

# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ II

Κωδικός Θ: ΤΠΕ00Λ, Κωδικός Ε: ΤΠΕ10Λ  
Ώρες (Θ - ΑΠ - Ε): 2 - 1 - 2  
Προαπαιτούμενα: Αντικειμενοστραφής  
Προγραμματισμός





# Software Modeling

## UML

**ΘΕΩΡΙΑ & ΕΡΓΑΣΤΗΡΙΟ 2**

# Σημερινό μάθημα

- Software Modeling Languages
- UML
  - Use Case Diagrams
  - Class Diagrams
  - Sequence Diagrams
- UML tools
  - Visual Paradigm
- Ασκήσεις

# Modelling Languages Definition

## Modelling Language

- A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure.

## A modeling language can be graphical or textual.

- Graphical modeling languages use a diagram technique with named symbols that represent concepts and lines that connect the symbols and represent relationships and various other graphical annotation to represent constraints.
- Textual modeling languages typically use standardized keywords accompanied by parameters to make computer-interpretable expressions.

# Modeling Languages: Graphical

## Behavior Trees

are a formal, graphical modelling language used primarily in systems and software engineering.

### Set of Natural Language Requirements

**Requirements - Train Station System**

Develop a system to model the behavior of a Train-Station. You need to model a train entering the station from the north and then leaving the station to the south. A crossing with boom gates and flashing red lights is located just south of the station. There is a signal to the north of the station that only allows a train to enter when the station is not occupied, that is, when the north signal is green. There is also an exit signal that ensures the train can only leave the station when the boom gates are down. There is also a north detector that can detect the train approaching the station region from the north. And, there is an exit detector that detects when a train leaves to the south. The following requirements capture the desired behavior.

**R1.** Initially the station is not occupied. The north signal turns green whenever the station is not occupied. Whenever the north signal is green a train may approach from the north. When approaching from the north a train is detected, by the north detector, which causes the north signal to turn red.

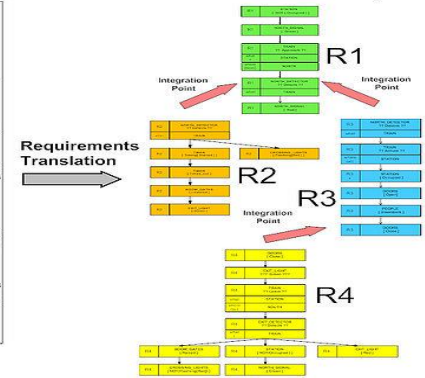
**R2.** When the north detector detects a train it causes the crossing lights to start flashing red. At the same time, a timer starts timing and when it times out it causes the boom gates to be lowered after which the exit light turns green.

**R3.** After the train is detected the north detector, it subsequently arrives at the station, the doors open, the people disembark, and then the doors close.

**R4.** After the doors close the train may leave the station only when and if the exit light is green. When the train leaves the station, heading south, it is detected by the exit detector which means the station is again not occupied. This causes the north signal to turn green and the exit light to turn red. When the exit detector detects the train leaving, it also causes the boom gates to be raised and then the crossing lights to stop flashing red.

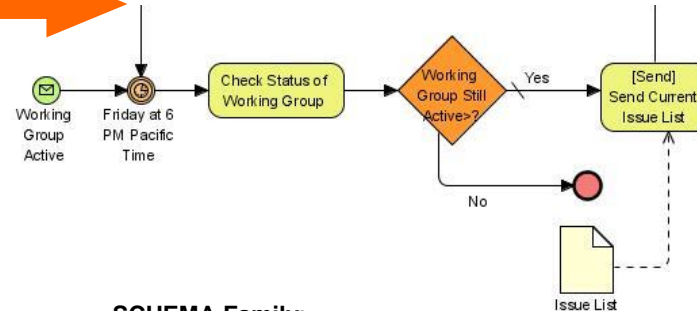
For the purposes of the exercise ignore trains approaching the station from the south. This additional requirement can be integrated later as a separate exercise. Also ignore situations where the train does not stop at the station - this too requires some refinements to the requirements.

### Set of Requirements Behavior Trees (RBTs)



## Business Process Modeling Notation

(BPMN, and the XML form BPML) is an example of a Process Modeling language.



## EXPRESS and EXPRESS-G

(ISO 10303-11) is an international standard general-purpose data modeling language. Extended Enterprise Modeling Language (EEML) is commonly used for business process modeling across a number of layers.

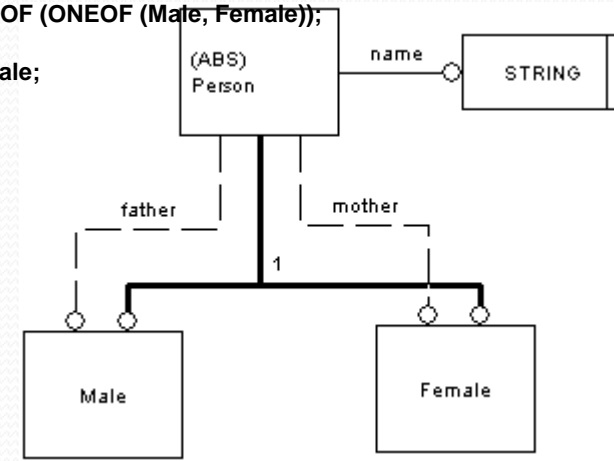
### SCHEMA Family; ENTITY Person

ABSTRACT SUPERTYPE OF (ONEOF (Male, Female));  
 name: STRING;  
 mother: OPTIONAL Female;  
 father: OPTIONAL Male;  
 END\_ENTITY;

ENTITY Female  
 SUBTYPE OF (Person);  
 END\_ENTITY;

ENTITY Male  
 SUBTYPE OF (Person);  
 END\_ENTITY;

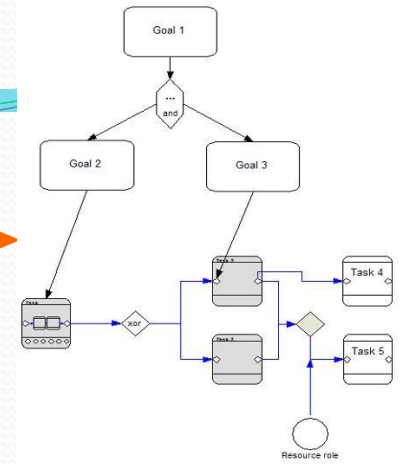
END\_SCHEMA;



# Modeling Languages: Graphical

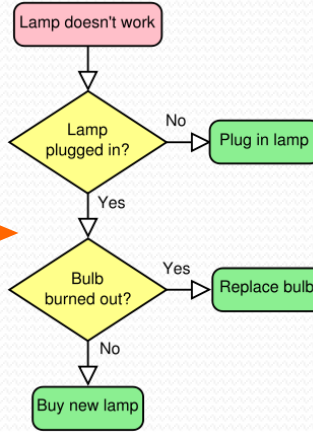
## Extended Enterprise Modeling Language (EEML)

is commonly used for business process modeling across a number of layers.



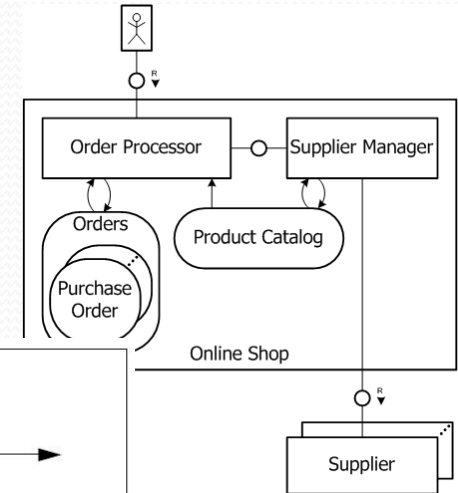
## Flowchart

is a schematic representation of an algorithm or a stepwise process,



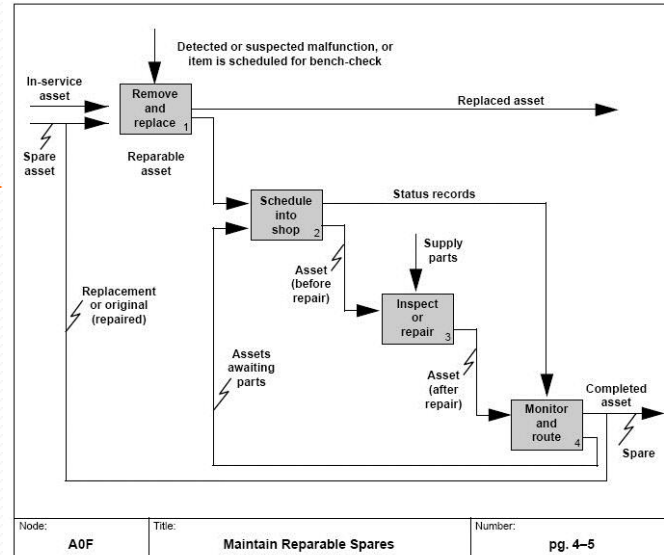
## Fundamental Modeling Concepts (FMC)

modeling language for software-intensive systems.



## IDEF

is a family of modeling languages, which include IDEF0 for functional modeling, IDEF1X for information modeling, IDEF3 for business process modelling, IDEF4 for Object-Oriented Design and IDEF5 for modeling ontologies.



# Modeling Languages: Graphical

## Jackson Structured Programming

(JSP) is a method for structured programming based on correspondences between data stream structure and program structure

## LePUS3

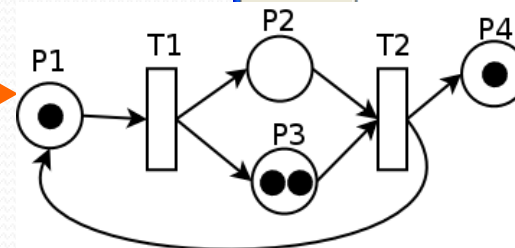
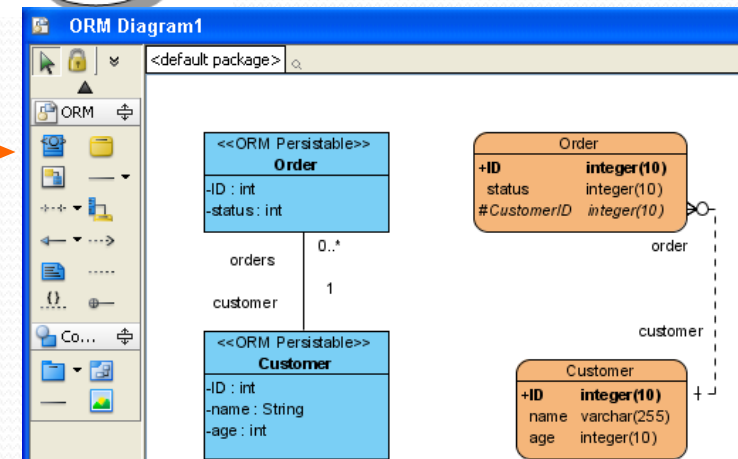
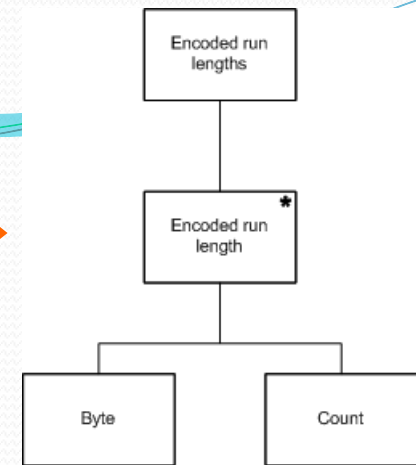
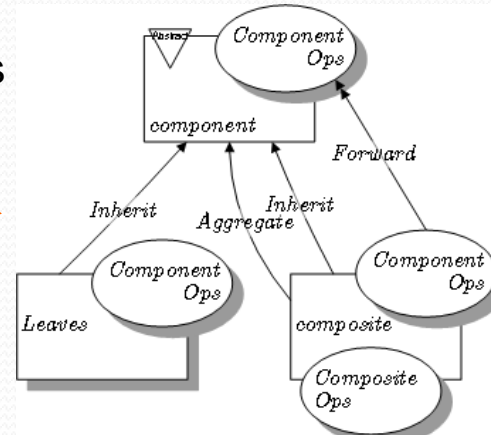
is an object-oriented visual Design Description Language and a formal specification language that is suitable primarily for modelling large object-oriented (Java, C++, C#) programs and design patterns.

## Object Role Modeling

(ORM) in the field of software engineering is a method for conceptual modeling, and can be used as a tool for information and rules analysis.

## Petri nets

The simplicity of its basic user interface easily enabled extensive tool support over the years, particularly in the areas of model checking, graphically-oriented simulation, and software verification.



# Modeling Languages: **UML (Unified Modeling Language)**

## Γενικά

- Είναι μιας γενικής χρήσης γλώσσα μοντελοποίησης η οποία έχει καταξιωθεί ως industry standard για συστήματα software-intensive.
- Η τελευταία έκδοση της UML, η έκδοση 2, υποστηρίζει 13 διαγράμματα.

## Η UML είναι μια γλώσσα για:

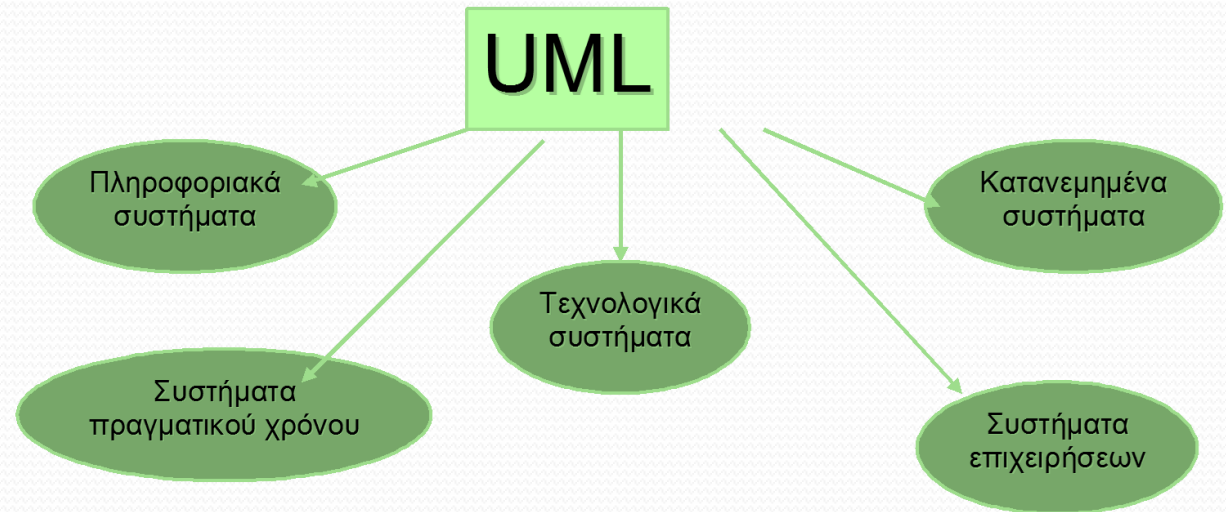
- Κατάρτιση προδιαγραφών λογισμικού και τεκμηρίωση τμημάτων λογισμικού
- Αναπαράσταση με οπτικό τρόπο (visualization) τμημάτων λογισμικού
- Μοντελοποίηση εταιρικών και άλλων συστημάτων που δεν αφορούν λογισμικό

## Στόχοι της UML

- Η μοντελοποίηση συστημάτων με βάση τις αρχές των αντικειμενοστραφών μοντέλων
- Η δημιουργία μιας μοντελοποιημένης γλώσσας που μπορεί να χρησιμοποιηθεί τόσο από τον άνθρωπο όσο κι από τις μηχανές

# Modeling Languages: **UML** (Unified Modeling Language)

## Η χρήση της UML



## Τμήματα της UML

- Όψεις: Δείχνουν διαφορετικά χαρακτηριστικά του συστήματος που μοντελοποιούνται. Μια όψη αποτελείται από ένα σύνολο διαγραμμάτων
- Διαγράμματα: Περιγράφουν τα περιεχόμενα μιας όψης. Υπάρχουν εννέα διαφορετικά διαγράμματα που χρησιμοποιούνται σε συνδυασμό
- Στοιχεία μοντέλου: Είναι οι έννοιες που χρησιμοποιούνται στα διαγράμματα για να αναπαραστήσουν τις κλάσεις, τα αντικείμενα και τις μεταξύ τους συσχετίσεις

# Modeling Languages: **UML** (Unified Modeling Language)

4 + 1 Όψεις της UML

## Logical View

(Design View)

*Structure*

*Behavior*

## Component View

(Implementation View)

*System Assembly*

*Configuration management*

## Use Case View

*Functionalities*

*System services*

*Actors*

## Deployment View

*System topology*

*Distribution*

*Delivery*

*Installation*

## Concurrency View

(Process View)

*Threads*

*Processes*

*Performance*

*Scaling*

# Modeling Languages: UML (Unified Modeling Language)

## Logical View

- **Επιδεικνύει** πως σχεδιάζεται η λειτουργικότητα του συστήματος.
- **Χρησιμοποιεί** διαγράμματα *class* και *object* για να αναπαραστήσει την στατική δομή του συστήματος.
- **Χρησιμοποιεί** διαγράμματα *state*, *sequence*, *collaboration*, και *activity* για να αναπαραστήσει την δυναμική συμπεριφορά του συστήματος.
- **Χρησιμοποιείται** από designers and developers.

## Use Case View

- **Επιδεικνύει** την λειτουργικότητα του συστήματος όπως αυτή εννοείται από εξωτερικούς actors.
- Οι Actors μπορεί να είναι χρήστες ή άλλα συστήματα.
- **Χρησιμοποιεί / Περιγράφεται** από διαγράμματα *use case* και *activity*.
- Είναι η κεντρική προσέγγιση / σκοπιά (view) η οποία κατευθύνει την υλοποίηση των υπόλοιπων.
- **Χρησιμοποιείται** από customers, designers, developers, testers.

## Deployment View.

- **Επιδεικνύει** την ανάπτυξη (deployment) ενός συστήματος μέσα στην αρχιτεκτονική σχεδίαση του με computers και devices.
- **Περιγράφεται** από διαγράμματα *deployment*.
- **Χρησιμοποιείται** από developers, system integrators, και testers.

## Concurrency View

- **Αναφέρεται** σε θέματα επικοινωνίας και συγχρονισμού συστημάτων concurrent.
- **Χρησιμοποιεί / Περιγράφεται** από διαγράμματα *state*, *sequence*, *collaboration*, *activity*, *deployment*, και *component*.
- **Χρησιμοποιείται** από developers και system integrators.

## Component View

- **Επιδεικνύει** την οργάνωση των κομματιών (component) κώδικα και την αλληλοεξάρτησή τους.
- **Χρησιμοποιεί / Περιγράφεται** από διαγράμματα *component*.
- **Χρησιμοποιείται** από developers.

# Modeling Languages: UML (Unified Modeling Language)

## Τα διαγράμματα της UML

### Structure Diagrams:

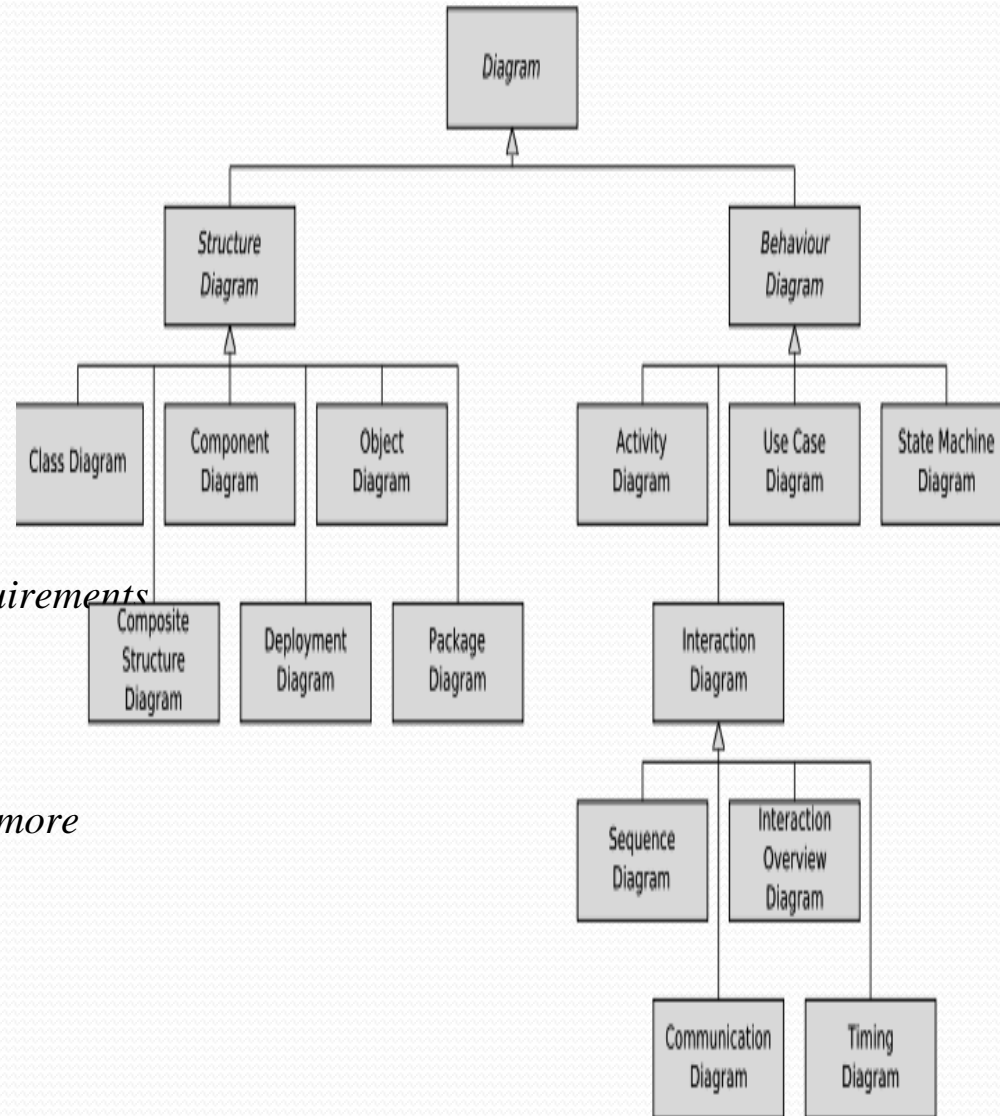
- Class Diagram,
- Object Diagram,
- Component Diagram,
- Composite Structure Diagram,
- Package Diagram,
- Deployment Diagram.

### Behaviour Diagrams:

- Use Case Diagram  
*(used by some methodologies during requirements gathering);*
- Activity Diagram,
- State Machine Diagram.

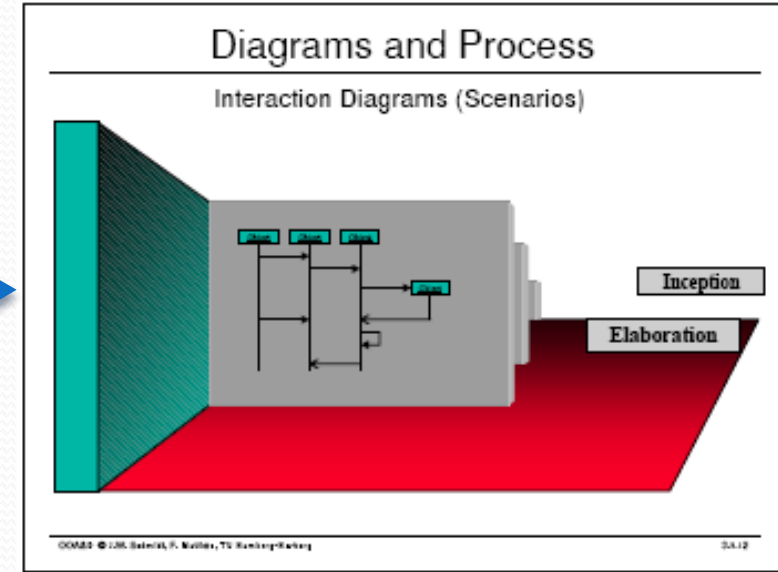
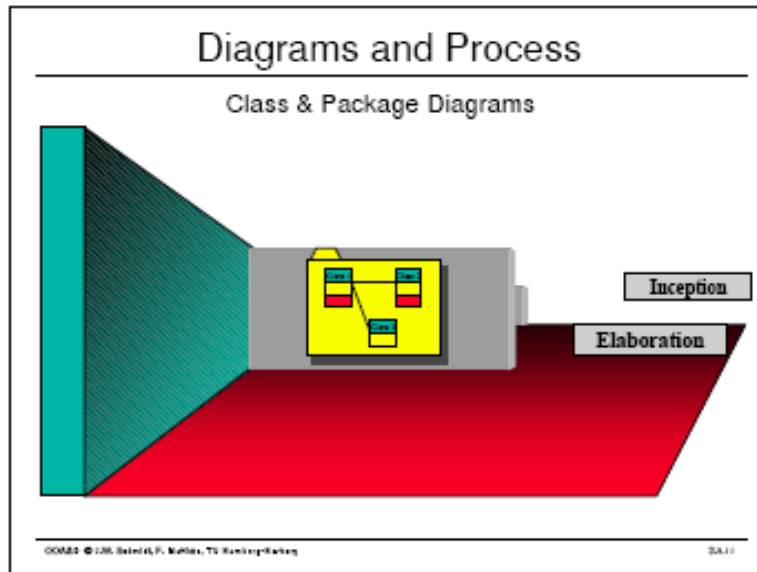
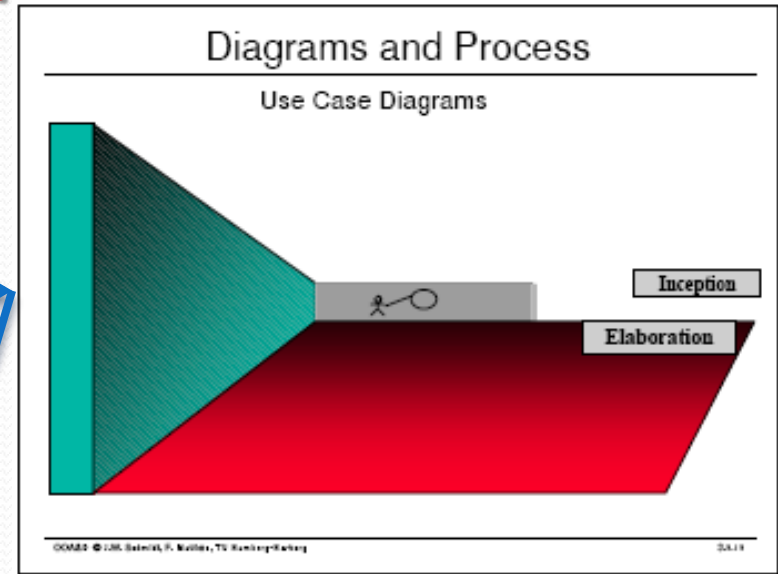
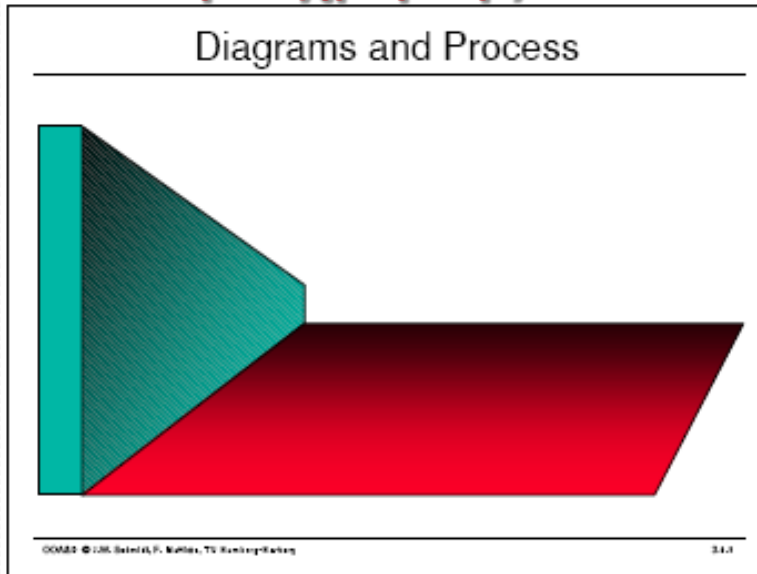
### Interaction Diagrams: *all derived from the more general Behaviour Diagram,*

- *Sequence Diagram,*
- *Communication Diagram,*
- *Timing Diagram,*
- *Interaction Overview Diagram.*



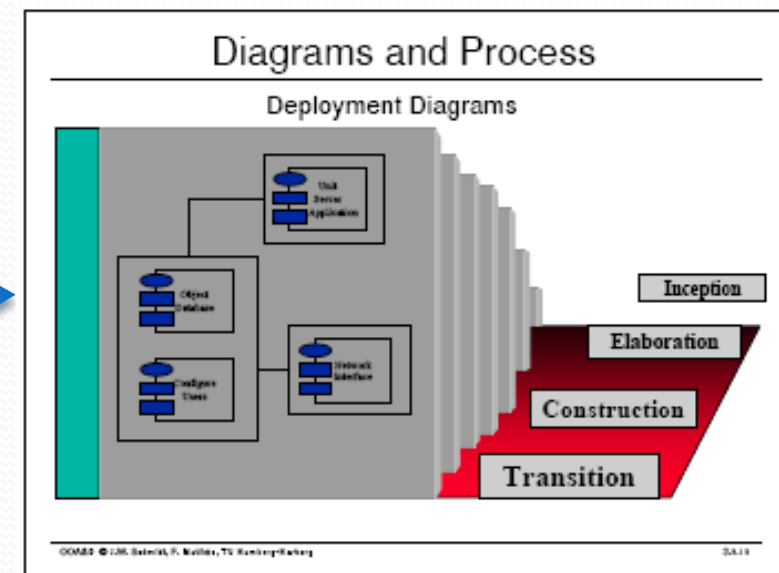
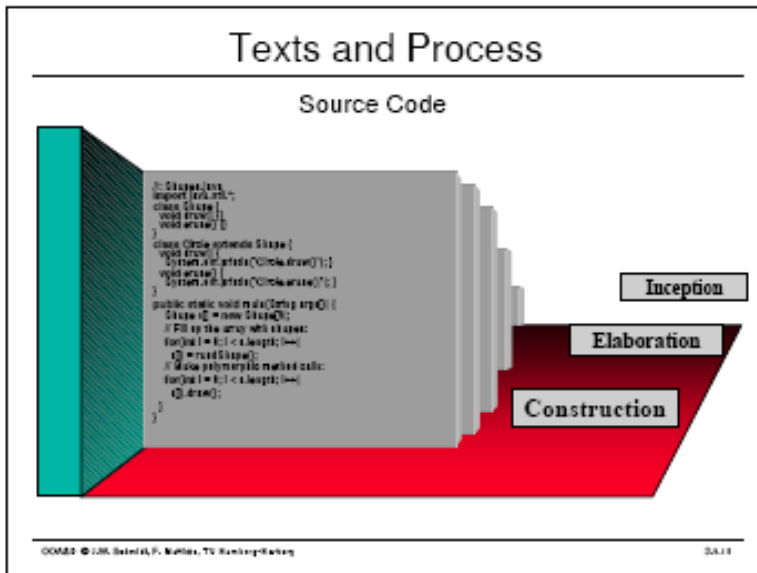
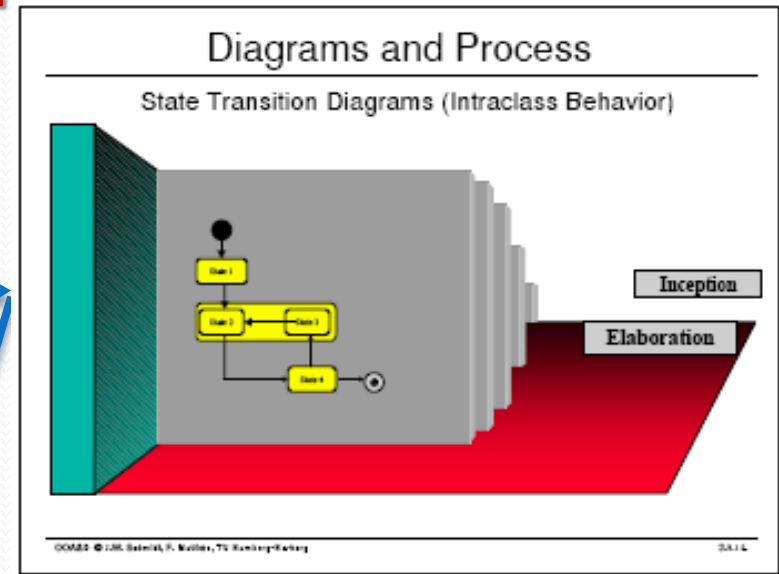
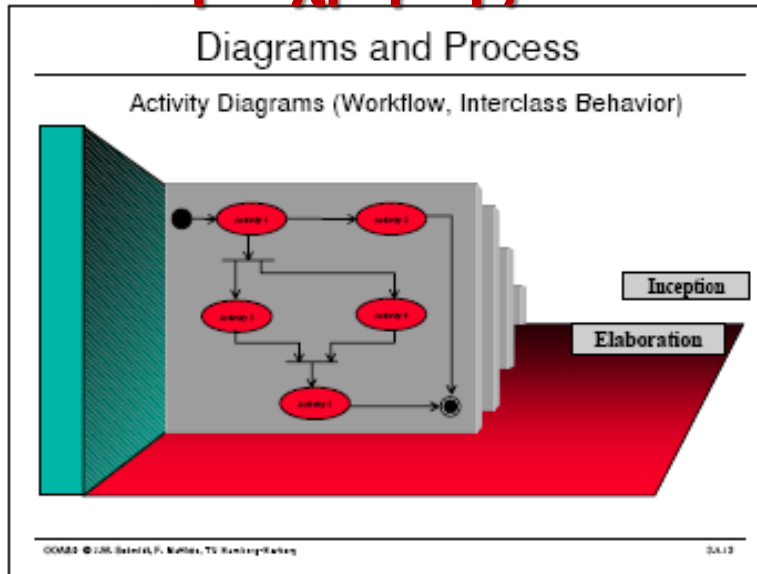
# Modeling Languages: UML (Unified Modeling Language)

## Μια σειρά χρήσης των διαγραμμάτων



# Modeling Languages: UML (Unified Modeling Language)

## Μια σειρά χρήσης των διαγραμμάτων



# Modeling Languages: **UML** (Unified Modeling Language)

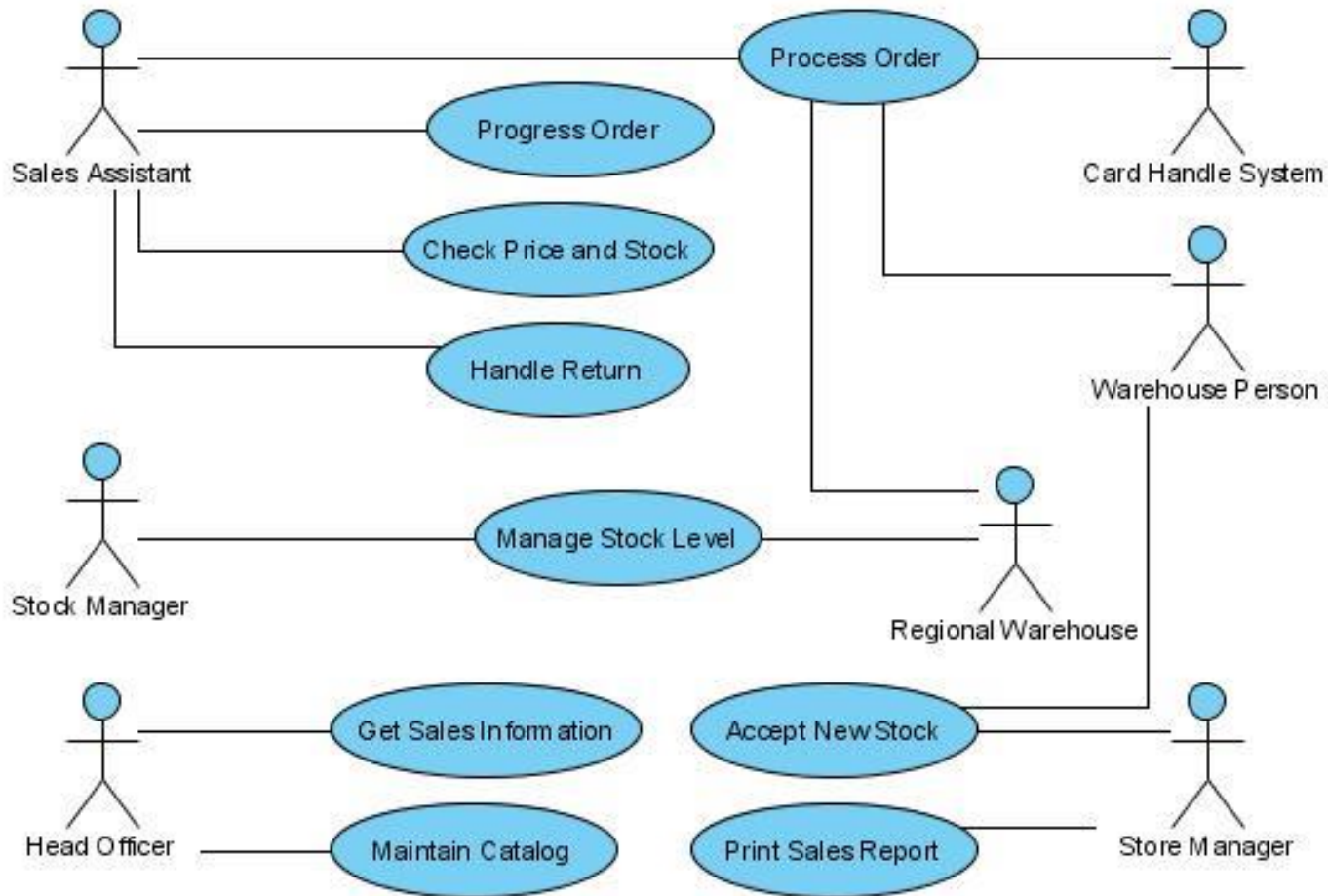
## Use Case diagram, Class Diagram and Sequence Diagram

Στο μάθημα θα ασχοληθούμε με 3 από τα σπουδαιότερα διαγράμματα

- Το διάγραμμα Use Case
- Το διάγραμμα Class
- Το διάγραμμα Sequence

# Modeling Languages: UML (Unified Modeling Language)

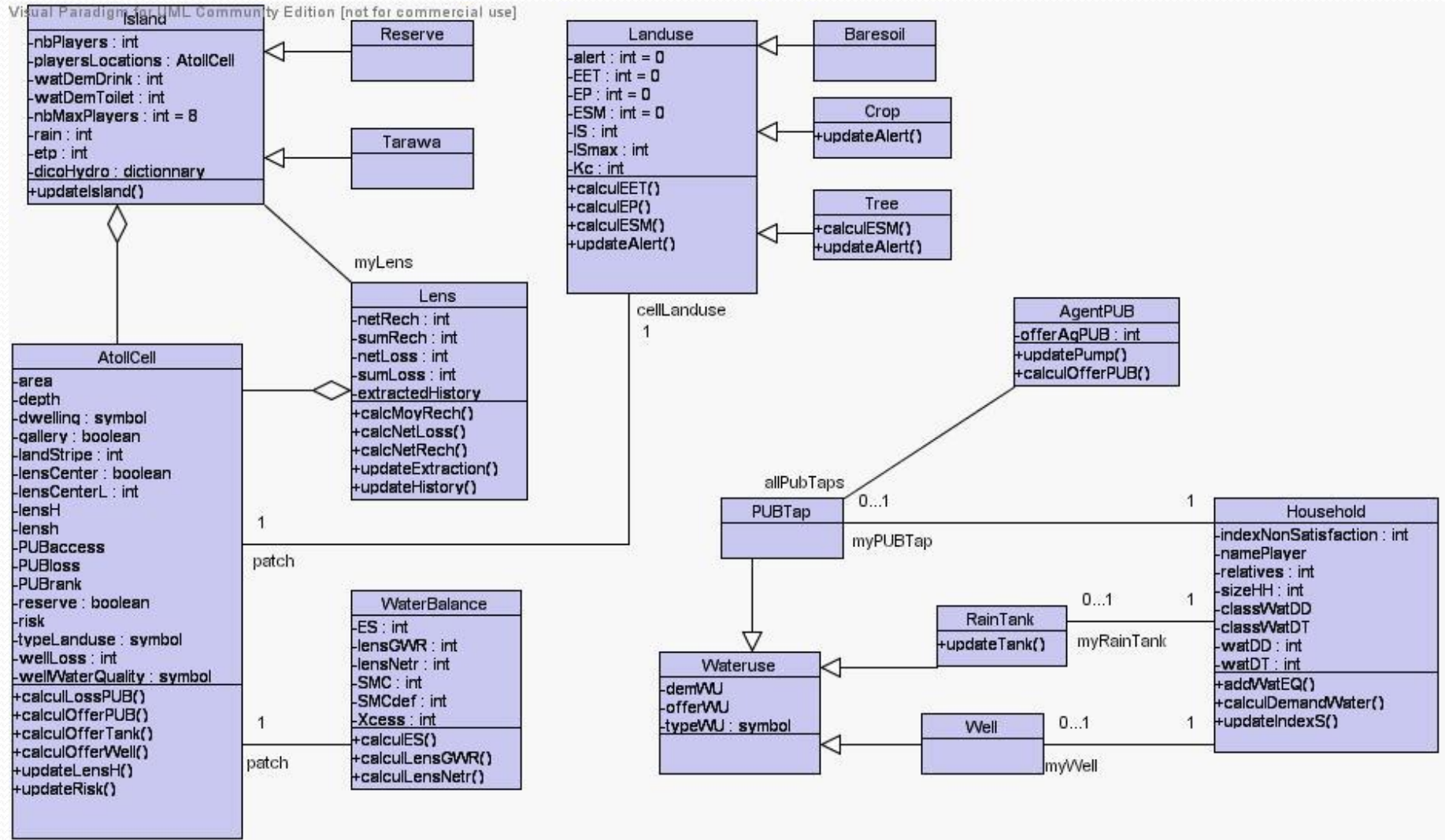
## Use Case Diagram in Visual Paradigm



# Modeling Languages: UML (Unified Modeling Language)

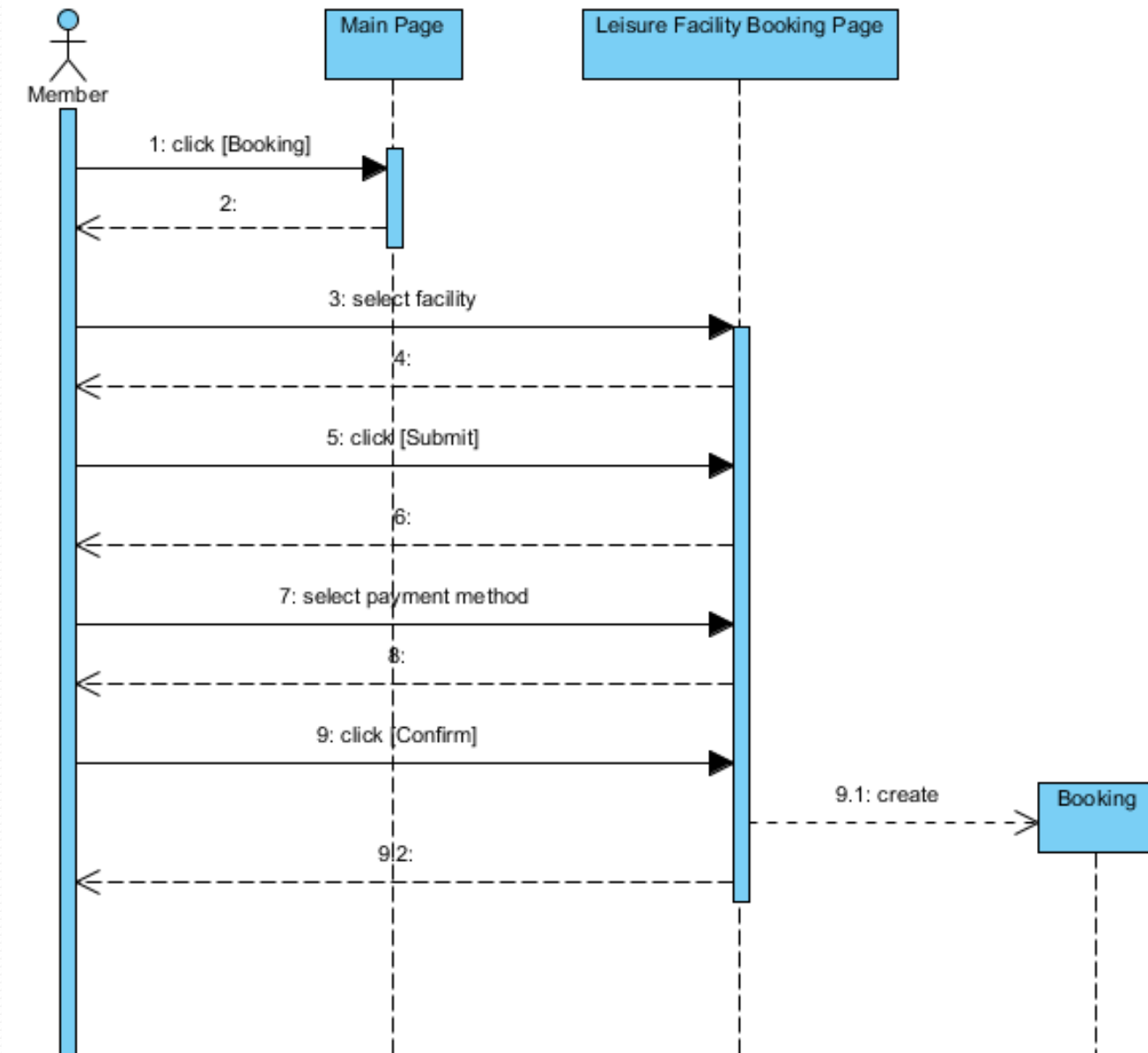
## Class Diagram in Visual Paradigm

Visual Paradigm for UML Community Edition [not for commercial use]



# Modeling Languages: UML (Unified Modeling Language)

## Sequence Diagram in Visual Paradigm



UML Modeling Tools  
http://www.uml.org/

140 Kendrick Street,  
Building A Suite 300  
Needham, MA 02494, U.S.A.



Ph:+1-781-444 0404  
Fax: +1-781-444 0320  
Email: info@omg.org

- About Us
- Press Room
- Calendar
- Documents
- Members Only
- Technology
- Industries
- Programs

[Free UML Modeling Tool](#) Support all UML 2 diagrams Generate XML, PDF and JPG [visual-paradigm.com/freeumltool](http://visual-paradigm.com/freeumltool)  
[UML 2.1 Tool](#) .NET, Java, C++, XSD, DDL, PHP, CORBA, Python & more. Free Trial! [www.sparxsystems.com](http://www.sparxsystems.com)  
[Cameo Simulation Toolkit](#) Executable UML 2, Statemachines Activities and SysML Parametrics [www.magicdraw.com/si](http://www.magicdraw.com/si)  
Ads by Google

[ [Submit Your Company's Product or Service](#) ] [ [Search Vendor Directory](#) ] [ [Home](#) ]

## UML Vendor Directory Listing

### OMG Members

[Altova GmbH](#)- OMG Member (T)  
[Armstrong Process Group, Inc.](#)- OMG Member (I)  
[Atego](#)- OMG Member (C)  
[Cephas Consulting Corp.](#)- OMG Member (I)  
[E2E Technologies Ltd.](#)- OMG Member (I)  
[IDS Scheer AG](#)- OMG Member (D)  
[International Business Machines](#)- OMG Member (C)

[International Business Machines](#)- OMG Member (C)  
[MID GmbH](#)- OMG Member (I)  
[Mentor Graphics Corporation](#)- OMG Member (F)  
[Micro Focus International Plc](#)- OMG Member (I)  
[Milestone Consulting SC](#)- OMG Member (T)  
[Mithun Training & Consulting](#)- OMG Member (I)  
[NEC](#)- OMG Member (T)

[No Magic, Inc.](#)- OMG Member (C)  
[Objektum Solutions Limited](#)- OMG Member (I)  
[Oracle](#)- OMG Member (D)  
[Softeam](#)- OMG Member (P)  
[Sparx Systems](#)- OMG Member (C)  
[Technologic Arts](#)- OMG Member (I)  
[oose Innovative Informatik GmbH](#)- OMG Member (I)

### Non Members

[@-portunity](#)  
[Adriano Comai](#)  
[Advanced Concepts Center, LLC](#)  
[BCAS](#)  
[BERTEC](#)  
[CS Odessa](#)  
[Change Vision](#)  
[Change Vision](#)  
[Excel Software](#)  
[Freetutes.com](#)

[LIANTIS GmbH & Co. KG](#)  
[LucidChart](#)  
[Meta Integration Technology, Inc.](#)  
[Mia-Software](#)  
[North State Software](#)  
[Oak Lodge Consulting](#)  
[Object Computing, Inc. \(OCI\)](#)  
[Pathfinder Solutions](#)  
[PragmaDev](#)

[The Software Factory](#)  
[VEGA Group PLC](#)  
[Vico Open Modeling](#)  
[Visual Paradigm International](#)  
[Willert Software Tools](#)  
[Zeligsoft, Inc.](#)  
[Zindell Technologies](#)  
[tracemodeler.com](#)



# Free UML Modeling Tools:

## Visual Paradigm

Ανοίξτε τα παρακάτω tutorials της Visual Paradigm

- Use Case tutorial



Visual Paradigm for UML  
Writing effective use case tutorial

Tutorial



Visual Paradigm  
for UML

Written Date : January 03, 2011

### Writing effective use case tutorial

Drawing class diagrams

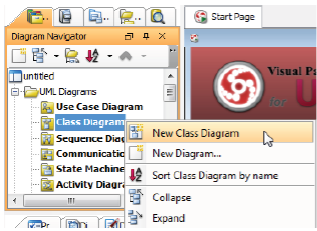
A [class diagram](#) shows the objects that are required and the relationships between them. Since it provides detailed information about the properties and interfaces of the classes, it can be considered as the main model and regard the other diagrams as supplementary models.

- Class tutorial



Creating class diagram

Right click Class Diagram on Diagram Navigator and select **New Class Diagram** from the pop-up menu to create a class diagram.




Create class diagram

- Sequence tutorial



Visual Paradigm for UML  
Constructing sequence diagram with existing classes

Tutorial



Visual Paradigm  
for UML

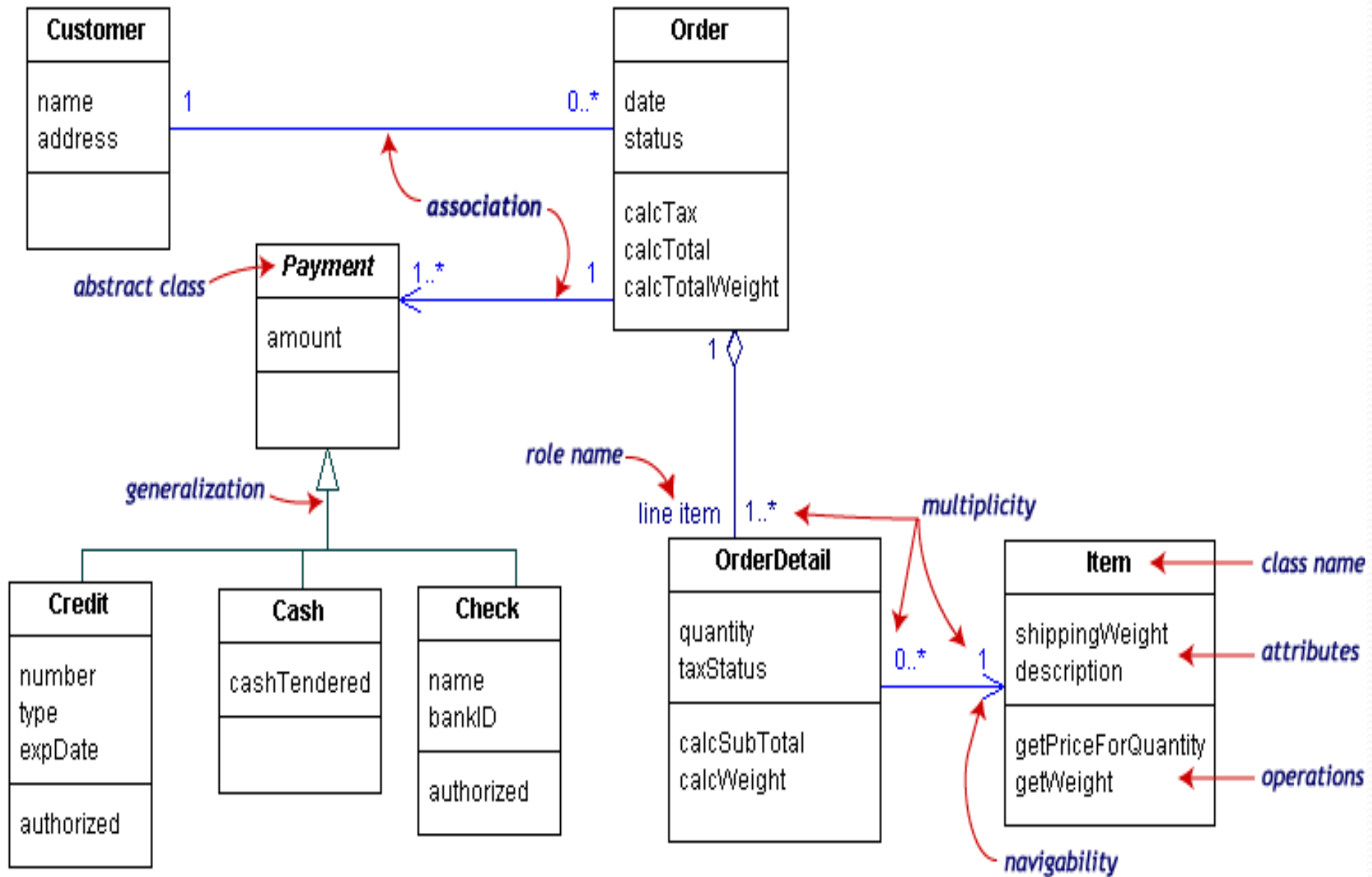
Constructing sequence diagram with existing classes

Written Date : June 14, 2010

For a full list of tutorials visit:

<http://www.visual-paradigm.com/product/vpuml/tutorials.jsp>

# Ασκήσεις



# Ασκήσεις(προαιρετικές)

# Σύνοψη Μαθήματος

# Βιβλιογραφία

- JAVA Tutorials

(<http://java.sun.com/docs/books/tutorial/>)

Επόμενο μάθημα