

# Object-Oriented Programming Design



1. Introduction to Object-Oriented Concepts
2. How to think in Terms of Objects
3. Advanced Object-Oriented Concepts
4. The anatomy of a Class
5. Class Design guidelines
6. Designing with objects
7. Mastering Inheritance and Composition
8. Frameworks and Reuse: Designing with interfaces and Abstract classes
9. Building objects

## **10. Creating Object Models with UML**

11. Objects and Portable Data: XML
12. Persistent Objects: Serialization and Relational Databases
13. Objects and the Internet
14. Objects and Client/Server Applications
15. Design Patterns

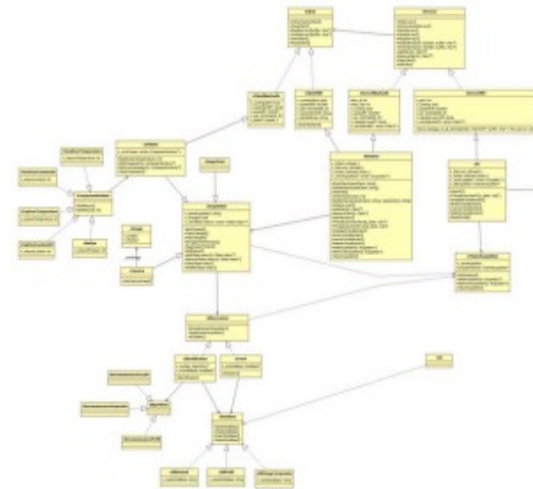




# 10. Creating Object Models with UML

## What is UML?

- ▶ UML, as its name implies, is a modeling language. The UML User Guide defines UML as “a graphical language for visualizing, specifying, constructing and documenting the artifacts of a software-intensive system”





# 10. Creating Object Models with UML

## What is UML?

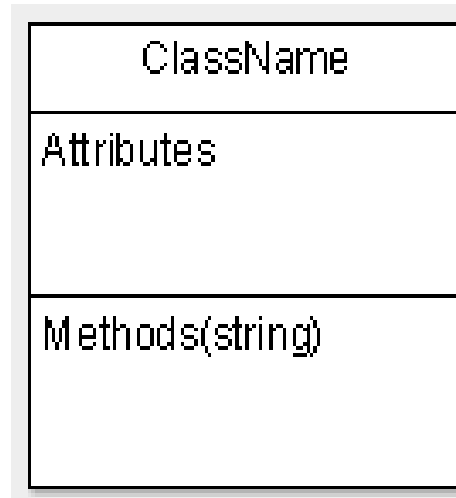
- ▶ UML gives you a standard way to write the system's blueprints.
- ▶ UML offers a way to graphically represent and manipulate an object-oriented (OO) software system. It is not only the representation of a the design of a system, but a tool to assist in this design.





## 10. Creating Object Models with UML

The structure of a Class Diagram:





## 10. Creating Object Models with UML

### Access Designations:

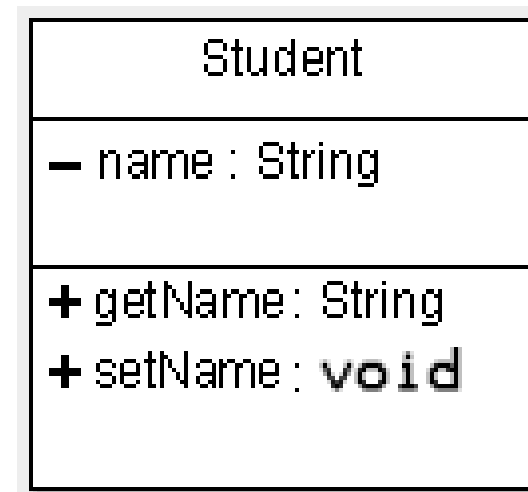
(+) public

(-) private

### Type:

Type of an Attribute

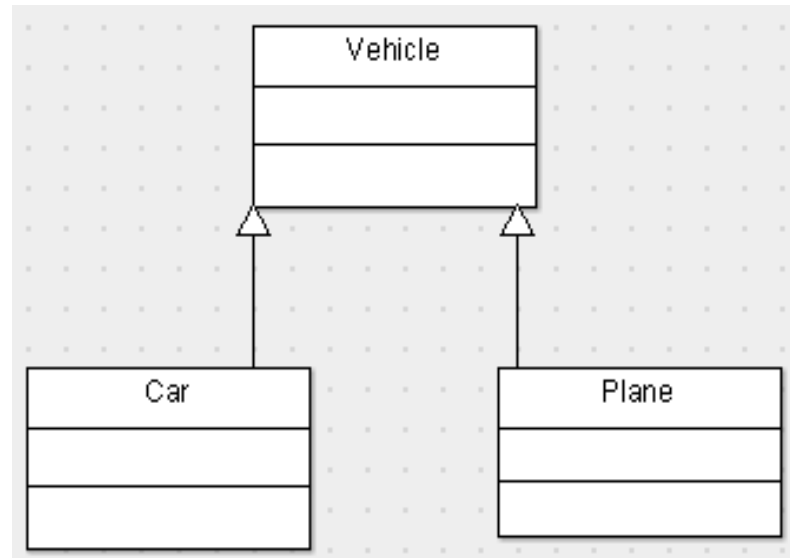
Return type of an method





# 10. Creating Object Models with UML

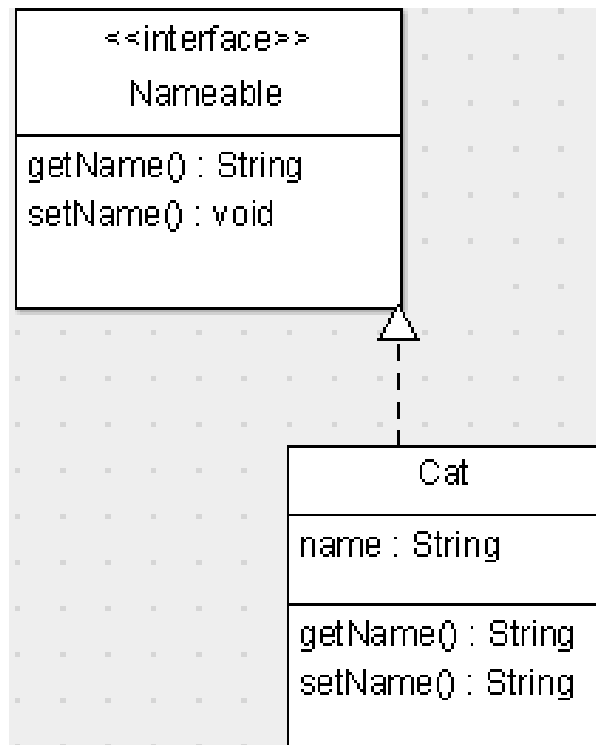
**Inheritance:**





# 10. Creating Object Models with UML

## Interfaces:



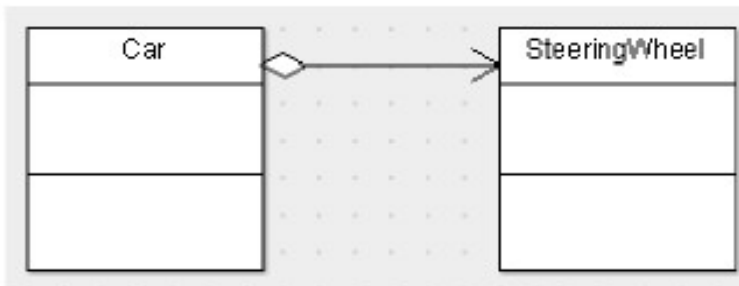
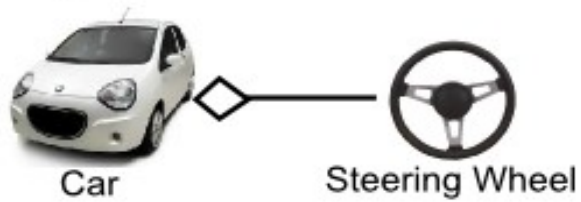




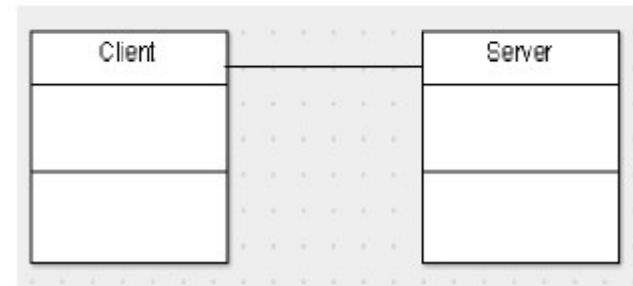
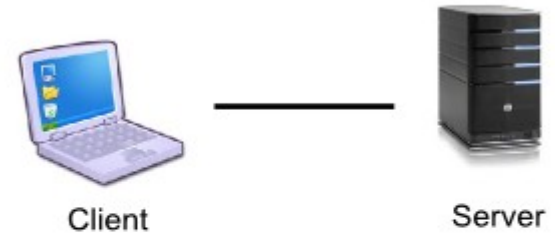
# 10. Creating Object Models with UML

## Composition:

### Aggregation



### Association





# 10. Creating Object Models with UML

## Cardinality:

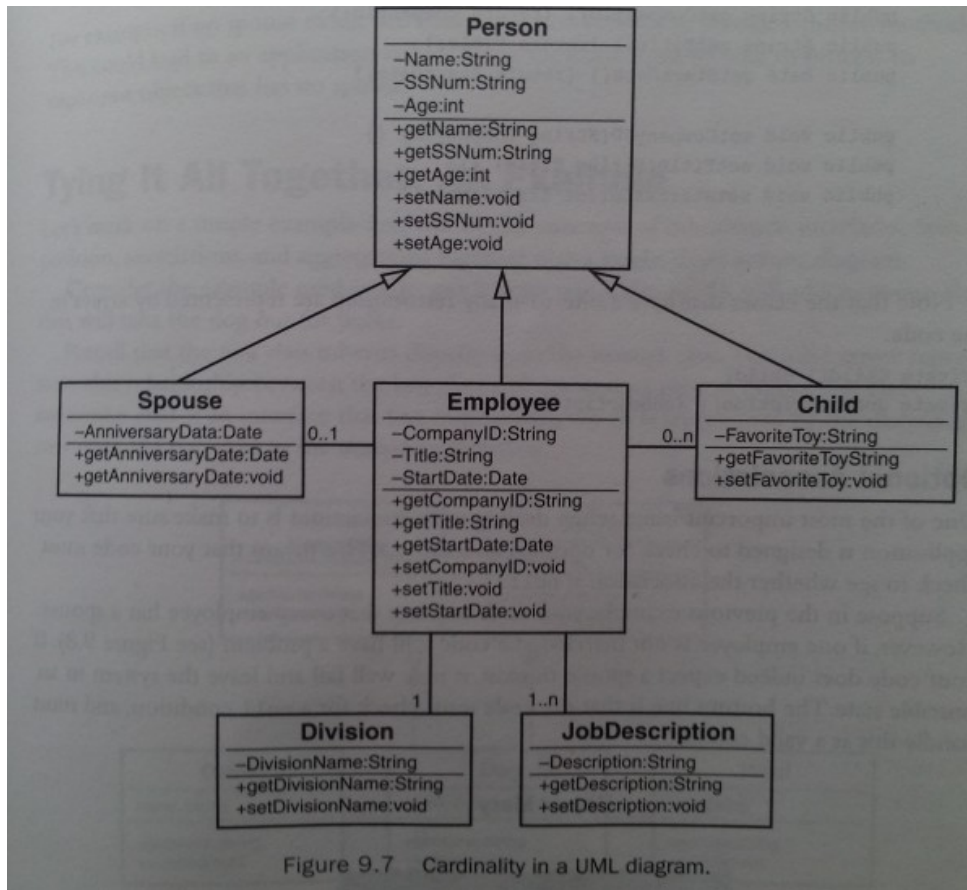


Figure 9.7 Cardinality in a UML diagram.

Table 9.1 Cardinality of Class Associations

Optional/Association	Cardinality	Mandatory
Employee/Division	1	Mandatory
Employee/JobDescription	1...n	Mandatory
Employee/Spouse	0...1	Optional
Employee/Child	0...n	Optional



## 10. Creating Object Models with UML

### An E-Business Example

- ▶ In this example, we'll use inheritance (via interfaces and abstract classes) and composition. Our goal is to create a framework that will make code reuse a reality, reduce coding time, and reduce maintenance.



## 10. Creating Object Models with UML

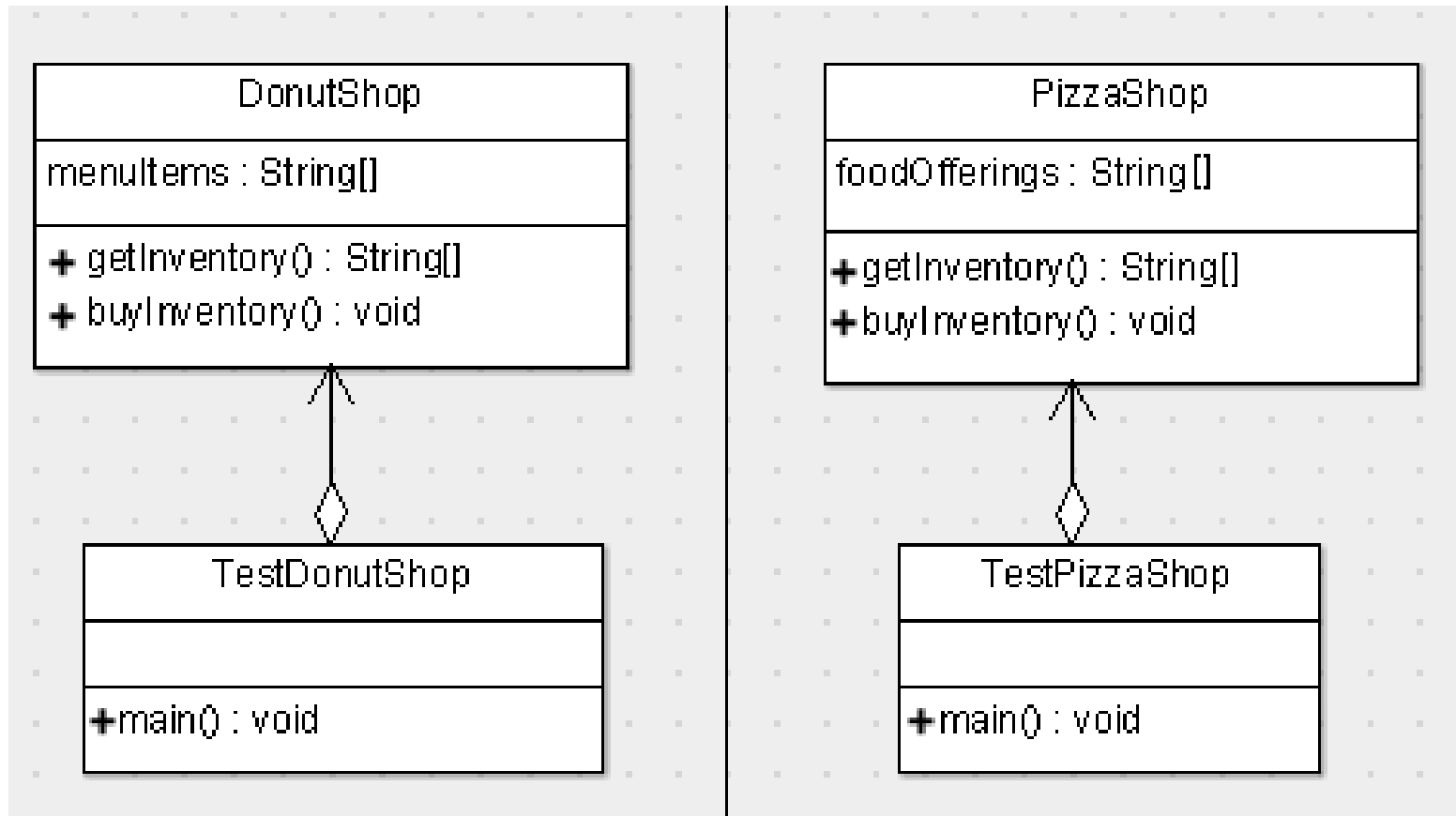
### An E-Business Example

- ▶ Let's start our own Internet business. Let's assume that we have a client, a small pizza shop. The owner would like that customers can order pizza online. At the site we develop, customers will be able to access the website, select the products they want to order.
  
- ▶ Then a donut shop wants to have the same system.



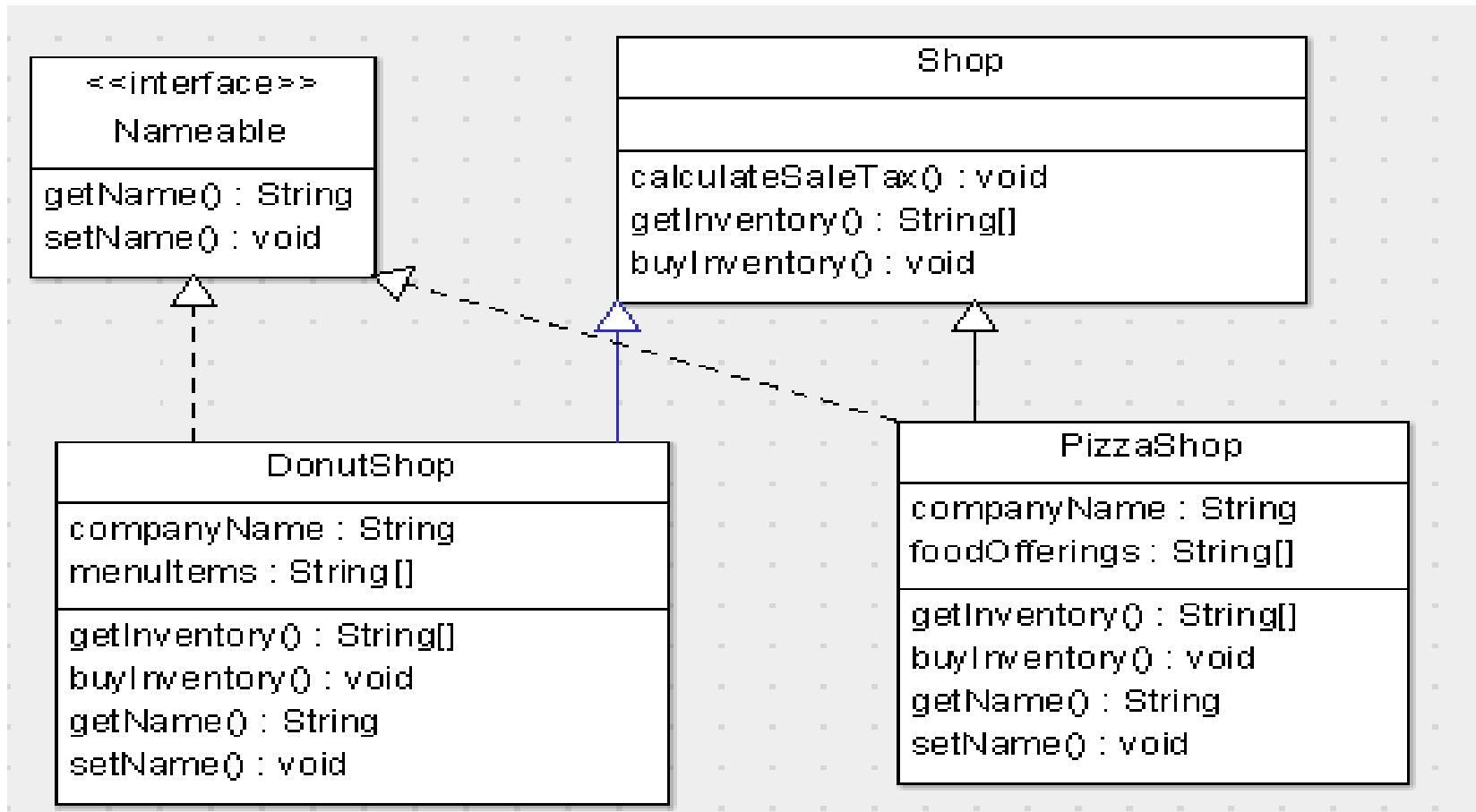
# 10. Creating Object Models with UML

## The Non-Reuse approach



# 10. Creating Object Models with UML

## Tan E-Business Solution



1. Introduction to Object-Oriented Concepts
2. How to think in Terms of Objects
3. Advanced Object-Oriented Concepts
4. The anatomy of a Class
5. Class Design guidelines
6. Designing with objects
7. Mastering Inheritance and Composition
8. Frameworks and Reuse: Designing with interfaces and Abstract classes
9. Building objects
10. Creating Object Models with UML

## **11. Objects and Portable Data: XML**

12. Persistent Objects: Serialization and Relational Databases
13. Objects and the Internet
14. Objects and Client/Server Applications
15. Design Patterns

