

Introduction to UML

UNIFIED
MODELING
LANGUAGE



- ◆ **Visual modeling**
- ◆ **Models and its importance**
- ◆ **History of the UML**
- ◆ **Basic UML concepts**

What is Visual Modeling?

- ◆ **Visual modeling** is a way of thinking about problems using models organized around real-world ideas.

What is a Model?

- ◆ **A Model** is an abstraction (or representation) of a complex problem or structure by filtering out nonessential details.
- ◆ **Models help** us
 - organize,
 - visualize,
 - understand, and
 - create complex things.

Models are useful for ...

- ◆ Understanding the problems
- ◆ Communicating with those involved in the project (customer, domain expert, analyst, designers, etc.)
- ◆ Modeling enterprises
- ◆ Preparing documentation
- ◆ Designing programs and databases

Tools for describing models

- ◆ Textual description
- ◆ Data
- ◆ Formulas
- ◆ **Diagrams**

Good Models

- ◆ Have the right amount of details & structure
- ◆ Represent what's important in the system
- ◆ Model complex systems
- ◆ Include different stakeholders' perspectives
- ◆ Represent functional & non-functional requirements
- ◆ Do not include any premature decision

Unified Modeling Language (UML 1997)



- ◆ **What?** UML is a standard graphical language for visualizing, specifying, constructing, and documenting the artifacts of an object-oriented system under development.
- ◆ **How?** By using icons, 2D symbols, Paths, and strings:
 - Every graphical shape has a certain meaning
 - A model could consists of many diagrams.
- ◆ **Why?** Because it's commonly used in industry and it's becoming a standard.

UML Diagrams

- ◆ Use case diagram
- ◆ Sequence diagram
- ◆ Collaboration diagram
- ◆ Statechart diagram
- ◆ Class diagram
- ◆ Object diagram
- ◆ Component diagram
- ◆ Deployment diagram
- ◆ Activity diagram

Benefits of Diagrams

Diagrams are good for

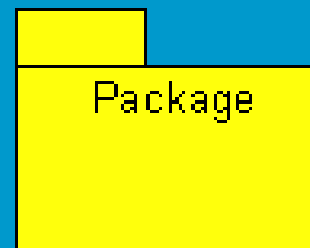
- ◆ Communicating ideas
- ◆ Generating new ideas
- ◆ Testing ideas & making predications
- ◆ Understanding structures & relationships

General Rules of Diagrams

- ◆ **Simplicity of representation:** show *only* what needs to be shown
- ◆ **Internal consistency** (and within a set of diagrams).
- ◆ **Completeness:** show *all* what needs to be shown
- ◆ **Hierarchical representation:** break the system down into layers & show more details at the lower levels.

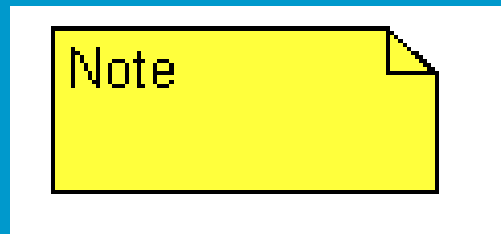
UML Concepts: Packages

- ◆ **A package** is a general-purpose mechanism for organizing elements into groups.
- ◆ Packages could also be used to present different views of system's architecture.
- ◆ Well-structured packages are loosely coupled and very cohesive.
- ◆ In the UML, a package is represented as a tapped folder.

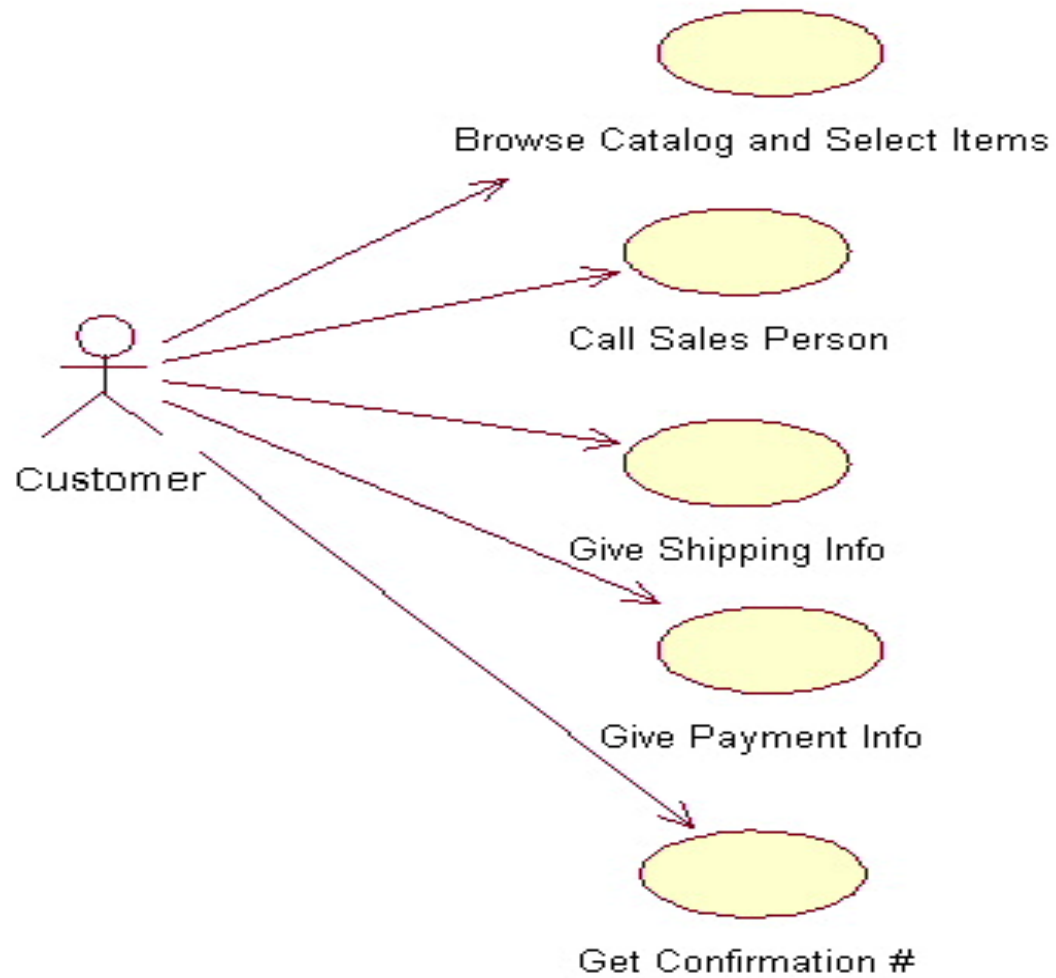


UML Concepts: Notes

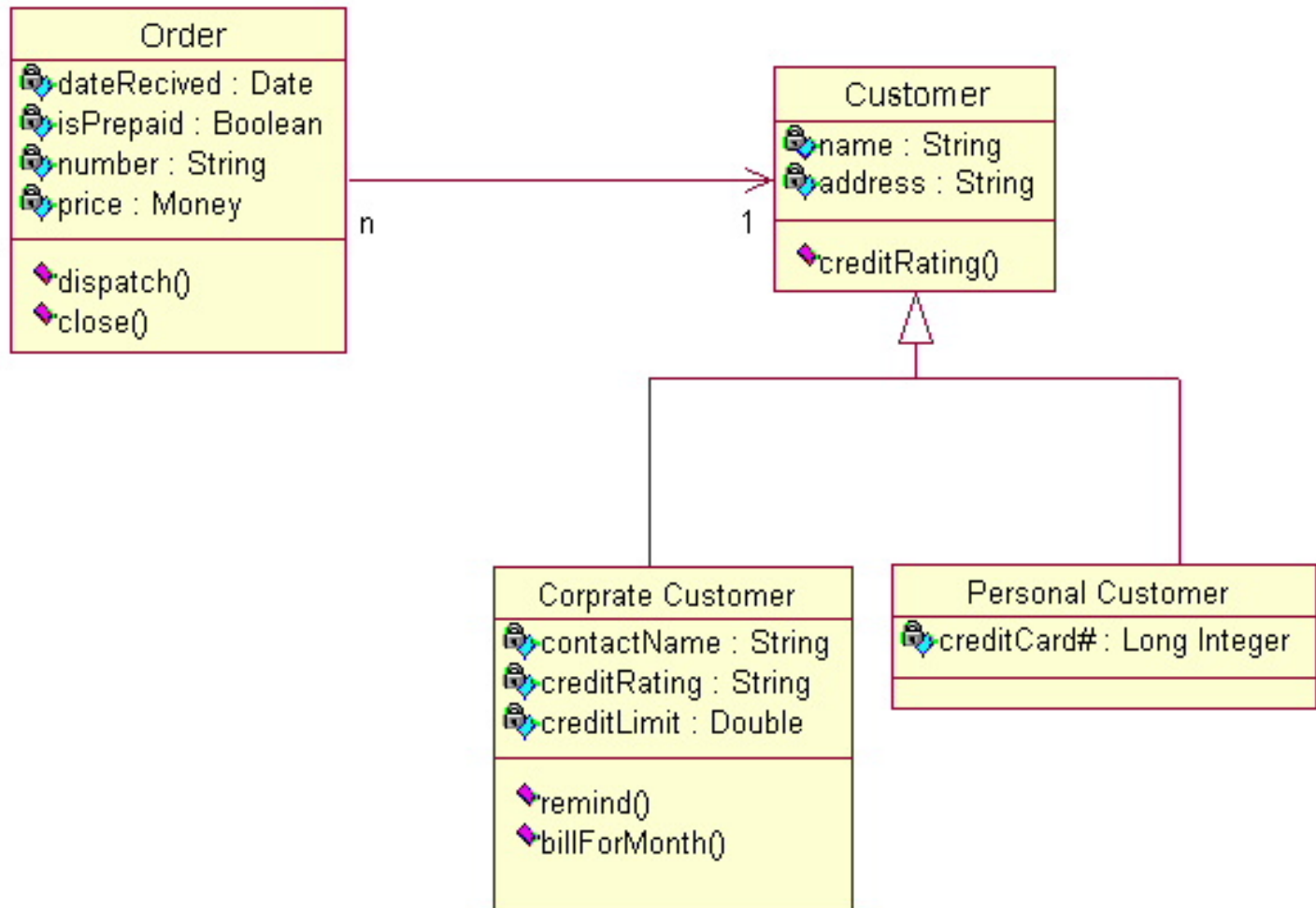
- ◆ **A note** is a graphical symbol for rendering constraints or comments attached to an element or a collection of elements.
- ◆ Notes have no semantic impact.
- ◆ Notes may be attached to one or more element using dependencies.
- ◆ In the UML, a note is represented as a rectangle with a dog-eared corner.



Use Case diagram



Class Diagrams



UML Resources

◆ [Click here](#)