

# 08 Η γλώσσα UML (I)

## Τεχνολογία Λογισμικού

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2017-18

Δρ. Κώστας Σαΐδης ([saiko@di.uoa.gr](mailto:saiko@di.uoa.gr))

# Unified Modeling Language (UML)

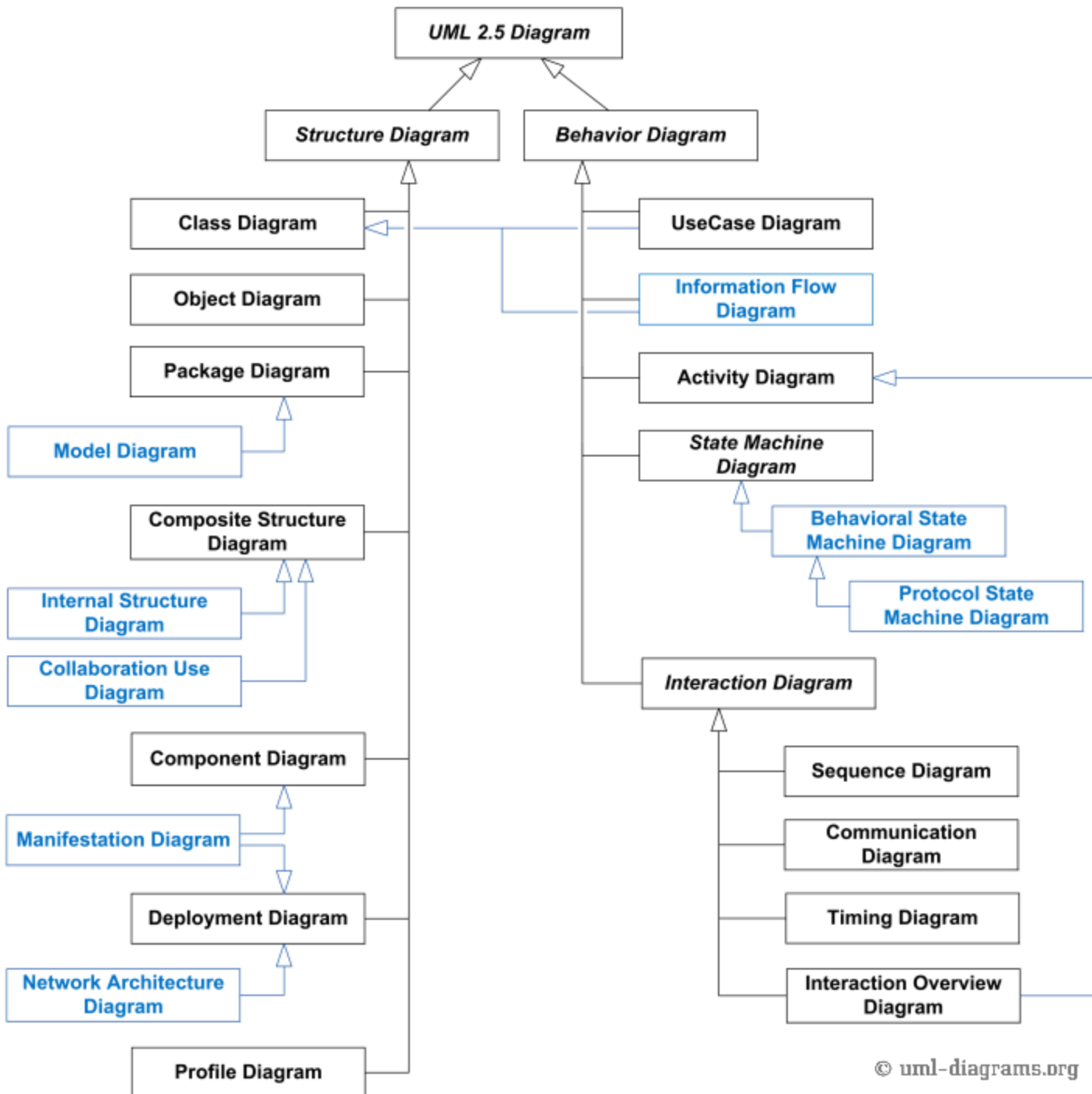
- Γλώσσα μοντελοποίησης γενικής χρήσης που στοχεύει στην παροχή ενός καθιερωμένου τρόπου οπτικοποίησης και επικοινωνίας των σημαντικών χαρακτηριστικών ενός συστήματος λογισμικού.
- ISO standard 19501:2005
- Τελευταία έκδοση 2.5 (2015)

# Στοιχεία και χαρακτηριστικά

- Ενοποιημένη μοντελοποίηση τόσο των απαιτήσεων όσο και του σχεδιασμού του λογισμικού
- Ανεξάρτητη του πεδίου εφαρμογής και της μεθοδολογίας ανάπτυξης του λογισμικού
- Είναι μια "γραφική - οπτική" γλώσσα
  - Δεν είναι αυστηρώς ορισμένη (διαφορετικές ερμηνείες ενός μοντέλου)
  - Δεν καλύπτει όλες τις πιθανές πτυχές (μη πλήρης μοντελοποίηση)
  - Δεν είναι γλώσσα προγραμματισμού

# Διαγράμματα της UML

- Το βασικό όχημα έκφρασης των χαρακτηριστικών του υπό μοντελοποίηση συστήματος
- Μοντελοποίηση τόσο στοιχείων συμπεριφοράς όσο και δομικών στοιχείων



# Μοντελοποίηση συμπεριφοράς

- Διαγράμματα περιπτώσεων χρήσης (Use-case diagrams)
- Διαγράμματα δραστηριότητας (Activity diagrams)
- Διαγράμματα μηχανών κατάστασης (State machine diagrams)
- Διαγράμματα ακολουθίας (Sequence diagrams)
- Διαγράμματα επικοινωνίας (Communication diagrams)
- Διαγράμματα χρονισμού (Timing diagrams)

# Μοντελοποίηση δομής

- Διαγράμματα κλάσεων (Class diagrams)
- Διαγράμματα συστατικών (Component diagrams)
- Διαγράμματα πακέτων (Package diagrams)
- Διαγράμματα "παράταξης / εγκατάστασης" (Deployment diagrams)
- Διαγράμματα σύνθετης δομής (Composite structure diagrams)

# Εργαλεία

- Διάφορα γραφικά εργαλεία (ελεύθερα / δωρεάν ή μη):
  - Visio
  - ArgoUML
  - Violet UML editor
  - UMLet
  - κ.ά
- Δοκιμάστε και επιλέξτε ένα που να ικανοποιεί όλα τα μέλη της ομάδας
- Προσωπική (καθαρά) προτίμηση:
  - PlantUML (text-based δημιουργία UML διαγραμμάτων)



# Διαγράμματα δομής

Τα διαγράμματα συμπεριφοράς θα τα κουβεντιάσουμε στην επόμενη διάλεξη

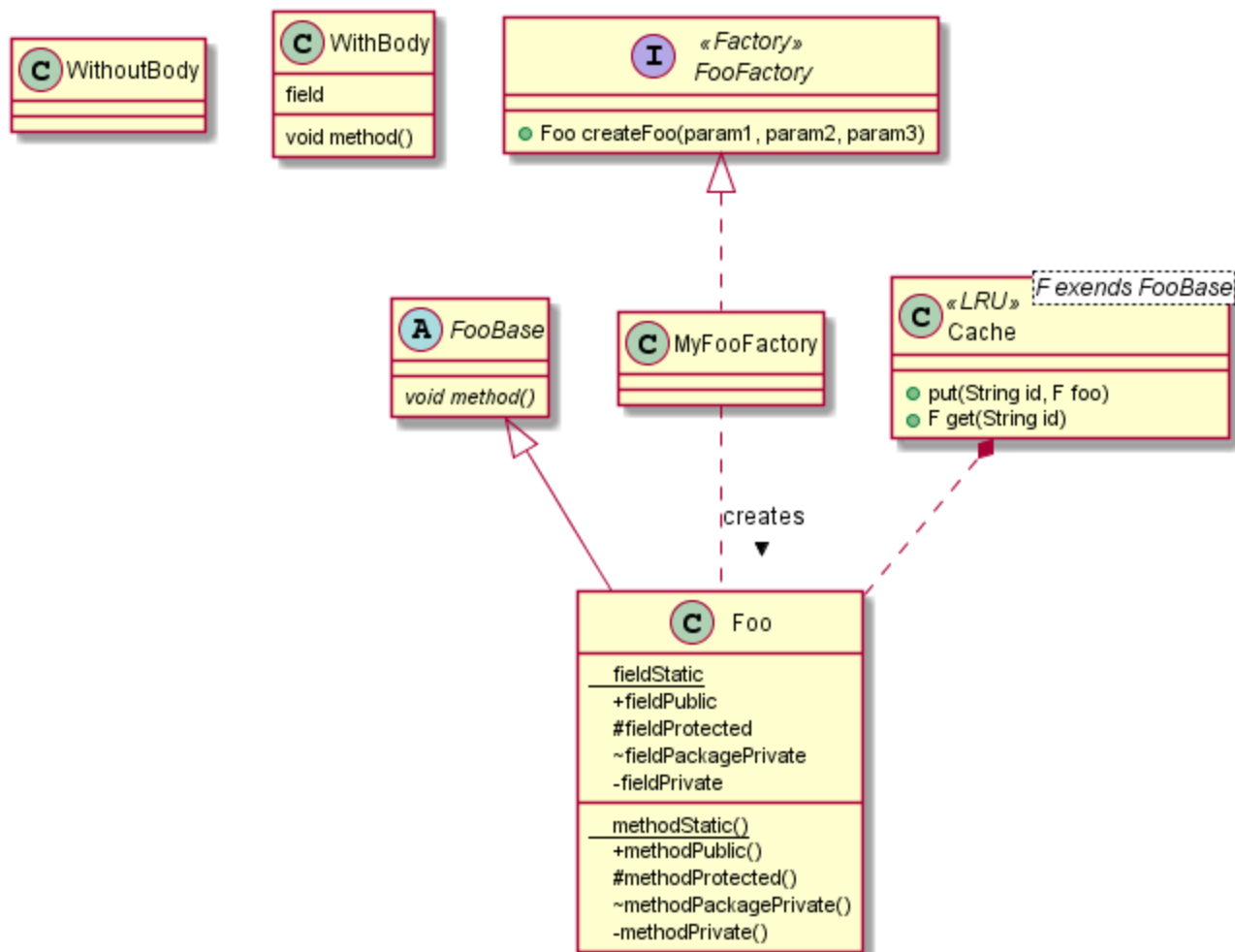
# Διαγράμματα κλάσεων (Class diagrams)

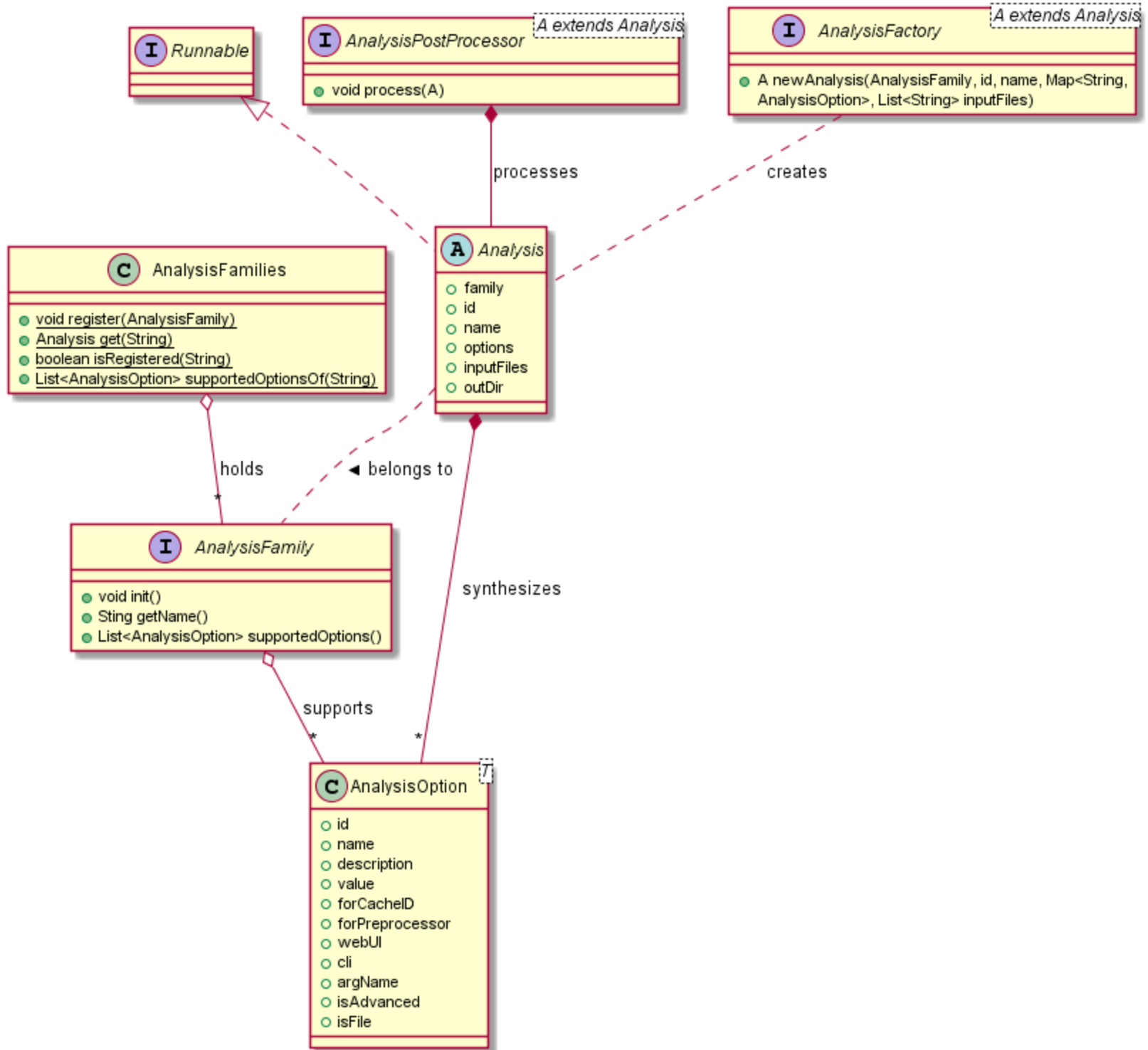
- Διαγράμματα αποτύπωσης του σχεδιασμού των κλάσεων (interface, class, abstract class), της δομής τους, των χαρακτηριστικών στοιχείων τους και των σχέσεων μεταξύ τους.
- Αντικειμενοστραφείς αρχιτεκτονικές (object-oriented).

# Βασικά στοιχεία της UML

- Class / Abstract class / Interface / Stereotype
- Property (attribute, field) / Operation (method)
- Visibility / Multiplicity / Constraint
- Association / Aggregation / Composition / Usage
- Generalization

# Παραδείγματα





# Συγκεντρωτικά

<https://www.uml-diagrams.org/class-reference.html>

# Διαγράμματα συστατικών (Component diagrams)

- Διαγράμματα αποτύπωσης των συστατικών του λογισμικού σε ένα υψηλότερο επίπεδο αφάιρεσης, όπου η έμφαση δίνεται στις διεπαφές, αλληλεπιδράσεις και αλληλεξαρτήσεις μεταξύ των συστατικών.
- Αρχιτεκτονικές επικεντρωμένες στα συστατικά (component-oriented).

Συστατικό = κλάση ή σύνολο κλάσεων που υλοποιούν μια -υψηλού αρχιτεκτονικά επιπέδου- λειτουργία

# Βασικά στοιχεία της UML

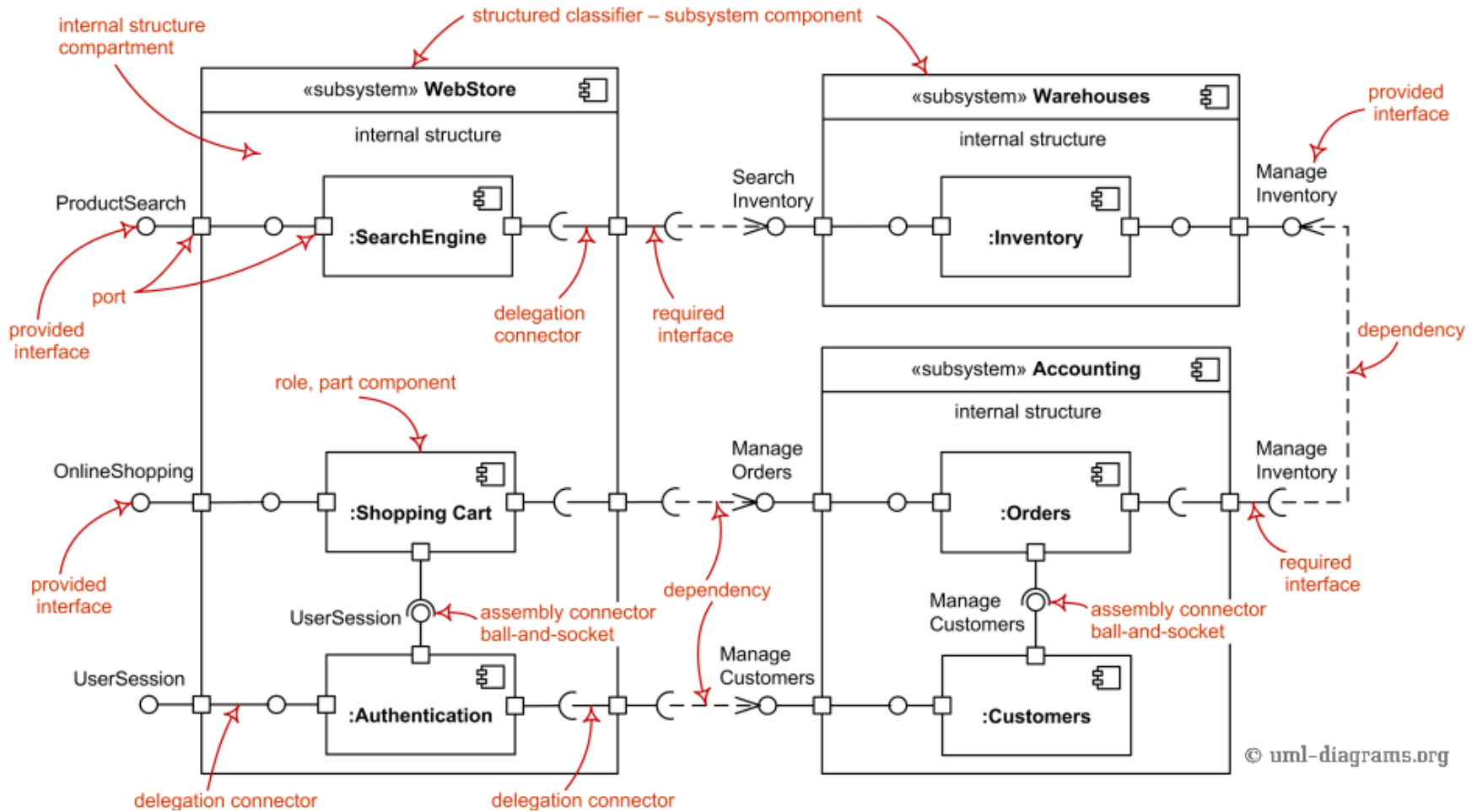
- Συστατικό
  - Μπορεί να είναι μια "λογική" οντότητα (π.χ. συστατικά επιχειρησιακής λογικής)
  - Μπορεί να είναι μια "φυσική" οντότητα κάποιου συγκεκριμένου framework (π.χ. EJB, WSDL, κ.ο.κ)



# Επιπρόσθετα

- Τα συστατικά παρέχουν (provide) και απαιτούν (require) διεπαφές (interfaces).
- Τα συστατικά διαθέτουν επιμέρους "μέρη" (parts).
- Τα συστατικά διαθέτουν θύρες (ports) για την επικοινωνία με το περιβάλλον τους.
- Τα συστατικά διαθέτουν συνδέσεις (connectors) μεταξύ τους μέσω των θυρών ή των διεπαφών.

# Παράδειγμα



# Συγκεντρωτικά

<https://www.uml-diagrams.org/component-diagrams-reference.html>

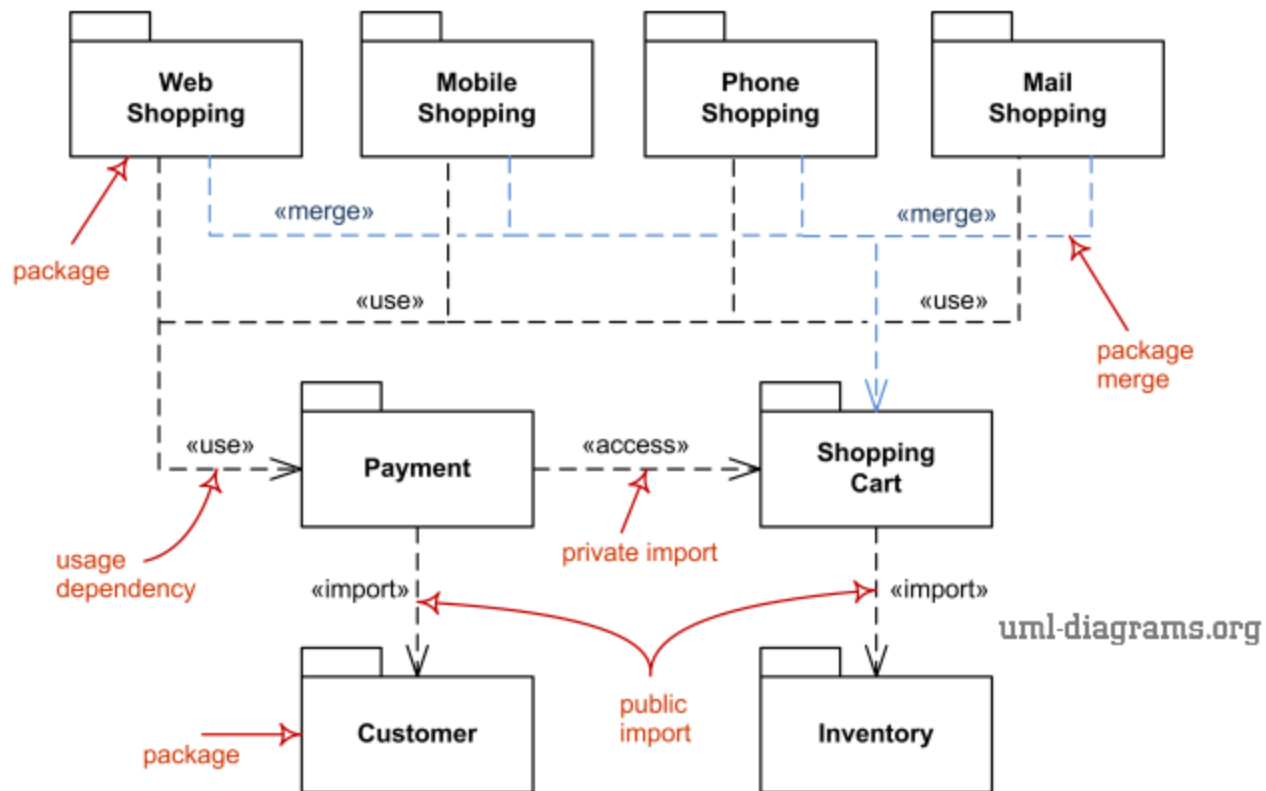
# Διαγράμματα πακέτων (Package diagrams)

- Αποτύπωση του λογισμικού σε επίπεδο πακέτων (packages), η οποία στοχεύει κυρίως στην -υψηλού αρχιτεκτονικά επιπέδου- παρουσίαση των εξαρτήσεων (dependencies).
- Πολυ-επίπεδες αρχιτεκτονικές.

# Βασικά στοιχεία της UML

- Πακέτα (packages) που χρησιμοποιούν (private / public import, use) άλλα πακέτα.
- Ένωση πακέτων (package merge).

# Παράδειγμα



# Συγκεντρωτικά

<https://www.uml-diagrams.org/package-diagrams-reference.html>

# Διαγράμματα "παράταξης / εγκατάστασης" (Deployment diagrams)

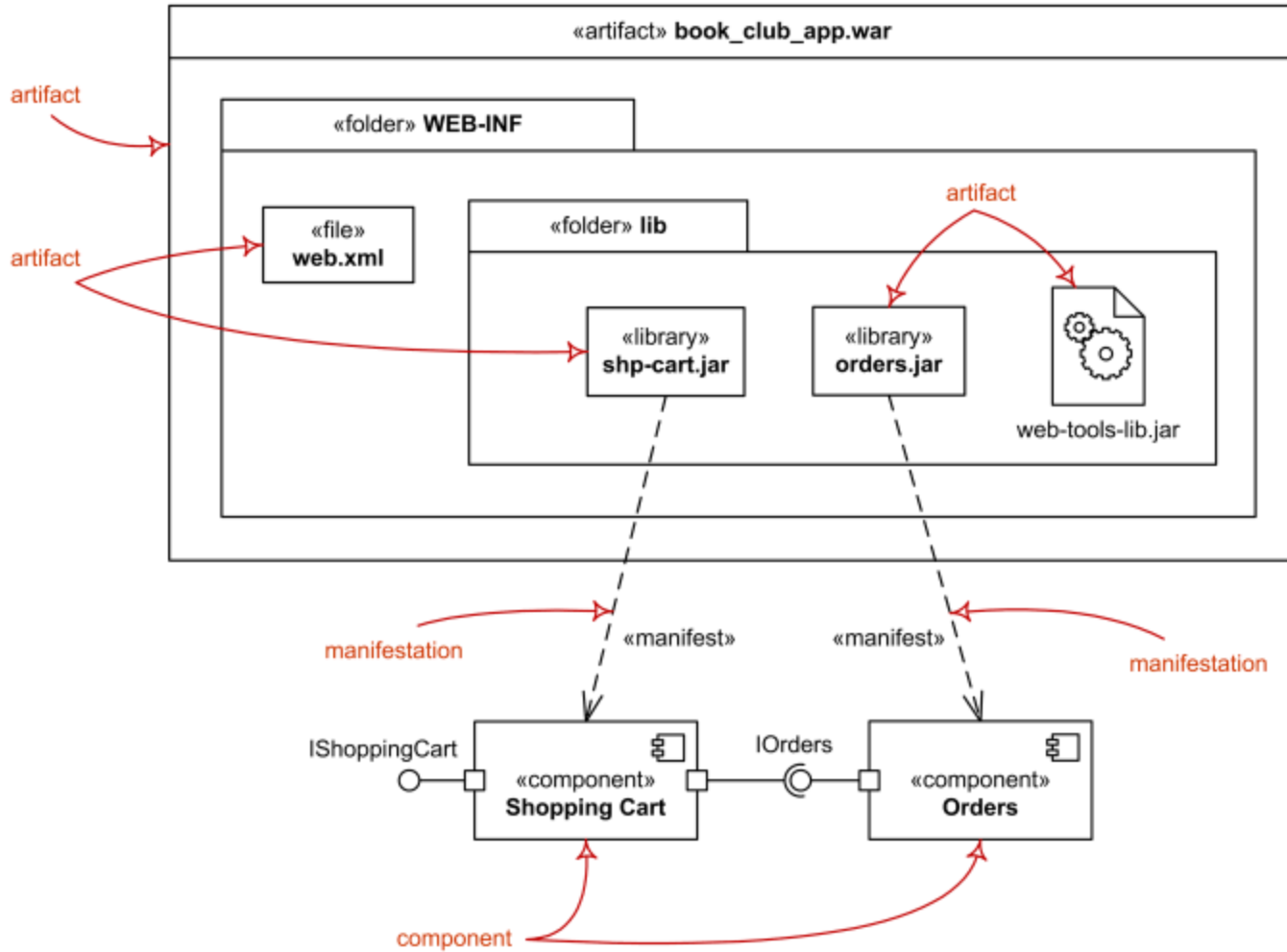
- Αποτύπωση της αρχιτεκτονικής του συστήματος μέσω της εγκατάστασης (deployment) των software artifacts σε συγκεκριμένες θέσεις (deployment targets).
- Φυσική αρχιτεκτονική (τι εγκαθίσταται που).

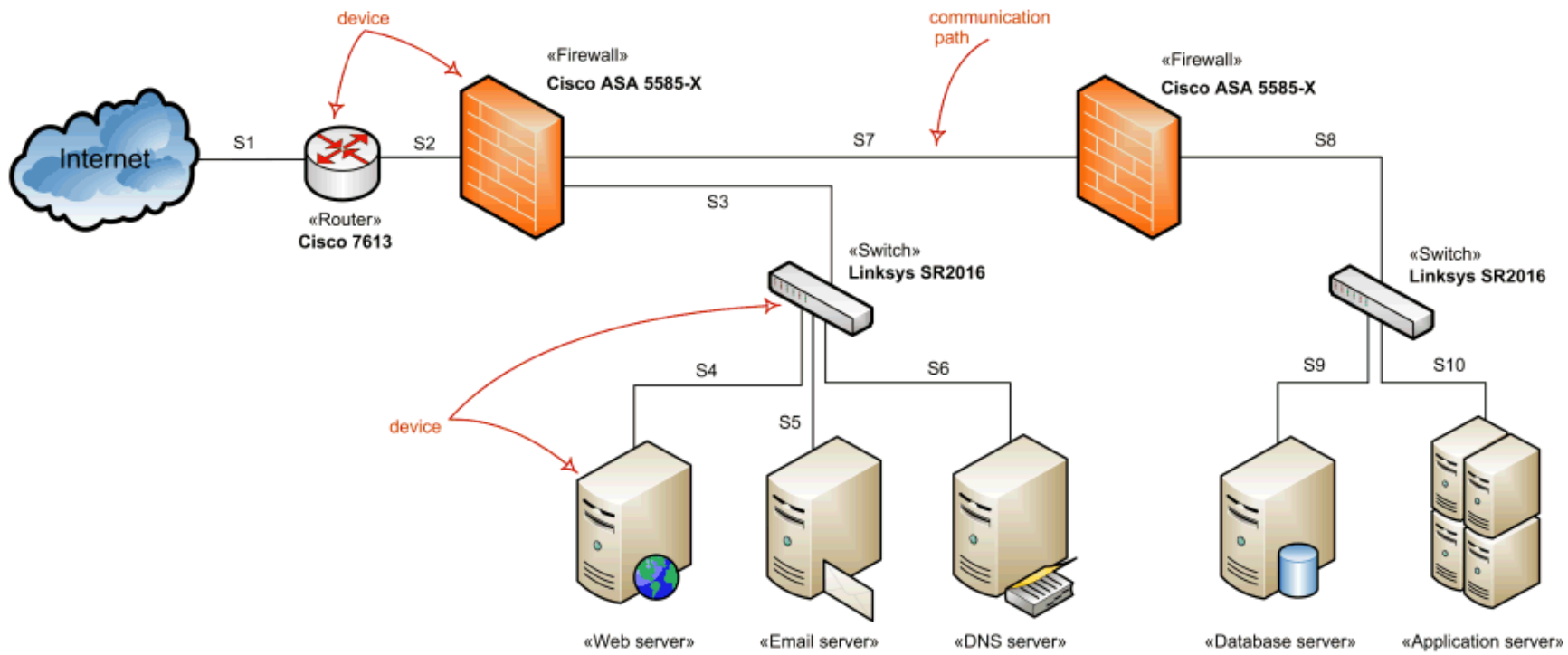


# Βασικά στοιχεία της UML

- Software artifacts (π.χ. αρχεία war, jar)
- Manifestations (π.χ. κάποιων components)
- Deployment targets (π.χ. εγκαθίστανται σε κάποιον application server)

# Παραδείγματα





# Συγκεντρωτικά

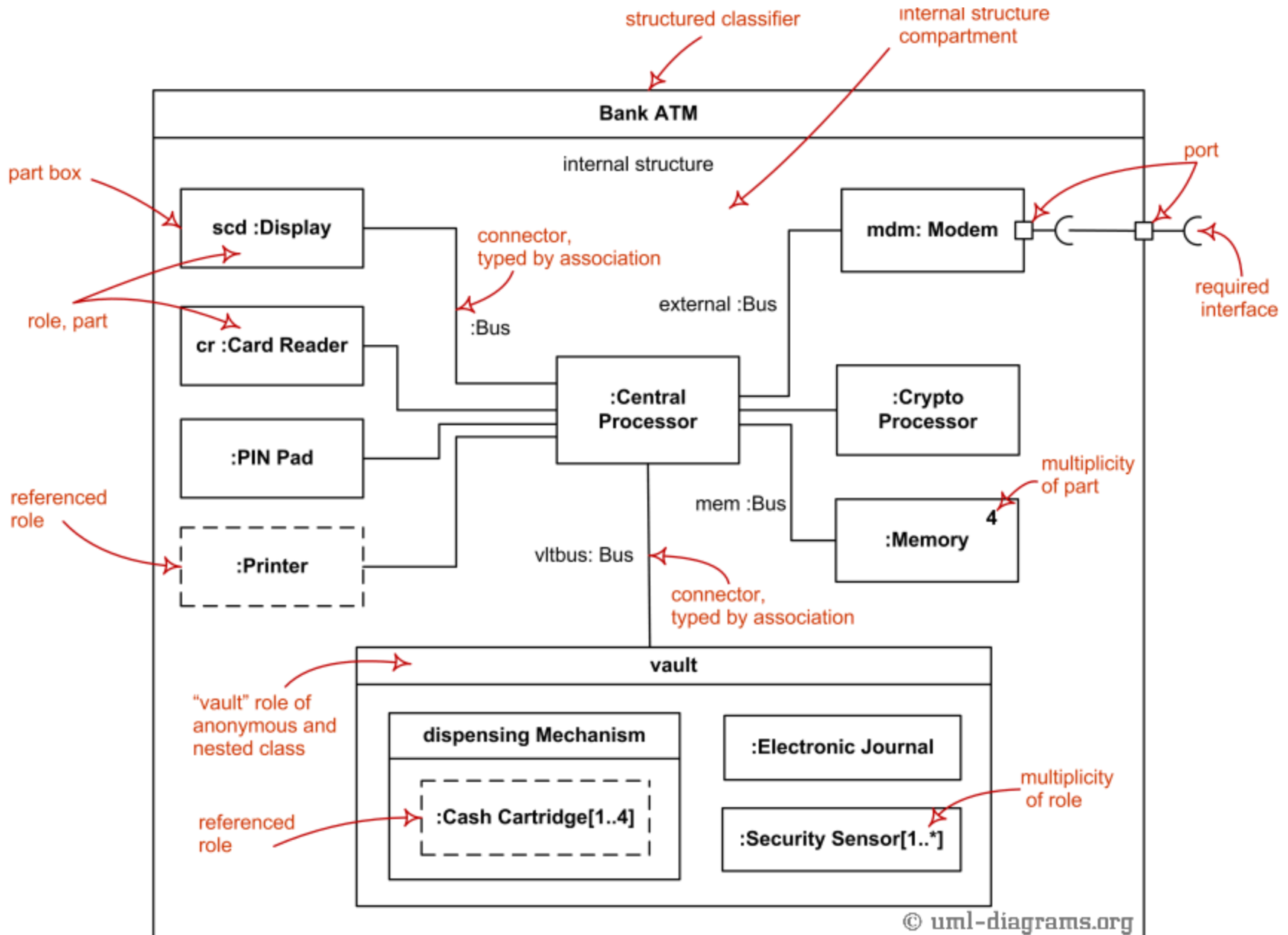
<https://www.uml-diagrams.org/deployment-diagrams-reference.html>

# Διαγράμματα σύνθετης δομής (Composite structure diagrams)

Αποτύπωση μιας σύνθετης δομής, όπως είναι η εσωτερική δομή ενός συστατικού / πακέτου.

Χρήσιμα όταν οι επιμέρους έννοιες/ρόλοι (κλάσεις, συστατικά, πακέτα) δεν είναι γνωστές ή δεν απαραίτητες για την κατανόηση μια σύνθετης δομής.

# Παράδειγμα



# Συγκεντρωτικά

<https://www.uml-diagrams.org/composite-structure-diagrams-reference.html>

# Εργασία

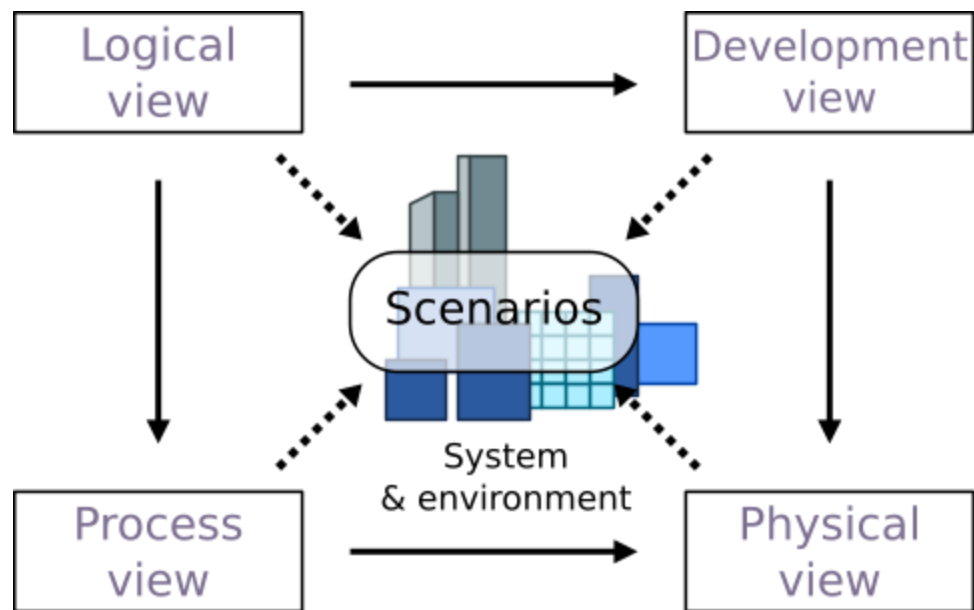
- Στο δεύτερο παραδοτέο της εργασίας θα πρέπει να αποτυπώσετε το σχεδιασμό και την αρχιτεκτονική του λογισμικού σας με βάση τα παραπάνω διαγράμματα (εσείς θα επιλέξετε ποια).
- Συνήθως τα class, component, deployment διαγράμματα είναι απαραίτητα.



# Θυμηθείτε

Από τη διάλεξη 5 (αρχιτεκτονικά πρότυπα)

# Πολλές αρχιτεκτονικές οπτικές (4+1)



By mpan - Based on File:4+1 Architectural View Model.jpg by User:Mdd, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=50144028>

# Ειδικότερα

- Logical view: έμφαση στη λειτουργικότητα του συστήματος σε υψηλό επίπεδο
- Physical view: έμφαση στην τοπολογία και διασύνδεση των συστατικών του σε φυσικό επίπεδο (deployment)
- Development view: έμφαση στην οπτική του προγραμματιστή
- Process view: έμφαση στη δυναμική συμπεριφορά του συστήματος κατά την εκτέλεσή του (απόδοση, κλιμάκωση, κτλ.)
- Scenarios - Use case view: έμφαση στη χρηστική πλευρά του συστήματος και στους σχετικούς ελέγχους αποδοχής

## 2ο Παραδοτέο

Πληρέστερο αν παρουσιάζει τις 4 οπτικές (ΔΕΝ είναι απαραίτητο να ασχοληθείτε με το process view στο πλαίσιο της εργασίας).