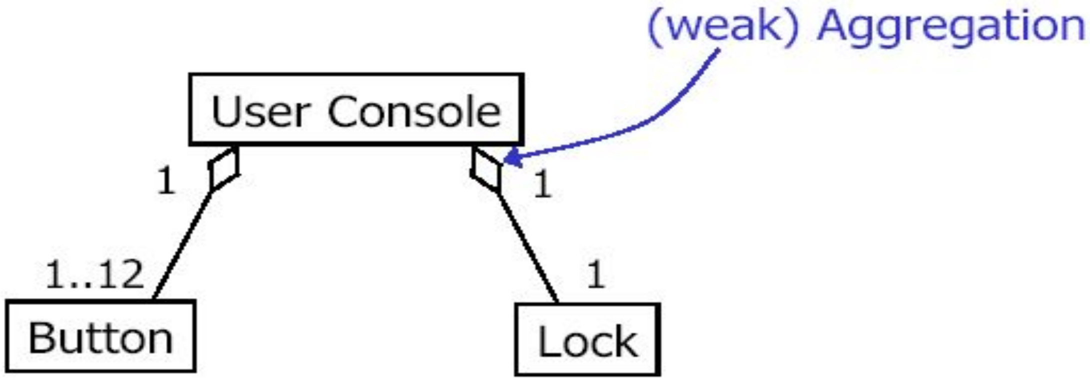


A **Person** works for at most one (i.e. „0..1") **Company**

A **Company** employs at least one (i.e. „1..*") **Person**

TYPES OF ASSOCIATION: AGGREGATION

- Aggregation is a kind of association used to model **Whole-Part** relationship.

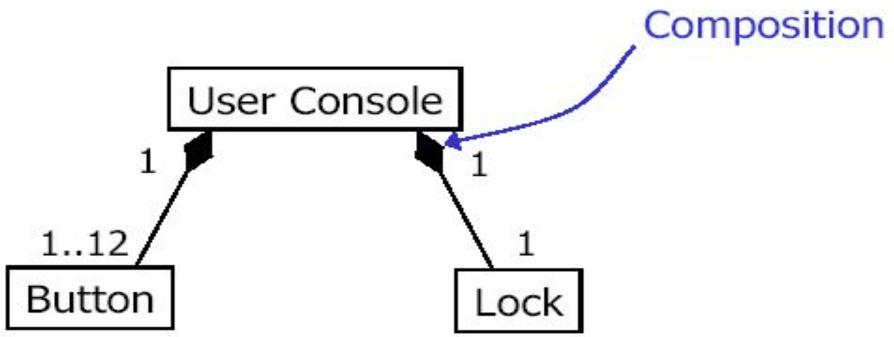


Every user console "has" 1-12 buttons and exactly one lock

Every lock/button belongs to exactly one user console

TYPES OF ASSOCIATION: COMPOSITION

- Composition means the part is a member of only one composite object (the whole), and it has an existence and disposition dependency on the composite

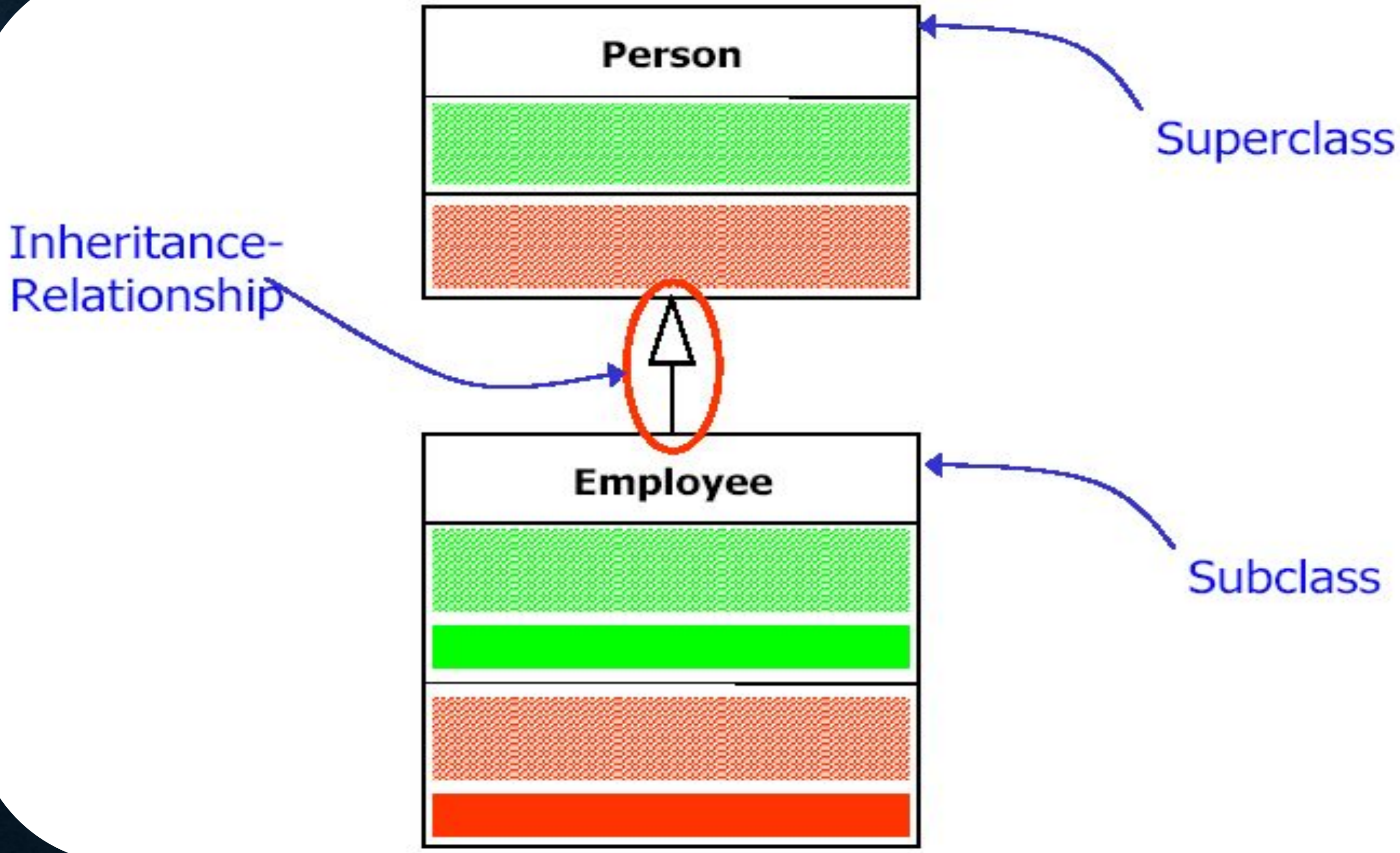


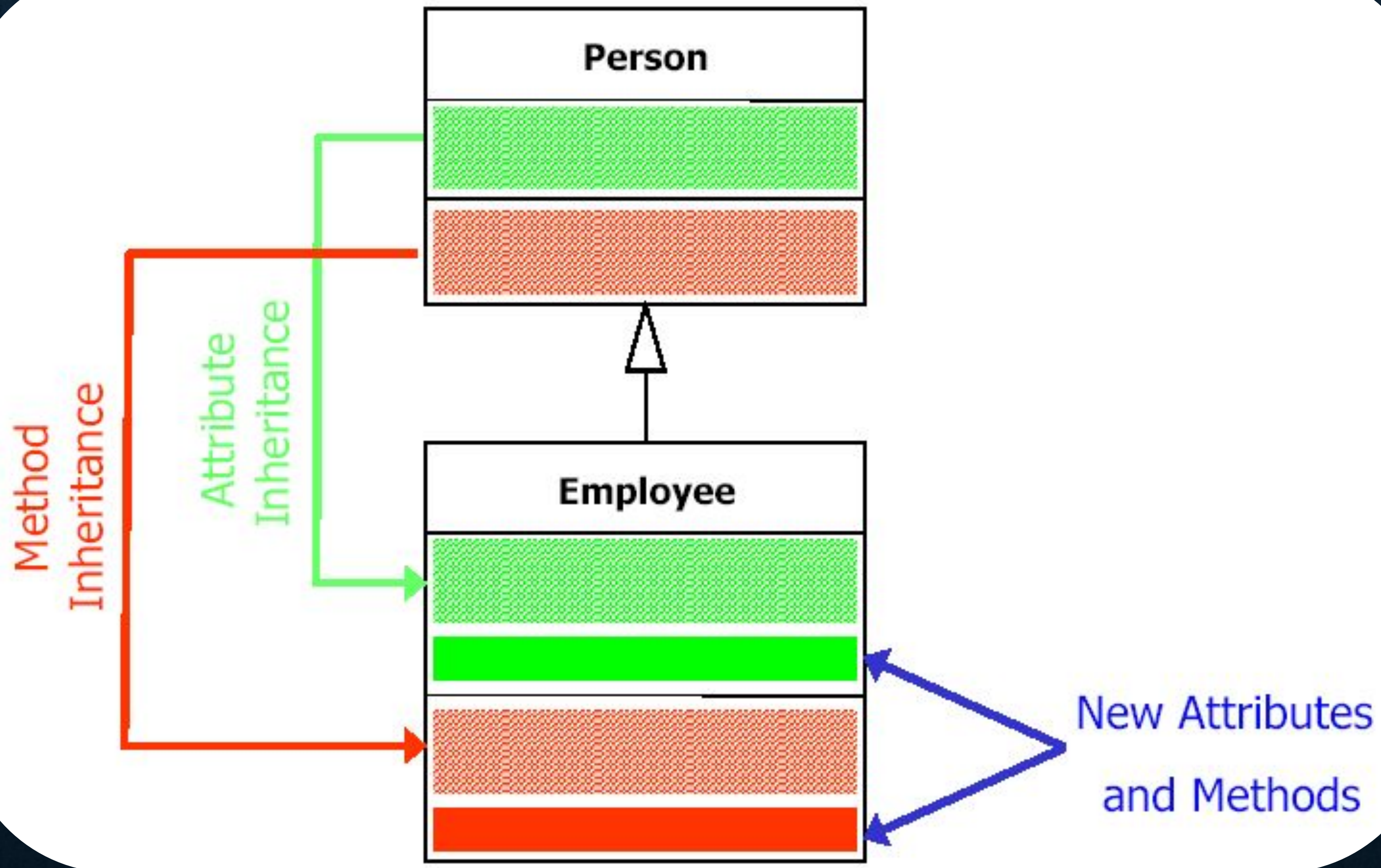
Every user console *contains* 1-12 buttons and exactly one lock

The user console is responsible for construction and destruction of locks and buttons; it controls the "life cycle" of its locks/buttons

Every button/lock belongs to exactly one user console

INHERITANCE





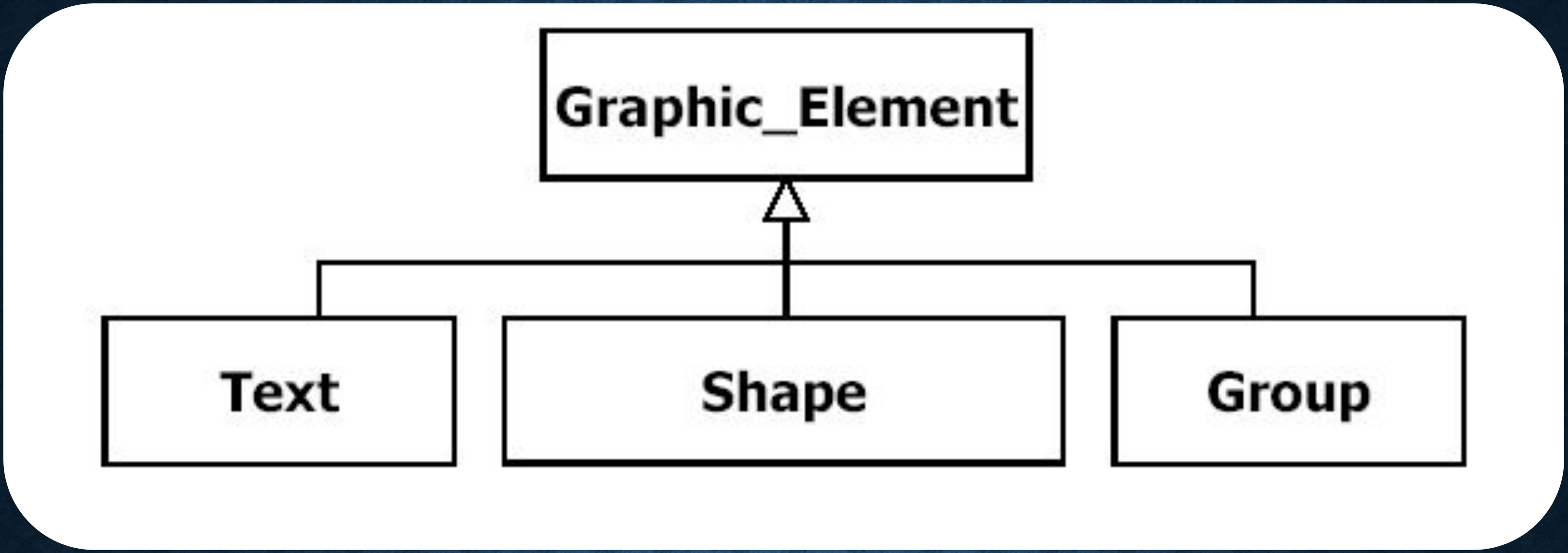
Person

Employee

Method Inheritance

Attribute Inheritance

New Attributes and Methods



- Subclasses inherit attributes, associations and operations from their Superclasses
- Subclasses can overwrite inherited operations
- Each instance of a subclass is also an instance of its superclass (Polymorphism)

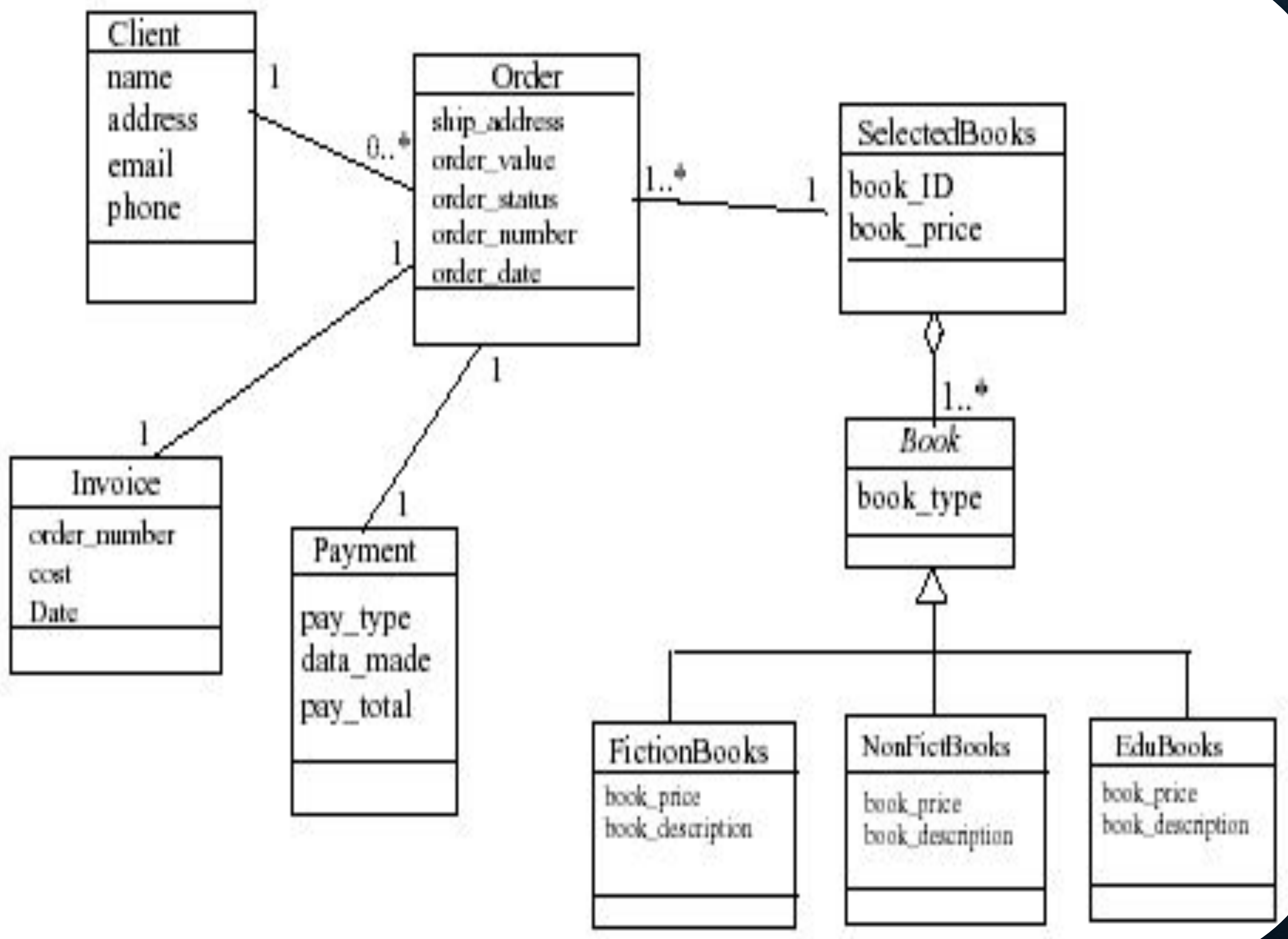
CASE STUDY: "BUYABLE BOOKS LTD"

Problem Statement

"Buyable Books ltd" would like to establish an internet-based online bookstore. The customer can interactively select any book from the categories fiction, non-fiction and educational. They can then examine a short description and price of each book, select the ones they want to purchase and proceed to paying for them. To complete payment a form must be filled out with shipping and payment information.

Payment may be by credit card or postal order. The customer may write to a help desk to ask for information. When the order is confirmed, a confirmation email is sent to the customer with details of the order and a reference number. The reference number can be used to check the status of the order on line. Behind the scenes the system must verify the customer's credentials, request the books from the warehouse, print an invoice and request a delivery to the customer.

Solution: Static Class Model



REFERENCES

- Applying UML and Patterns by Craig Larman
  - **Chapter 19**: 19.1, 19.4, 19.5, 19.6
  - **Chapter 26**: 26.1, 26.2, 26.4, 26.6, 26.7
  - **Chapter 27**: 27.1, 27.2 ,27.4 ,27.5, 27.6, 27.7 ,27.8 ,27.10