

Unified Modeling Language (UML)

Three Amigos
Grady Booch
James Rumbaugh
Ivar Jacobson

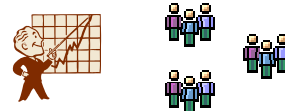
Ενοποιημένη Γλώσσα Μοντελοποίησης

1

UML



Μεγάλα Έργα Λογισμικού
= επικοινωνία



Πολλαπλές μονάδες, πολλαπλές γενιές, πολλοί προγραμματιστές, πολλοί πελάτες-χρήστες επιβάλλουν τη χρήση **κοινής ορολογίας**

Σε άλλα τεχνικά έργα η χρήση κοινής ορολογίας και συμβολισμών είναι αυτονόητη και καθιερωμένη

2

UML

Τυπικός ορισμός:

Η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language, στο εξής UML) είναι μία γραφική γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται για τον προσδιορισμό, οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων (artifacts) ενός συστήματος λογισμικού.

Πρακτικά:

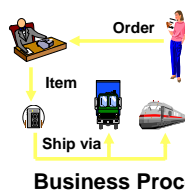
Η UML είναι μία γλώσσα μοντελοποίησης (σύνολο από διαγράμματα). (Δεν είναι γλώσσα προγραμματισμού)

Χρήση:

- με οποιαδήποτε διαδικασία
- σε οποιοδήποτε στάδιο ανάπτυξης
- για οποιαδήποτε τεχνολογική πλατφόρμα

3

Μοντελοποίηση ?



"Modeling captures essential parts of the system."
Dr. James Rumbaugh

Business Process



Visual Modeling is modeling using standard graphical notations



Computer System

4

Μοντελοποίηση

- Ένα μοντέλο είναι μία **αναπαράσταση** σε ένα συγκεκριμένο μέσο, κάποιας οντότητας στο ίδιο ή σε **άλλο μέσο**.
- Το μοντέλο αναπαριστά τα σημαντικά (από μία ορισμένη οπτική γωνία), στοιχεία της οντότητας που μοντελοποιείται, απλοποιώντας ή παραλείποντας τα υπόλοιπα.
- Το μέσο επάνω στο οποίο αναπτύσσεται το μοντέλο είναι βολικό για επεξεργασία.
- Για παράδειγμα το μοντέλο ενός κτιρίου κατασκευάζεται σε χαρτί ή το μοντέλο ενός αεροσκάφους σε μία μακέτα.

5

Πλεονεκτήματα ενός μοντέλου

- Ακριβής καθορισμός των απαιτήσεων έτσι ώστε όλοι οι εμπλεκόμενοι να τις κατανοούν με κοινό τρόπο. Μοντέλα ενός συστήματος λογισμικού κατασκευάζονται ώστε οι αναλυτές, προγραμματιστές, διαχειριστές έργων, πελάτες και τελικοί χρήστες να κατανοήσουν το σύστημα από την οπτική γωνία που τους ενδιαφέρει.
- Μελέτη του τρόπου σχεδίασης. Το μοντέλο ενός συστήματος λογισμικού παρέχει τη δυνατότητα διερεύνησης διαφόρων αρχιτεκτονικών, με οικονομικό τρόπο, αρκετά πριν από την κωδικοποίηση.

6

Πλεονεκτήματα ενός μοντέλου

- Παραγωγή χρήσιμων πρωτοτύπων. Το μοντέλο ενός συστήματος λογισμικού μπορεί να χρησιμοποιηθεί για την παραγωγή πρωτοτύπων τα οποία με τη σειρά τους θα αξιοποιηθούν για τη διερεύνηση των απαιτήσεων.
- Διαχείριση περίπλοκων συστημάτων. Τα σύγχρονα συστήματα λογισμικού χαρακτηρίζονται από υψηλή πολυπλοκότητα όσον αφορά το μέγεθος τους, τον αριθμό των λειτουργιών τους, τον αριθμό των μονάδων που τα απαρτίζουν, το μέγεθος της ομάδας ανάπτυξης κ.ο.κ. Ένα μοντέλο μπορεί να συμβάλει στην αντιμετώπιση της πολυπλοκότητας εισάγοντας διάφορα επίπεδα αφάιρησης, αποκρύπτοντας τον τεράστιο όγκο των λεπτομερειών.

7

UML

Η UML αποτυπώνει τόσο τη στατική δομή, όσο και τη δυναμική συμπεριφορά ενός συστήματος.

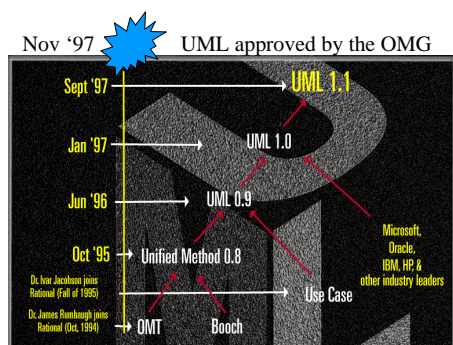
Ένα αντικειμενοστραφές σύστημα μοντελοποιείται ως μία συλλογή αντικειμένων που αλληλεπιδρούν για την εκτέλεση μίας λειτουργίας η οποία είναι τελικά αξιοποιήσιμη από τον χρήστη του συστήματος.

Η **στατική δομή** καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους.

Η **δυναμική συμπεριφορά** προσδιορίζει την εξέλιξη των αντικειμένων σε σχέση με τον χρόνο και την επικοινωνία μεταξύ τους.

8

Ιστορία της UML



9

Partners

- Rational Software Corporation
 - Hewlett-Packard
 - I-Logix
 - IBM
 - ICON Computing
 - Intellicorp
 - MCI Systemhouse
 - Microsoft
 - ObjecTime
 - Oracle
 - Platinum Technology
 - Taskon
 - Texas Instruments/Sterling Software
 - Unisys
 - ...
- > ισχυρή δυναμική

10

Περιπτώσεις Χρήσης

- Τα διαγράμματα **περιπτώσεων χρήσης** στη UML χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς ενός συστήματος, υποσυστήματος ή κλάσης, **όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη.**
- Τα διαγράμματα περιπτώσεων χρήσης διαμερίζουν τη λειτουργικότητα του συστήματος σε **συναλλαγές που έχουν νόημα για τους χαρακτήρες** (actors) - ιδανικούς χρήστες του συστήματος. Τα επιμέρους τμήματα της λειτουργικότητας ονομάζονται περιπτώσεις χρήσης.
- Το σύνολο των περιπτώσεων χρήσης συνιστούν τη συμπεριφορά του συστήματος. Ο τυπικός ορισμός μιας περίπτωσης χρήσης είναι **μία ακολουθία συναλλαγών που πραγματοποιείται από το σύστημα για την παραγωγή μετρήσιμων αποτελεσμάτων που έχουν νόημα για τον χρήστη.**

11

Περιπτώσεις Χρήσης

- Σε κάθε διάγραμμα περίπτωσης χρήσης απεικονίζεται ένας χρήστης του συστήματος (άνθρωπος ή άλλο σύστημα) ως ένα σχηματικό ανθρωπάκι (**stick person**).
- Η ίδια η περίπτωση χρήσης, ως σύνολο λειτουργιών που έχουν κάποιο νόημα για το χρήστη, απεικονίζεται ως μία **έλλειψη**.
- Ο χρήστης ενεργοποιεί μία περίπτωση χρήσης αναμένοντας την εκτέλεση κάποια συμπεριφοράς. Η συσχέτιση μεταξύ χρήστη και περίπτωσης χρήσης απεικονίζεται με μία ακμή μεταξύ τους ενώ η φορά της ενεργοποίησης με τη χρήση προσανατολισμένης ακμής.

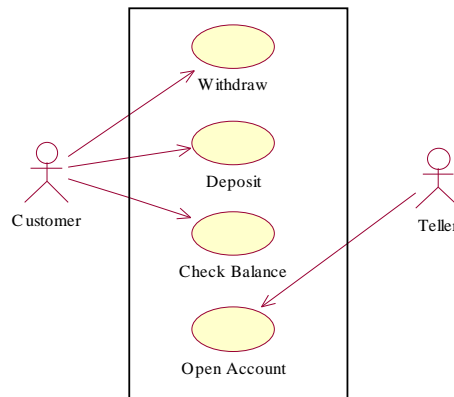
12

Περιπτώσεις Χρήσης

- Η απόφαση σχετικά με τις απαιτήσεις που πρόκειται να υλοποιηθούν στο σύστημα, είναι **αρμοδιότητα και δικαίωμα των πελατών**
- Ένας χαρακτήρας που αλληλεπιδρά με το σύστημα υπό διαφορετικό ρόλο κάθε φορά, αναγνωρίζεται ως διαφορετικός χρήστης. (Π.χ. ένας καθηγητής μπορεί να λειτουργεί είτε ως καθηγητής (Faculty) είτε ως πρόεδρος τμήματος).
- Δράσεις που ενεργοποιούνται από το ίδιο το σύστημα δεν καταγράφονται ως περιπτώσεις χρήσης

13

Περιπτώσεις Χρήσης



14

Περιπτώσεις Χρήσης

- Ο ορισμός μιας περίπτωσης χρήσης περιλαμβάνει όλες τις δυνατές συμπεριφορές που εμπρικλείει (**κανονική** και **εναλλακτική** συμπεριφορά)
- Η άποψη του συστήματος βάσει ενός μοντέλου περιπτώσεων χρήσης θεωρεί ότι η εκτέλεση κάθε περίπτωσης χρήσης είναι ανεξάρτητη από τις υπόλοιπες (**ορθογωνικότητα**), παρόλο που η υλοποίηση του συστήματος μπορεί να συνεπάγεται έμμεσες εξαρτήσεις μεταξύ τους λόγω κοινών αντικειμένων.
- Η αναλυτική περιγραφή μιας περίπτωσης χρήσης προσδιορίζεται με άλλους τύπους διαγραμμάτων της UML όπως τα διαγράμματα καταστάσεων, διαγράμματα αλληλεπίδρασης (διαγράμματα ακολουθίας ή/και διαγράμματα συνεργασίας) ή και άτυπες περιγραφές κειμένου.

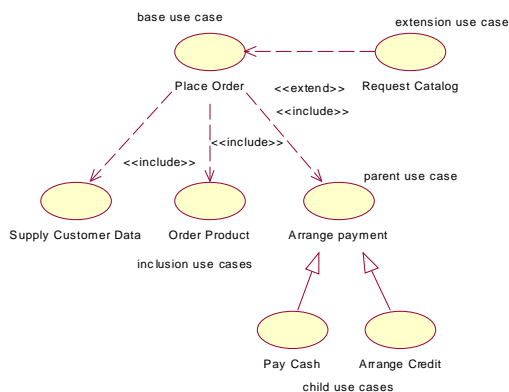
15

Περιπτώσεις Χρήσης

Μία περίπτωση χρήσης μπορεί να συμμετέχει σε διάφορες σχέσεις με άλλες περιπτώσεις χρήσης ή χαρακτήρες

Συσχέτιση	Το μονοπάτι επικοινωνίας μεταξύ ενός χαρακτήρα και μιας περίπτωσης χρήσης στην οποία συμμετέχει	—————
Επέκταση	Η προσθήκη επιπρόσθετης λειτουργικότητας σε μία βασική περίπτωση χρήσης (η προσθήκη εξαρτάται από το context)<< extend >>.....
Γενίκευση	Μία συσχέτιση μεταξύ μιας γενικής περίπτωσης χρήσης και μιας ειδικότερης που κληρονομεί στοιχεία συμπεριφοράς και προσθέτει νέα χαρακτηριστικά.	—————>
Περιεκτικότητα	Η προσθήκη επιπρόσθετης λειτουργικότητας σε μία βασική περίπτωση χρήσης (η οποία περιγράφει σαφώς την περίπτωση που εμπεριέχεται)<< include >>..... ¹⁶

Περιπτώσεις Χρήσης



17

Τεκμηρίωση Περιπτώσεων Χρήσης

a. Σύντομη περιγραφή

- Περιγραφή σε μία ή δύο σειρές της συμπεριφοράς που εκτελείται και των χρηστών της συγκεκριμένης περίπτωσης

b. Προ-συνθήκες

- Συνθήκες που θα πρέπει να ισχύουν ώστε να είναι δυνατή η έναρξη της περίπτωσης χρήσης

c. Κύρια ροή γεγονότων

- Περιγραφή υπό μορφή κειμένου της ακολουθία των γεγονότων που λαμβάνουν χώρα για την υλοποίηση της συμπεριφοράς.

d. Εναλλακτική ροή γεγονότων

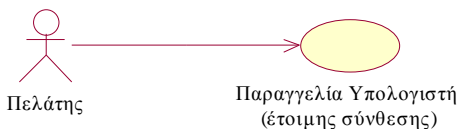
- Περιγραφή εξαιρέσεων ή λανθασμένων καταστάσεων (π.χ. αντίδραση του συστήματος σε μη έγκυρη είσοδο)

e. Μετά-συνθήκες

- Συνθήκες που θα ισχύουν μετά την ομαλή εκτέλεση της εν λόγω περίπτωσης χρήσης.

18

Τεκμηρίωση Περιπτώσεων Χρήσης - Παράδειγμα



19

Τεκμηρίωση Περιπτώσεων Χρήσης - Παράδειγμα

a. Σύνοψη περιγραφή

• Η περίπτωση χρήσης επιτρέπει σε έναν *Πελάτη* να δώσει μία παραγγελία αγοράς υπολογιστή. Η περίπτωση χρήσης περιλαμβάνει και την εισαγωγή διεύθυνσης αποστολής και λεπτομέρειες πληρωμής

b. Προ-συνθήκες

• Ο *Πελάτης* επιλέγει μέσω ενός Internet browser τη σελίδα όπου ο κατασκευαστής επιτρέπει την αίτηση παραγγελίας. Η σελίδα παρουσιάζει τα χαρακτηριστικά του υπολογιστή και την τιμή του

c. Κύρια ροή γεγονότων

Η περίπτωση χρήσης ξεκινά όταν ο *Πελάτης* αποφασίσει να παραγγείλει την έτοιμη σύνθεση επιλέγοντας τη λειτουργία *Συνέχεια* (ή με παρόμοιο όνομα) όταν οι λεπτομέρειες της παραγγελίας φαίνονται στην οθόνη.

Το σύστημα ζητά από τον πελάτη να εισάγει το ονοματεπώνυμο του, τη διεύθυνση του, τον τρόπο πληρωμής (μετρητά ή κάρτα) και οποιαδήποτε σχόλια. Ο *Πελάτης* επιλέγει τη λειτουργία *Αγορά* για την αποστολή της παραγγελίας στον κατασκευαστή.

Το σύστημα αναθέτει έναν μοναδικό αριθμό στην παραγγελία και αποθηκεύει τον αριθμό στη βάση δεδομένων. Ο αριθμός αποστέλλεται και με email στον πελάτη μαζί με την επιβεβαίωση της παραγγελίας

Τεκμηρίωση Περιπτώσεων Χρήσης - Παράδειγμα

d. Εναλλακτική ροή γεγονότων

• Ο χρήστης επιλέγει τη λειτουργία *Αγορά* πριν την εισαγωγή των απαραίτητων στοιχείων. Το σύστημα εκτυπώνει ένα μήνυμα σφάλματος και ζητά την πληροφορία που λείπει.

• Ο χρήστης επιλέγει τη λειτουργία *Επανεκκίνηση* για να επιστρέψει σε άδεια φόρμα εισαγωγής στοιχείων. Το σύστημα επιτρέπει στον πελάτη να ξαναεισάγει στοιχεία.

e. Μετά-συνθήκες

• Αν η περίπτωση χρήσης είναι επιτυχής, η παραγγελία καταχωρείται στη βάση δεδομένων. Αλλιώς, η κατάσταση του συστήματος παραμένει αναλλοίωτη.

21

Στατική Αποψη

• Η στατική άποψη ενός μοντέλου είναι θεμελιώδης στη UML καθώς αποτυπώνει την αρχιτεκτονική του συστήματος (μονάδες + μεταξύ τους σχέσεις)

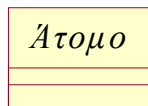
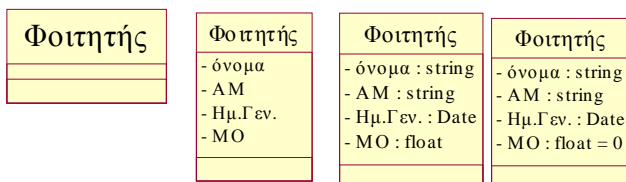
• Σε ένα αντικειμενοστρεφές σύστημα τα δομικά του στοιχεία είναι οι κλάσεις και οι σχέσεις μεταξύ των κλάσεων επιτρέπουν τη συνεργασία των αντικειμένων τους

• Τα *διαγράμματα κλάσεων* αποτυπώνουν τη στατική δομή

• Πολύ συχνά, τα διαγράμματα κλάσεων είναι το μόνο είδος διαγραμμάτων που χρησιμοποιείται λόγω των πληροφοριών που παρέχει σχετικά με τον κώδικα

22

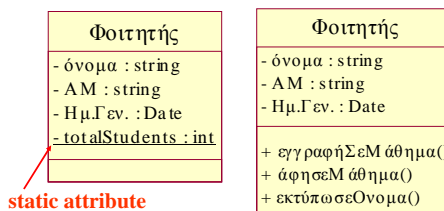
Διαγράμματα Κλάσεων



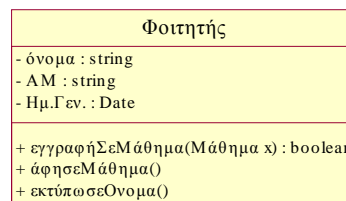
αφηρημένη (abstract) κλάση

23

Διαγράμματα Κλάσεων



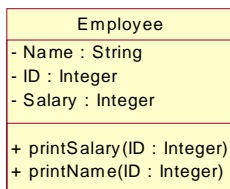
static attribute



Συχνά δεν είναι πρακτικό να αναριστώνται όλες οι ιδιότητες και λειτουργίες. Λειτουργίες *get()* και *set()* δεν φαίνονται συνήθως σε **διαγράμματα κλάσεων**

24

Στατική Άποψη



Οι ιδιότητες των κλάσεων προσδιορίζονται με τον ακόλουθο συμβολισμό:

ορατότητα όνομα : τύπος = προκαθορισμένη_τιμή

25

Ορατότητα

Η **ορατότητα** μπορεί να λάβει τις τιμές:
δημόσια (*public*) συμβολίζεται με (+),
προστατευμένη (*protected*) συμβολίζεται με (#)
ιδιωτική (*private*) συμβολίζεται με (-).

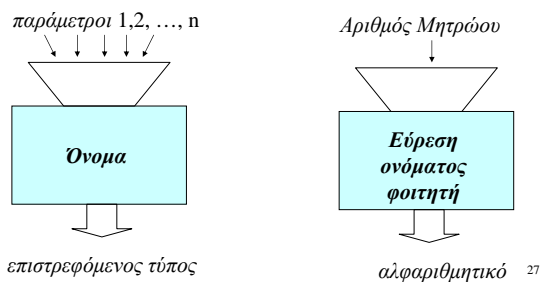
- μία ιδιότητα με **δημόσια** ορατότητα είναι απευθείας προσπελάσιμη από άλλες κλάσεις (και αντικείμενα)
- μία ιδιότητα με **ιδιωτική** ορατότητα είναι προσπελάσιμη μόνο από την ίδια την κλάση
- μία ιδιότητα με **προστατευμένη** ορατότητα είναι προσπελάσιμη μόνο από την ίδια την κλάση και τις υποκείμενες κλάσεις της.

26

Λειτουργίες

Οι λειτουργίες των κλάσεων προσδιορίζονται με τον ακόλουθο συμβολισμό:

ορατότητα όνομα (λίστα παραμέτρων) : επιστρεφόμενος_τύπος



27

Σχέσεις

Σχέση	Λειτουργία	Συμβολισμός
Συσχέτιση	Περιγραφή μιας διασύνδεσης μεταξύ αντικείμενων	_____
Εξάρτηση	Μία σχέση εξάρτησης μεταξύ δύο στοιχείων ενός μοντέλου	----->
Ροή	Μία συσχέτιση μεταξύ δύο εκδόσεων ενός αντικείμενου σε διαδοχικές χρονικές στιγμές	----->
Γενίκευση	Μία συσχέτιση μεταξύ μιας γενικής περιγραφής και μιας ειδικότερης περίπτωσης - υποδηλώνει κληρονομικότητα	_____>
Υλοποίηση	Σχέση μεταξύ μιας προδιαγραφής και της υλοποίησής της	----->
Χρήση	Μία κατάσταση κατά την οποία ένα στοιχείο του μοντέλου απαιτεί την ύπαρξη κάποιου άλλου για την ορθή λειτουργία του	----->

Σύνθεση -
Συσσωμάτωση

Σχέση περιεκτικότητας
μεταξύ στοιχείων

28

Συσχέτιση (Association)

Οι συσχετίσεις παρέχουν τις διασυνδέσεις βάσει των οποίων τα αντικείμενα διαφόρων κλάσεων μπορούν να αλληλεπιδρούν και να ανταλλάσσουν μηνύματα.

Οι συσχετίσεις είναι η "κόλλα" του συστήματος: χωρίς συσχετίσεις υπάρχουν μόνο ανεξάρτητες κλάσεις που δεν συνεργάζονται.

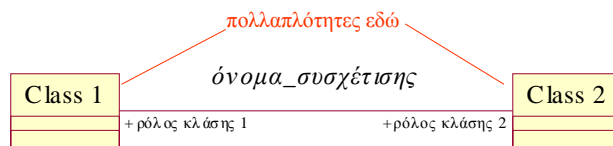
Κάθε συσχέτιση έχει δύο άκρα τα οποία είναι δυνατόν να ονομαστούν ώστε να προσδιορίζουν ρόλους.

Η σημαντικότερη ιδιότητα των άκρων είναι η **πολλαπλότητα** (multiplicity) που υποδηλώνει τον αριθμό των στιγμιότυπων μιας κλάσης που μπορούν να συσχετιστούν με ένα στιγμιότυπο μιας άλλης κλάσης.

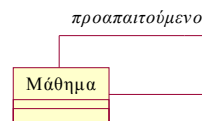
Μία συσχέτιση συμβολίζεται με μία συνεχή γραμμή που συνδέει τις κλάσεις που συμμετέχουν.

29

Συσχέτιση (Association)



Αυτοσυσχέτιση: Σχέση μεταξύ διαφορετικών αντικειμένων της ίδιας κλάσης



30

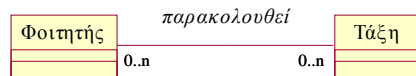
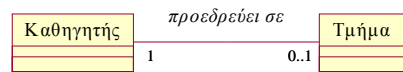
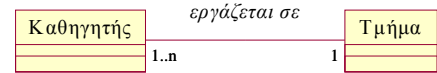
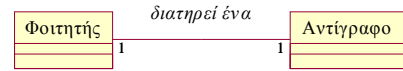
Πολλαπλότητα

*	Οποιοσδήποτε αριθμός αντικειμένων
1	Ακριβώς ένα αντικείμενο
<i>n</i>	Ακριβώς <i>n</i> αντικείμενα (όπου <i>n</i> ακέραιος)
0..1	Μηδέν ή ένα αντικείμενα (υποδηλώνει ότι η συσχέτιση είναι προαιρετική)
<i>n..m</i>	Εύρος τιμών με μικρότερη δυνατή τιμή <i>n</i> και μέγιστη τιμή <i>m</i>
<i>n, m</i>	Διακριτοί συνδυασμοί (π.χ. ένα αυτοκίνητο μπορεί να έχει 2 ή 4 πόρτες)

31

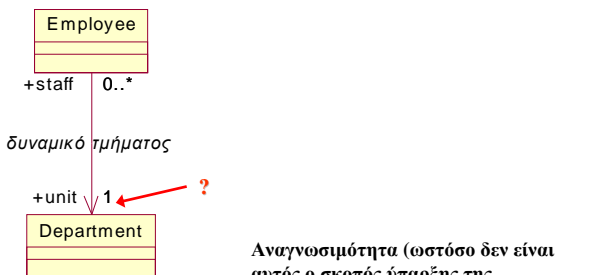
UML – Παραδείγματα Πολλαπλότητας

Ένας Φοιτητής διατηρεί ακριβώς ένα Αντίγραφο, ενώ ένα Αντίγραφο ανήκει σε έναν ακριβώς Φοιτητή

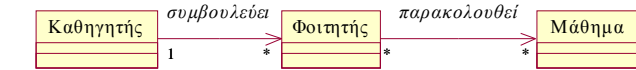


32

Δυνατότητα Πλοήγησης



Αναγνωσιμότητα (ωστόσο δεν είναι αυτός ο σκοπός ύπαρξης της δυνατότητας πλοήγησης !!)



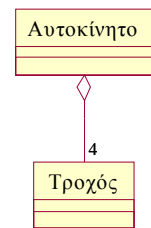
"Ένας καθηγητής συμβουλεύει πολλούς φοιτητές" και "Οι φοιτητές παρακολουθούν πολλά μαθήματα".

33

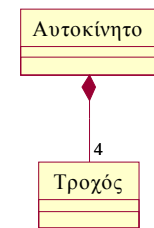
Συσσωμάτωση - Σύνθεση

Σχέσεις Συνόλου-Τμήματος (Whole – part)

Συσσωμάτωση (Aggregation)



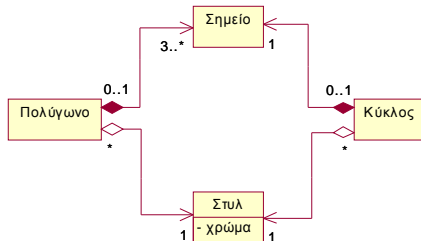
Σύνθεση (Composition)



συνήθως η συσσωμάτωση - σύνθεση δεν ονομάζεται καθώς πάντοτε σημαίνει "περιέχει"

34

Συσσωμάτωση - Σύνθεση



• Οι συνθέσεις στην κλάση "Σημείο" υποδηλώνουν ότι το σημείο αποτελεί τμήμα ενός πολυγώνου είτε ενός κύκλου, αλλά καθώς η διαχείριση του πραγματοποιείται από αυτές τις κλάσεις, ένα σημείο δεν μπορεί να ανήκει ταυτόχρονα και σε ένα Πολύγωνο και σε ένα Κύκλο.

• Η κλάση "Στυλ" ωστόσο, συσσωμάτωση και κατά συνέπεια ένα στυλ μπορεί να μοιράζεται μεταξύ πολλών πολυγώνων και πολλών κύκλων.

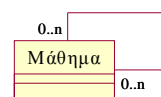
• Επιπλέον, η διαγραφή ενός Πολυγώνου συνεπάγεται τη διαγραφή των συσχετισμένων με αυτό Σημείων αλλά όχι τη διαγραφή του συσχετισμένου με αυτό Στυλ.

35

UML – Παραδείγματα Πολλαπλότητας

Πολλαπλότητα μπορεί να υπάρξει και σε αυτοσυσχετίσεις

προαπαιτούμενο για



και σε συσσωματώσεις - συνθέσεις

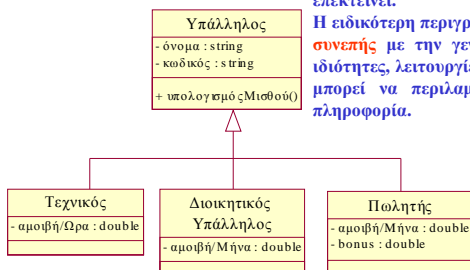


36

Γενίκευση

Η σχέση γενίκευσης αφορά μία σχέση μεταξύ μιας γενικής περιγραφής και μιας ειδικότερης περιγραφής που την επεκτείνει.

Η ειδικότερη περιγραφή είναι **απολύτως συνεπής** με την γενική (έχει τις ίδιες ιδιότητες, λειτουργίες και σχέσεις) αλλά μπορεί να περιλαμβάνει επιπρόσθετη πληροφορία.



37

Γενίκευση

Ο πρώτος σκοπός της γενίκευσης είναι η **δυνατότητα υποκατάστασης** (substitutability principle) που αναφέρεται στη δυνατότητα χρήσης μιας υποκειμένης κλάσης όταν έχει δηλωθεί μία υπερκείμενη κλάση.

Για παράδειγμα, αν μία μεταβλητή τύπου "Επιχείρηση" έχει δηλωθεί ότι περιλαμβάνει "Υπάλληλους", τότε ένα αντικείμενο "Τεχνικός" θα πρέπει να είναι επιτρεπτή τιμή για έναν υπάλληλο.

Ο δεύτερος σκοπός της γενίκευσης είναι η σταδιακή περιγραφή ενός στοιχείου, αξιοποιώντας την περιγραφή των προγόνων του, αρχή που συναντάται ως **κληρονομικότητα**. Με τον μηχανισμό της κληρονομικότητας, η περιγραφή (ιδιότητες και λειτουργίες) μιας κλάσης, κατασκευάζεται τόσο από την περιγραφή της ίδιας της κλάσης όσο και από τις περιγραφές των υπερκείμενων κλάσεων.

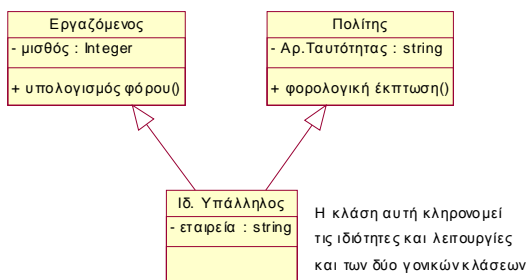
Παρόλο που **μία λειτουργία μπορεί να δηλωθεί σε περισσότερες από μία κλάσεις** που συνδέονται με κληρονομικότητα, **μία ιδιότητα δεν θα πρέπει να δηλωθεί περισσότερες από μία φορές**.

Η υλοποίηση μιας λειτουργίας (μέθοδος) σε μία υποκειμένη κλάση επικαλύπτει οποιαδήποτε μέθοδο με το ίδιο όνομα σε υπερκείμενες κλάσεις.

38

Πολλαπλή Κληρονομικότητα

Αν μία κλάση έχει πολλαπλές γονικές κλάσεις, κληρονομεί την περιγραφή κάθε μίας από αυτές. Οι ιδιότητες και οι λειτουργίες είναι η ένωση των ιδιοτήτων και λειτουργιών των γονικών κλάσεων.



39

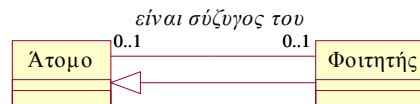
UML – Παραδείγματα Πολλαπλότητας

Δεν είναι λογικό να υπάρχει πολλαπλότητα σε σχέσεις κληρονομικότητας, καθώς η κληρονομικότητα **αφορά κλάσεις και όχι αντικείμενα**.



Υποδηλώνει ότι **κάθε αντικείμενο** της υποκειμένης κλάσης είναι ταυτοχρόνος και αντικείμενο της γονικής κλάσης

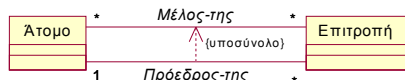
Αν θέλουμε να παραστήσουμε κάποιο είδος σχέση μεταξύ αντικειμένων των δύο κλάσεων, μπορούμε να εισάγουμε μία ξεχωριστή συσχέτιση



40

Περιορισμοί (Constraints)

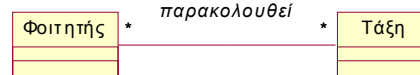
- Ένας περιορισμός είναι μία **λογική συνθήκη** (έκφραση Boole) που πρέπει να είναι αληθής για να λάβει χώρα μία ενέργεια ή μία συσχέτιση.
- Η UML επιτρέπει τον καθορισμό περιορισμών με οποιοδήποτε τρόπο (αρκεί η περιγραφή να βρίσκεται μέσα σε άγκιστρα { }).
- Ωστόσο, η UML περιλαμβάνει τον ορισμό μιας τυπικής γλώσσας περιορισμών (**Object Constraint Language - OCL**).



41

Κλάσεις Συσχετίσεων

- Συχνά μία ιδιότητα αναγνωρίζεται ως σημαντική για το μοντέλο, ωστόσο δεν εντοπίζεται ότι ανήκει με ιδανικό τρόπο σε κάποια κλάση. Έστω για παράδειγμα η περίπτωση μιας συσχέτισης μεταξύ ενός *Φοιτητή* και μιας *Τάξης* για ένα μάθημα που παρακολουθεί:



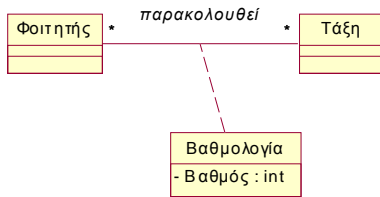
Στο τέλος κάθε εξαμήνου, ο φοιτητής λαμβάνει τη βαθμολογία για το μάθημα του οποίου την τάξη παρακολούθησε. **Ωστόσο, που ανήκει η ιδιότητα Βαθμός ;**

- Δεν πρόκειται για ιδιότητα της κλάσης *Φοιτητής* διότι ένας φοιτητής δεν λαμβάνει ένα συγκεντρωτικό βαθμό για όλες τις τάξεις που παρακολούθησε αλλά ένα διαφορετικό βαθμό για κάθε τάξη.

- Δεν πρόκειται για ιδιότητα της κλάσης *Τάξη* διότι οι φοιτητές που παρακολουθούν μία τάξη δεν λαμβάνουν τον ίδιο βαθμό.

Κλάσεις Συσχετίσεων

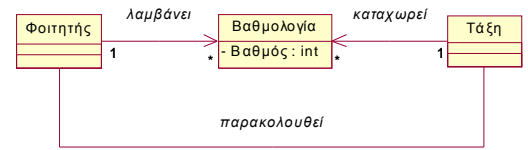
- Εξετάζοντας μία τέτοια κατάσταση, καταλήγουμε στο ότι ο βαθμός είναι στην πραγματικότητα **μία ιδιότητα της συσχέτισης** μεταξύ ενός αντικειμένου *Φοιτητής* και ενός αντικειμένου *Τάξη*.
- Στη UML, για τη φιλοξενία ιδιοτήτων που ανήκουν στη συσχέτιση δημιουργούμε μία ξεχωριστή κλάση, γνωστή ως **κλάση συσχέτισης** (association class), η οποία συμβολίζεται ως εξής:



43

Κλάσεις Συσχετίσεων

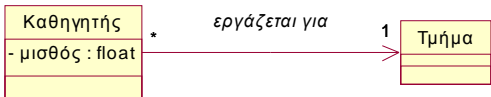
- Ωστόσο, κάθε φορά που εντοπίζεται μία κλάση συσχέτισης σε ένα διάγραμμα κλάσεων, θα πρέπει να είναι γνωστό ότι υπάρχει **εναλλακτική δυνατότητα αναπαράστασης** της ίδιας κατάστασης **χωρίς τη χρήση κλάσης συσχέτισης**.
- Στην περίπτωση μιας συσχέτισης *πολλά-προς-πολλά* στην οποία εμπλέκεται και μία κλάση συσχέτισης, η συσχέτιση **μπορεί να διασπαστεί σε δύο ένα-προς-πολλά συσχετίσεις**, εισάγοντας την κλάση που προηγουμένως ήταν η κλάση συσχέτισης ως 'κανονική' κλάση μεταξύ των δύο άλλων.



44

Κλάσεις Συσχετίσεων

- Σε περίπτωση που μία κλάση συσχέτισης υπάρχει σε μία συσχέτιση *ένα-προς-πολλά* όπως στο ακόλουθο παράδειγμα



45

Πλεονάζουσα Πληροφορία

- Η ύπαρξη μιας συσχέτισης υπονοεί τη χρήση ιδιοτήτων ως "δείκτες" σε όλες τις συσχετιζόμενες κλάσεις:

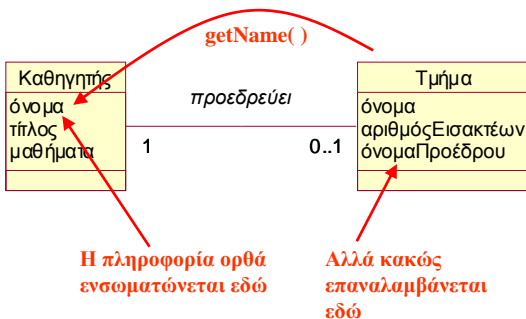


Κίνδυνος διπλής παραγωγής ιδιοτήτων στον κώδικα

46

Πλεονάζουσα Πληροφορία 2

- Συχνά εισάγεται πλεονάζουσα πληροφορία υπό μορφή ιδιοτήτων:



47

Ροή της Πληροφορίας

- Στην πράξη, κάθε γραμμή συσχέτισης σε ένα διάγραμμα, θα πρέπει να θεωρείται ως ένας **νοητικός 'αγωγός'** δια μέσου του οποίου μπορεί να ρέει η πληροφορία μεταξύ αντικειμένων.

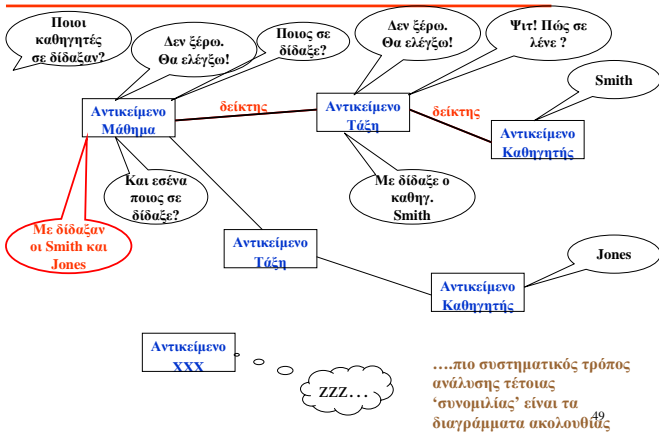
Στη φάση της ανάλυσης, δεν θα πρέπει να ανησυχούμε για την ορατότητα κάθε ιδιότητας; Υποθέτουμε ότι όλες οι τιμές των ιδιοτήτων είναι διαθέσιμες καλώντας τις κατάλληλες *get* μεθόδους

Έστω ότι κάποιος χρήστης επιθυμεί να αποκτήσει μία λίστα όλων των καθηγητών που έχουν διδάξει το μάθημα 'Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό'.

Επειδή κάθε αντικείμενο τύπου *Μάθημα* διατηρεί δείκτες προς όλες τις *Τάξεις*, τορινές ή παρελθοντικές, το αντικείμενο *Μάθημα* μπορεί να 'ρωτήσει' τις συσχετιζόμενες *Τάξεις* το όνομα του *Καθηγητή* που δίδαξε ή διδάσκει την αντίστοιχη *Τάξη*.

48

Ροή της Πληροφορίας



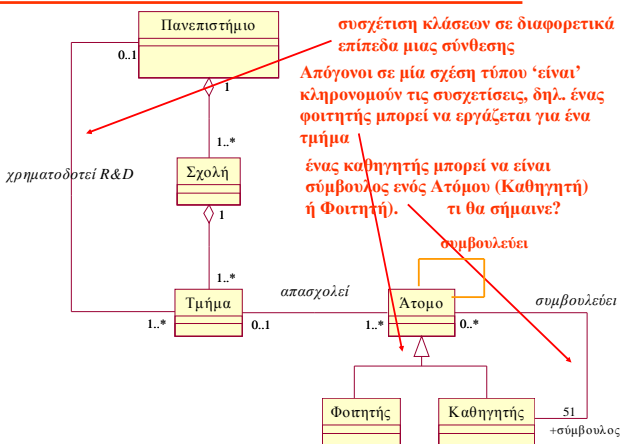
Ροή της Πληροφορίας

Σε περίπτωση που από την ανάλυση προέκυπτε ότι συχνά απαιτούνταν να γνωρίζουμε ποιο καθηγητές δίδαξαν ένα μάθημα, θα μπορούσαμε να επιλέξουμε να προσθέσουμε μια πλεονάζουσα συσχέτιση μεταξύ Καθηγητή και Μαθήματος



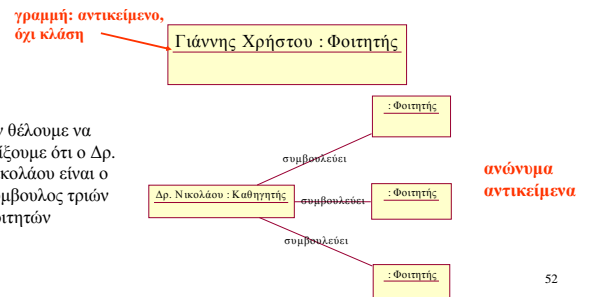
50

Μίξη Συμβολισμών



UML – Διαγράμματα Αντικειμένων

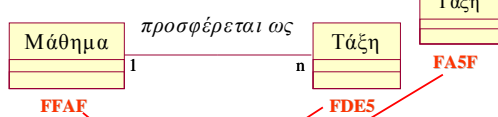
Συχνά, για την επεξήγηση του τρόπου με τον οποίο τα αντικείμενα αλληλεπιδρούν στα πλαίσια ενός σεναρίου, απαιτείται να παρασταθούν τα αντικείμενα και οι δεσμοί μεταξύ τους. Για το λόγο αυτό δημιουργούνται διαγράμματα αντικειμένων. Ένα διάγραμμα αντικειμένων είναι ένα χρονικό στιγμιότυπο μιας κατά τα άλλα δυναμικής δομής



52

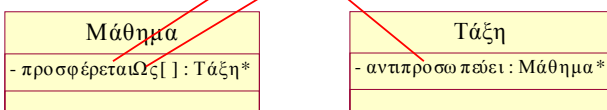
Συσχετίσεις ως Ιδιότητες

Έστω το ακόλουθο διάγραμμα της συσχέτισης 'ένα Μάθημα προσφέρεται ως Τάξη'



Τι σημαίνει ότι δύο αντικείμενα συσχετίζονται ?

Κάθε αντικείμενο διατηρεί έναν δείκτη προς το άλλο, ώστε να μπορούν να επικοινωνούν και να συνεργάζονται



Μοντελοποίηση της Δυναμικής Συμπεριφοράς

Μέχρι στιγμής μελετήθηκε η στατική δομή του συστήματος

Στοιχεία του διαγράμματος κλάσεων:

- Κλάσεις
- Συσχετίσεις
- Ιδιότητες
- Ιεραρχικές σχέσεις (κληρονομικότητα)
- Λειτουργίες/Μέθοδοι ? ? ?

Καθορισμός των μεθόδων συνιστά μοντελοποίηση της δυναμικής συμπεριφοράς του συστήματος

Κατά την ανάπτυξη του δυναμικού μοντέλου ενδέχεται να προκύψουν βελτιώσεις της στατικής δομής

54

Άποψη Αλληλεπίδρασης

- Σε ένα αντικειμενοστραφές σύστημα, η συμπεριφορά υλοποιείται από την αλληλεπίδραση αντικειμένων. Αυτή η αλληλεπίδραση περιγράφεται στα πλαίσια ενός μοντέλου της UML με δύο είδη συμπληρωματικών διαγραμμάτων, τα **διαγράμματα ακολουθίας** και τα **διαγράμματα συνεργασίας**.
- Τα διαγράμματα αυτά, και ιδιαίτερα τα διαγράμματα ακολουθίας δεν χρησιμοποιούνται μόνο για την τεκμηρίωση του συστήματος, αλλά και για τη **διερεύνηση διαφόρων σεναρίων χρήσης** του συστήματος με σκοπό τον εντοπισμό των λειτουργιών κάθε αντικειμένου
- Τα διαγράμματα ακολουθίας (και συνεργασίας) είναι κατάλληλα για την απεικόνιση της στατικής συμπεριφοράς των αντικειμένων. Ως **στατική συμπεριφορά ενός αντικειμένου** νοείται το σύνολο των λειτουργιών που δεν εξαρτώνται από εσωτερικά ή εξωτερικά γεγονότα. (Η λειτουργία υπολογισμού της τετραγωνικής ρίζας ενός αριθμού αποτελεί περίπτωση στατικής συμπεριφοράς)

Άποψη Αλληλεπίδρασης

- Η στατική άποψη περιγράφει τις εγγενείς ιδιότητες μιας κλάσης. Η **περιγραφή συμπεριφοράς συνίσταται στο σύνολο των μηνυμάτων που ανταλλάσσονται** από τα αντικείμενα που αντιστοιχούν στους ρόλους. Ένα τέτοιο σύνολο μηνυμάτων μιας συνεργασίας ονομάζεται αλληλεπίδραση (interaction).
- Ένα μήνυμα είναι μία μονόδρομη επικοινωνία μεταξύ δύο αντικειμένων, μία ροή ελέγχου με πληροφορία από έναν αποστολέα προς έναν αποδέκτη. Ένα μήνυμα μπορεί να έχει παραμέτρους μεταφέροντας τιμές μεταξύ των αντικειμένων.
- Η ακολουθία των μηνυμάτων παρουσιάζεται είτε με διαγράμματα ακολουθίας που **εστιάζουν στην χρονική ακολουθία** των μηνυμάτων είτε με διαγράμματα συνεργασίας που **εστιάζουν στις σχέσεις μεταξύ των αντικειμένων** που ανταλλάσσουν μηνύματα

56

Διαγράμματα Ακολουθίας (Sequence Diagrams) ***

- Ένα διάγραμμα ακολουθίας παρουσιάζει την αλληλεπίδραση μεταξύ αντικειμένων σε δύο διαστάσεις, όπου:
 - η **κάθετη διάσταση** αντιστοιχεί στην **κλίμακα του χρόνου**
 - η **οριζόντια διάσταση** στα **ανεξάρτητα αντικείμενα**.
- Σε κάθε αντικείμενο αντιστοιχεί μία κάθετη γραμμή που ονομάζεται **γραμμή ζωής** (lifeline). Για όσο χρόνο ένα αντικείμενο υφίσταται, η γραμμή αυτή είναι διακεκομμένη ενώ για όσο χρόνο μία διαδικασία του εν λόγω αντικειμένου είναι ενεργή η γραμμή ζωής σχεδιάζεται ως μία **διπλή γραμμή**.
- Ένα μήνυμα συμβολίζεται ως μία **ακμή** από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Η θέση των ακμών αντιστοιχεί στην τοποθέτησή τους σε σχέση με τον χρόνο (μία ακμή χαμηλότερα από μία άλλη αντιστοιχεί σε μεταγενέστερο χρόνο).
- Απαντήσεις σε μηνύματα υποδηλώνονται ως οριζόντιες διακεκομμένες ακμές

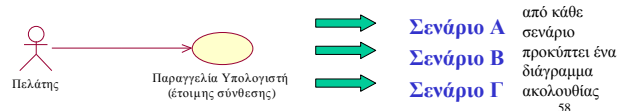
Σενάρια

Τα γεγονότα που ξεκινούν έξω από το σύστημα, ενεργοποιούνται σε τυχαίες χρονικές στιγμές και κατά συνέπεια δεν μοντελοποιούνται

Τα **εσωτερικά γεγονότα** όμως που ενεργοποιούνται ως απόκριση στα εξωτερικά, δεν πρέπει να εμφανίζονται τυχαία.

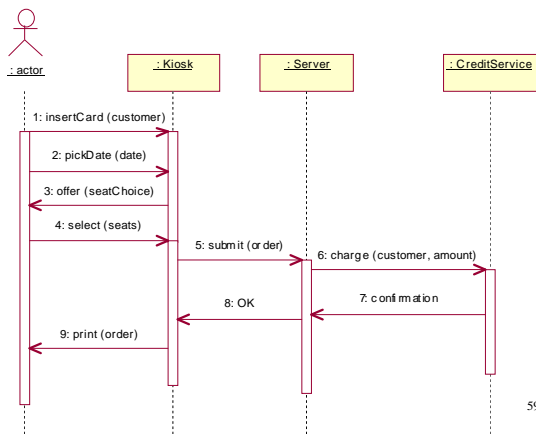
Απαιτείται επομένως η οργάνωση και μοντελοποίησή τους κατά τρόπο αιτίου-αιτιατού, ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα

Ένα **σενάριο** καταγράφει την **αλληλουχία των εσωτερικών μηνυμάτων** που πρέπει να λάβει χώρα, ώστε να ολοκληρωθεί (από την αρχή μέχρι το τέλος) μία λειτουργία του συστήματος

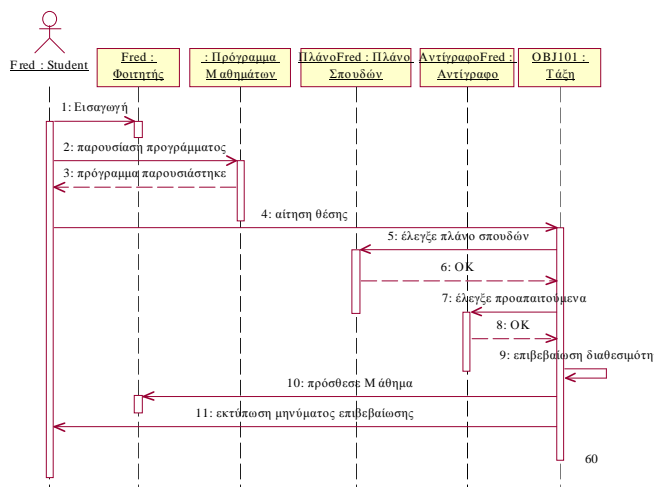


58

Διαγράμματα Ακολουθίας (Sequence Diagrams) ***



59



60

Διαγράμματα Ακολουθίας

- Τα διαγράμματα ακολουθίας είναι εξαιρετικά χρήσιμα για τον εντοπισμό λειτουργιών στις διάφορες κλάσεις.
- Αν ένα αντικείμενο λαμβάνει ένα μήνυμα *alpha*, **συνεπάγεται ότι η κλάση στην οποία ανήκει το εν λόγω αντικείμενο θα πρέπει να έχει μία μέθοδο με το ίδιο όνομα για να εξυπηρετεί τον αποστολέα. (Joe kicked the ball !)**
- Υπονοείται επίσης, ότι αφού δύο αντικείμενα επικοινωνούν μεταξύ τους μέσω μηνυμάτων, θα πρέπει να υπάρχει κάποιου είδους συσχέτιση μεταξύ τους που θα λειτουργεί ως διάλογος επικοινωνίας.
- Για παράδειγμα, αν ένα αντικείμενο A στέλνει ένα μήνυμα ζητώντας κάποια λειτουργία από ένα αντικείμενο B, το A θα πρέπει με κάποιο τρόπο να γνωρίζει που να αποστείλει το μήνυμα. Αν αυτό συνεπάγεται την ύπαρξη μιας μονόδρομης συσχέτισης μεταξύ των αντίστοιχων κλάσεων, θα πρέπει κατά την υλοποίηση του συστήματος, στο συγκεκριμένο αντικείμενο A να γνωστοποιηθεί η θέση του αντικειμένου B

Γεγονότα

- Η αλληλεπίδραση μεταξύ αντικειμένων **ενεργοποιείται από γεγονότα**
- **Γεγονός** είναι οποιοδήποτε εξωτερικό ερέθισμα σε ένα αντικείμενο, που σηματοδοτείται ως λήψη ενός **μηνύματος**. Ένα γεγονός μπορεί να ενεργοποιείται από:
 - εξωτερικούς χρήστες (click στο mouse)
 - άλλα υπολογιστικά συστήματα (λήψη ενός email)
 - άλλο αντικείμενο στο ίδιο σύστημα (αλυσιδωτή ενεργοπ.)
 - την πάροδο του χρόνου

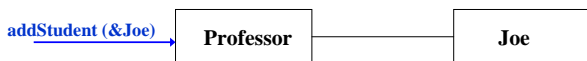
62

Γεγονότα

Όταν ένα αντικείμενο ειδοποιηθεί για ένα γεγονός μέσω μηνύματος:

- **μπορεί να αλλάξει την κατάστασή του**

```
class Professor {
public:
    addStudent(Student* s)
    { πρόσθεσε το s στο supervisedStudents; }
private:
    std::list<Student*> supervisedStudents;
};
```



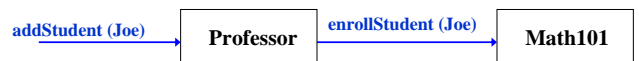
63

Γεγονότα

Όταν ένα αντικείμενο ειδοποιηθεί για ένα γεγονός μέσω μηνύματος:

- **μπορεί να στείλει μήνυμα (γεγονός) σε άλλο αντικείμενο**

```
class Section {
public:
    addStudent(Student* s)
    { στείλε μήνυμα στο course; }
private:
    Course* course;
};
```



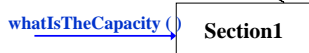
Κλασικό παράδειγμα delegation (κατανόησης αρμοδιοτήτων) όπου ένα αντικείμενο υποστηρίζει μία λειτουργία άλλου αντικειμένου⁶⁴

Γεγονότα

Όταν ένα αντικείμενο ειδοποιηθεί για ένα γεγονός μέσω μηνύματος:

- **μπορεί να επιστρέψει μία τιμή**

```
class Section {
public:
    whatIsTheCapacity()
    { επιστροφή τιμής capacity; }
private:
    Course* course;
    int capacity;
};
```



Κλασικό παράδειγμα get συνάρτησεων

65

Γεγονότα

Όταν ένα αντικείμενο ειδοποιηθεί για ένα γεγονός μέσω μηνύματος:

- **μπορεί να αγνοήσει το γεγονός**

Έστω ότι ο Καθηγητής λαμβάνει το μήνυμα “Προσθήκη ενός Φοιτητή στους Επιβλεπόμενους”

Αν ο φοιτητής είναι ήδη εγγεγραμμένος στη λίστα, το αντικείμενο Καθηγητής δεν θα κάνει τίποτα – δηλαδή δεν θα αλλάξει κατάσταση, ούτε θα αποστείλει νέο μήνυμα (απλοποίηση)

66

Διαγράμματα Ακολουθίας

- Κατά τη μελέτη σεναρίων (ανάπτυξη διαγραμμάτων ακολουθίας) είναι πιθανό να προκύψουν νέες κλάσεις, ιδιότητες και σχέσεις και συνεπώς να τροποποιηθεί το στατικό μοντέλο. Κάτι τέτοιο συμβαίνει συχνά και είναι επιθυμητό
- Αν για παράδειγμα από τη μελέτη διαφόρων διαγραμμάτων ακολουθίας προκύπτει ότι δύο αντικείμενα αποκρίνονται με τον ίδιο τρόπο στα διάφορα δυνατά μηνύματα, τότε τα δύο αυτά αντικείμενα είτε θα ανήκουν στην ίδια κλάση είτε θα ανήκουν σε κλάσεις που συνδέονται με σχέση κληρονομικότητας.
- Υπό αυτή την έννοια, η επαναληπτική εξέταση διαγραμμάτων ακολουθίας και διαγραμμάτων κλάσεων οδηγεί στην ανάπτυξη του τελικού συστήματος, πολύ πριν αρχίσει η φάση της κωδικοποίησης. Θα πρέπει να γίνει κατανοητό, ότι αυτή η επαναληπτική διαδικασία στη φάση της ανάλυσης βρίσκεται στο επίκεντρο της αντικειμενοστραφούς μεθοδολογίας για την ανάπτυξη συστημάτων

Χρήση Διαγρ. Ακολουθίας για τον καθορισμό Μεθόδων

-Όλες οι ακμές που αντιπροσωπεύουν απόκριση από μία λειτουργία που εκτέλεσε κάποιο άλλο αντικείμενο, δεν μοντελοποιούνται ως μέθοδοι

- Αυτό-κλήσεις υποδηλώνουν λειτουργίες που το αντικείμενο πραγματοποιεί στον εαυτό του: είτε ιδιωτικές μεθόδους είτε δημόσια διαθέσιμες μεθόδους που μπορούν να χρησιμοποιήσουν και άλλα αντικείμενα.

Η εξέταση όλων των διαγραμμάτων ακολουθίας (σεναρίων) θα αποκαλύψει τις περισσότερες λειτουργίες/μεθόδους του συστήματος.

Ωστόσο, κάποιες από τις μεθόδους δεν θα έρθουν στην επιφάνεια μέχρι την υλοποίηση του συστήματος. Αυτό είναι αναμενόμενο.

UML – Διαγραμματικά Στοιχεία

Στοιχείο	Περιγραφή	Συμβολισμός
Αντικείμενο	Ένα αντικείμενο αναπαριστάται με ένα ορθόγιο στο οποίο αναγράφεται το όνομα του αντικείμενου στη συνάρτηση μία άνω κάτω τελεία και ακολουθεί το όνομα της κλάσης στην οποία ανήκει.	
Πλαίσιο Ενεργοποίησης	Κάτω από κάθε αντικείμενο εκτείνεται μια διακεκομμένη γραμμή που αντιστοιχεί στη γραμμή ζωής του αντικείμενου. Όταν λαμβάνεται ένα μήνυμα ξεκινά ένα πλαίσιο ενεργοποίησης (διπλή γραμμή) το οποίο αντιστοιχεί στη διάρκεια εκτέλεσης της λειτουργίας εξυπηρέτησης του μηνύματος.	
Αποστολή Μηνύματος	Μία προσανατολισμένη ακμή από το αντικείμενο που αποστέλλει το μήνυμα προς το αντικείμενο που είναι αποδέκτης. Υλοποιείται με μία λειτουργία στην κλάση του αντικείμενου που είναι αποδέκτης του μηνύματος. Στο παράδειγμα η τμήση Department θα πρέπει να έχει μία μέθοδο με όνομα getCustomer(). Αν η αποστολή του μηνύματος γίνεται από συνάρτηση, η συνθήκη μπαίνει σε αγκύρα πριν το όνομα του μηνύματος. Αν το μήνυμα αποστέλλεται επαναληπτικά, συμβολίζεται με ένα αστερίδιο πριν το όνομα του μηνύματος.	

UML – Διαγραμματικά Στοιχεία

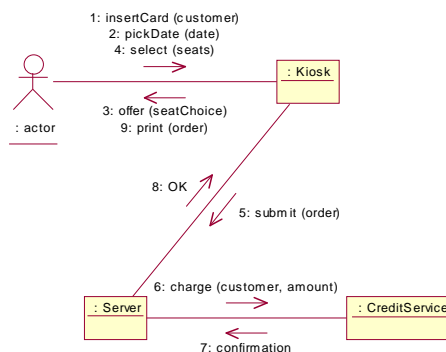
Επιστροφή Μηνύματος	Μία διακεκομμένη ακμή που επιστρέφει από το κληθέν αντικείμενο σε αυτό που πραγματοποίησε την κλήση υποδηλώνοντας ολοκλήρωση της λειτουργίας εξυπηρέτησης ενός μηνύματος. (Προαιρετικό)	
Αυτο-κλήση	Αποστολή μηνύματος ενός αντικείμενου στον εαυτό του ή αλλιώς κλήση μιας μεθόδου του αντικείμενου από το ίδιο το αντικείμενο.	
Ασύγχρονο Μήνυμα	Ένα ασύγχρονο μήνυμα δεν εμποδίζει το αντικείμενο που πραγματοποιεί την κλήση να συνεχίσει την όποια λειτουργία του χωρίς να αναμένει αν το μήνυμα λήφθηκε ή όχι. Συνεπώς, ένα ασύγχρονο μήνυμα δημιουργεί ένα νέο νήμα εκτέλεσης ή μπορεί να δημιουργήσει ένα νέο αντικείμενο.	

Διαγράμματα Συνεργασίας (Collaboration Diagrams)

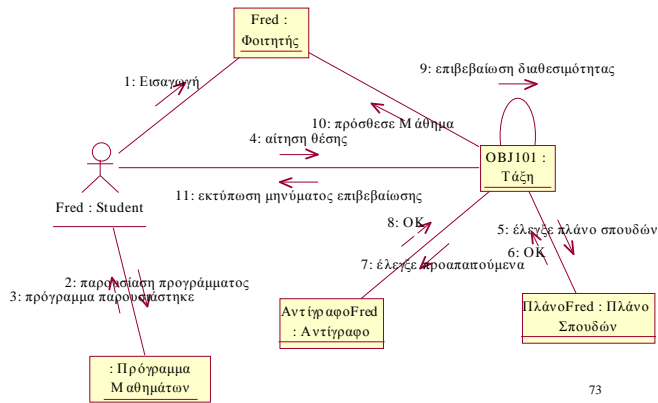
- Σε ένα διάγραμμα συνεργασίας τα αντικείμενα απεικονίζονται με τις γραμμές συσχετίσεων των κλάσεων τους να τα ενώνουν, δηλαδή απεικονίζονται οι στατικές συνδέσεις μεταξύ των αντικειμένων.
- Ενώ τα διαγράμματα ακολουθίας απεικονίζουν κυρίως τη χρονική ροή των μηνυμάτων σε ένα σενάριο μιας περίπτωσης χρήσης, τα διαγράμματα συνεργασίας χρησιμοποιούνται για να παρουσιάσουν τις σχέσεις μεταξύ αντικειμένων.
- Δεν υπάρχει συγκεκριμένη μορφή (τα αντικείμενα μπορούν να εμφανίζονται σε οποιοδήποτε σημείο του διαγράμματος) ενώ για να απεικονιστεί η ακολουθία των μηνυμάτων που ανταλλάσσονται χρησιμοποιείται αριθμηση. Τα διαγράμματα ακολουθίας και συνεργασίας θεωρούνται συμπληρωματικά καθώς περιέχουν την ίδια πληροφορία αλλά κάθε ένα δίνει μια διαφορετική οπτική γωνία (σε πολλά εργαλεία το ένα είδος διαγράμματος παράγει αυτόματα από το άλλο).

Διαγράμματα Συνεργασίας (Collaboration Diagrams)

- Τα διαγράμματα συνεργασίας που αντιστοιχούν στα προηγούμενα διαγράμματα ακολουθίας είναι τα ακόλουθα:



Διαγράμματα Συνεργασίας



73

Αποψη Ροής Ελέγχου

- Τα διαγράμματα ακολουθίας/συνεργασίας δίνουν έμφαση στη ροή της πληροφορίας. **Δεν είναι κατάλληλα για απεικόνιση της ροής του ελέγχου (control flow)**
- Η ροή του ελέγχου παρουσιάζεται με τα διαγράμματα δραστηριότητας και καταστάσεων
- Δεν συναντώνται συχνά
- Αντικαθιστούν τα παλαιότερα διαγράμματα ροής (flowcharts)

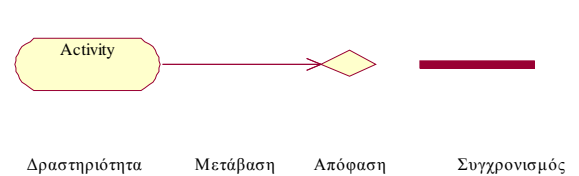
74

Διαγράμματα Δραστηριότητας (Activity Diagrams)

- Ένα **διάγραμμα δραστηριότητας** μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Bohm & Jacorini).
- Παρουσιάζεται η **ροή του ελέγχου** μεταξύ δραστηριοτήτων του ίδιου αντικειμένου ή πολλών αντικειμένων
- Μία **κατάσταση δραστηριότητας** συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες με περιγραφή της δραστηριότητας.
- Η **μετάβαση** κατά τη συμπλήρωση μιας δραστηριότητας συμβολίζεται ως ακμή.
- Οι διακλαδώσεις συμβολίζονται είτε με **συνθήκες φρουρούς** επί των μεταβάσεων είτε με **κόμβους απόφασης** (ρόμβους) με πολλαπλές εξερχόμενες ακμές.
- Μία **ένωση** (join) συμβολίζει συνένωση πολλών εισερχομένων μεταβάσεων σε μία εξερχόμενη ενώ μία **διχάλα** (fork) την ανάλυση μιας εισερχόμενης σε πολλές παράλληλες ταυτόχρονες.

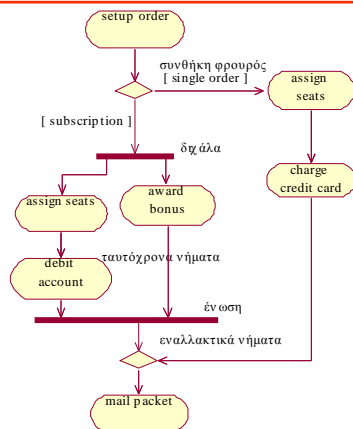
Διαγράμματα Δραστηριότητας (Activity Diagrams)

Βασικά Διαγραμματικά Στοιχεία (UML)



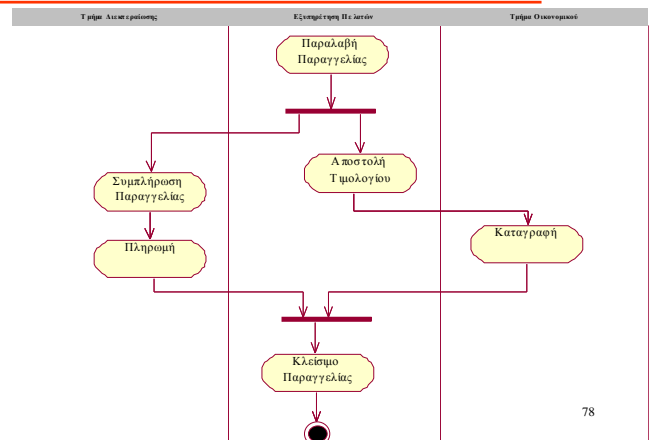
76

Διαγράμματα Δραστηριότητας (Activity Diagrams)



77

Διάδρομοι (Swimlanes)



78

Άποψη Μηχανής Καταστάσεων

- Η άποψη μηχανής καταστάσεων (state machine view) ενός μοντέλου περιγράφει τη **δυναμική συμπεριφορά των αντικειμένων** στη διάρκεια του χρόνου, αναπαριστώντας τον κύκλο ζωής των αντικειμένων μιας κλάσης.
- Κάθε αντικείμενο αντιμετωπίζεται ως ξεχωριστή οντότητα που επικοινωνεί με το περιβάλλον ανιχνεύοντας γεγονότα και αντιδρώντας σε αυτά.
- **Γεγονός είναι οτιδήποτε μπορεί να γίνει αντιληπτό από ένα αντικείμενο**, όπως η λήψη σημάτων (κλήση μεθόδων) από άλλα αντικείμενα, αλλαγές σε συγκεκριμένες τιμές ή η πάροδος του χρόνου.
- **Μία κατάσταση είναι ένα σύνολο τιμών αντικειμένων** για μία δεδομένη κλάση που αντιδρούν ποιοτικά κατά τον ίδιο τρόπο σε ένα γεγονός. **Με άλλα λόγια, όλα τα αντικείμενα που βρίσκονται στην ίδια κατάσταση εκτελούν τις ίδιες ενέργειες με την ανίχνευση του ίδιου γεγονότος.**

Γεγονότα (Events)

- Ένα γεγονός θα πρέπει να σημειωθεί ότι έχει χωρική και χρονική θέση στο σύστημα, αλλά **δεν έχει διάρκεια**.
- Η μοντελοποίηση κάποιου πράγματος ως γεγονός, συνεπάγεται ότι η εμφάνιση του θα έχει κάποιες συνέπειες. Ο όρος γεγονός αντιστοιχεί στην περιγραφή όλων των γεγονότων ίδιου τύπου που μπορεί να λάβουν χώρα, ακριβώς όπως μια κλάση αποτελεί την κατηγορία στην οποία ανήκουν αντικείμενα με κοινά χαρακτηριστικά. Ένα συγκεκριμένο γεγονός ονομάζεται και **στιγμιότυπο γεγονότος** και μπορεί να έχει παραμέτρους που το χαρακτηρίζουν σε σχέση με άλλα στιγμιότυπα του ίδιου γεγονότος. Υπάρχουν διάφοροι τύποι γεγονότων:

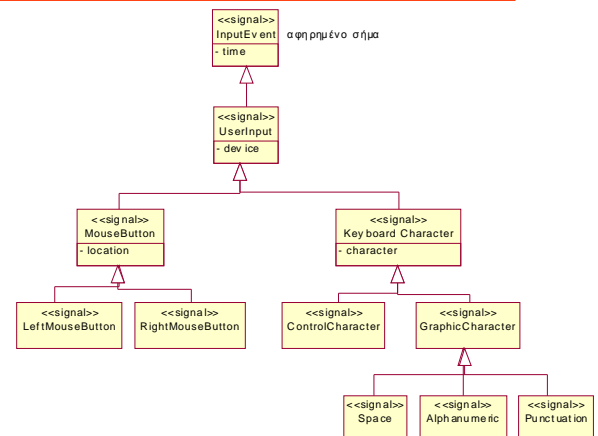
80

Γεγονότα (Events)

- **Γεγονός-σήμα:** Ένα σήμα είναι μία ονοματισμένη οντότητα που λειτουργεί ως μέσο επικοινωνίας μεταξύ δύο αντικειμένων. Η **λήψη ενός σήματος αποτελεί γεγονός για τον αποδέκτη**.
- Τα σήματα αποτελούν **ασύγχρονη** και **μονόδρομη** επικοινωνία: ο αποστολέας δεν περιμένει να επεξεργαστεί το σήμα ο αποδέκτης αλλά συνεχίζει ανεξάρτητα τη δική του εργασία. Για τη μοντελοποίηση αμφίδρομης επικοινωνίας, θα πρέπει να χρησιμοποιηθούν πολλαπλά σήματα, τουλάχιστον ένα σε κάθε κατεύθυνση.
- Τα σήματα είναι δυνατόν να δηλωθούν σε διαγράμματα κλάσεων ως οντότητες με τη λέξη `<<signal>>`. Οι παράμετροι του σήματος δηλώνονται ως ιδιότητες. Τα σήματα επίσης μπορούν να συμμετέχουν σε σχέσεις γενίκευσης και να κληρονομούν παραμέτρους από γονικά σήματα.

81

Γεγονότα (Events)



Γεγονότα (Events)

- **Γεγονός-Αλλαγή:** **Μία αλλαγή είναι η ικανοποίηση μιας λογικής συνθήκης** που εξαρτάται από ορισμένες τιμές ιδιοτήτων. Με αυτό τον τρόπο επιτυγχάνεται αναμονή μέχρι την ικανοποίηση της συνθήκης αλλά θα πρέπει να χρησιμοποιείται με προσοχή καθώς συνεπάγεται έναν συνεχή και ενδεχομένως μη τοπικό υπολογισμό (επειδή οι τιμές που ελέγχονται μπορεί να είναι απομακρυσμένες).
- **Γεγονός-Χρόνος:** Ένα χρονικό γεγονός αναπαριστά την πάροδο του χρόνου. Ένα χρονικό γεγονός μπορεί να προσδιοριστεί είτε με **απόλυτο τρόπο** (ώρα, ημέρα) ή με **σχετικό τρόπο** (χρόνος που παρήλθε από ένα γεγονός).

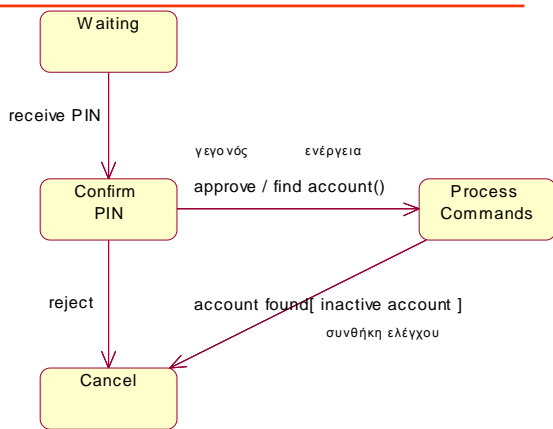
83

Μηχανή Καταστάσεων

- **Κατάσταση.** Μία κατάσταση περιγράφει μία χρονική περίοδο κατά τη διάρκεια ζωής ενός αντικειμένου. Μπορεί να χαρακτηριστεί με τρεις συμπληρωματικούς τρόπους:
 - ως ένα σύνολο τιμών αντικειμένων που είναι παρόμοιες από κάποια άποψη,
 - ως μία περίοδος κατά την οποία ένα αντικείμενο αναμένει την εμφάνιση ενός γεγονότος,
 - ως μία περίοδος κατά την οποία ένα αντικείμενο εκτελεί μία εργασία.
- Όταν ένα αντικείμενο βρίσκεται σε μία κατάσταση, αποκρίνεται σε γεγονότα που βρίσκονται στις μεταβάσεις που ξεκινούν από την κατάσταση αυτή. Μία κατάσταση συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες.

84

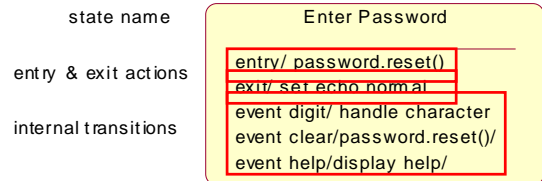
Διάγραμμα Καταστάσεων



85

Διάγραμμα Καταστάσεων

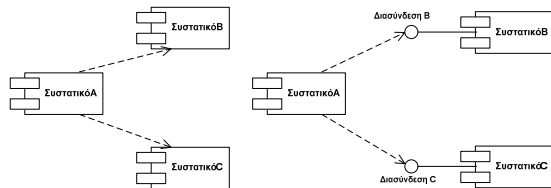
Μία κατάσταση είναι δυνατόν να περιλαμβάνει **ενέργειες εισόδου** που λαμβάνουν χώρα όταν μία μετάβαση καθιστά την κατάσταση ενεργή καθώς και **ενέργειες εξόδου** που λαμβάνουν χώρα προτού μία μετάβαση "φύγει" από την ενεργό κατάσταση. Μία **εσωτερική μετάβαση** έχει αρχική κατάσταση αλλά όχι τελική. Κατά συνέπεια δεν προκαλεί αλλαγή της ενεργού κατάστασης, αλλά μπορεί να έχει ενέργειες που λαμβάνουν χώρα



86

Φυσική Άποψη – Διαγράμματα Συστατικών

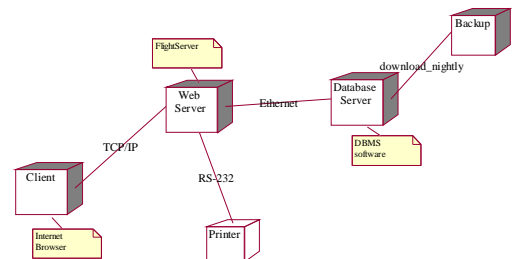
- Τα διαγράμματα συστατικών (component diagrams) αναπαριστούν το φυσικό διαχωρισμό του συστήματος
- Ένα συστατικό είναι μία **φυσική μονάδα υλοποίησης κώδικα** (πηγαίο αρχείο, αρχείο επικεφαλίδων, εκτελέσιμο αρχείο, μια βιβλιοθήκη .dll) κ.ο.κ.
- Το διάγραμμα συστατικών απεικονίζει το δίκτυο των εξαρτήσεων μεταξύ των συστατικών



87

Φυσική Άποψη – Διαγράμματα Ανάπτυξης

- Τα διαγράμματα ανάπτυξης (deployment diagrams) αναπαριστούν την αντιστοίχιση του λογισμικού σε επεξεργαστικές μονάδες - κόμβους
- Ένας κόμβους (node) είναι ένα φυσικό αντικείμενο που στη γενική περίπτωση έχει τουλάχιστον μνήμη και δυνατότητα επεξεργασίας



88