
Η γλώσσα μοντελοποίησης UML

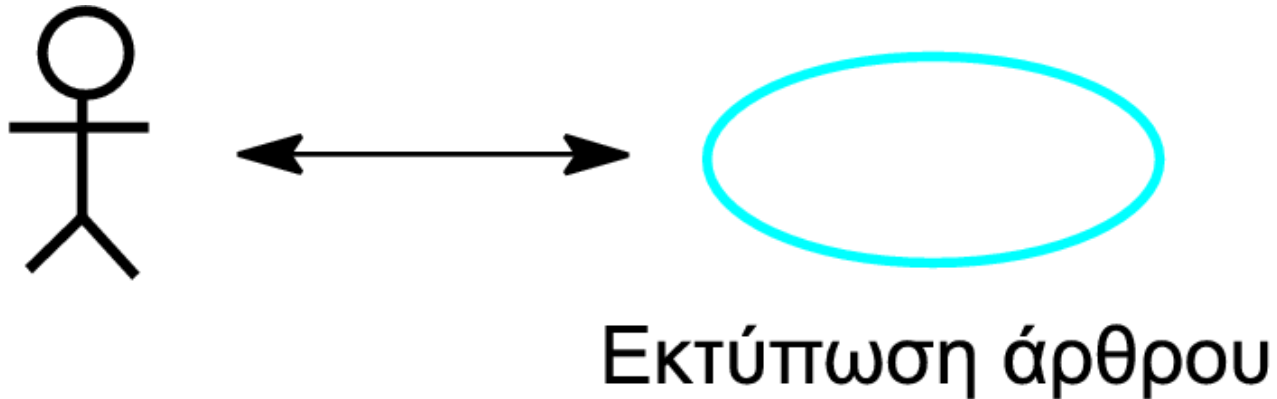
Περιεχόμενα

- Διαγράμματα περιπτώσεων χρήσης
- Διαγράμματα ακολουθίας
- Διαγράμματα συνεργασίας
- Διαγράμματα καταστάσεων
- Διαγράμματα κλάσεων

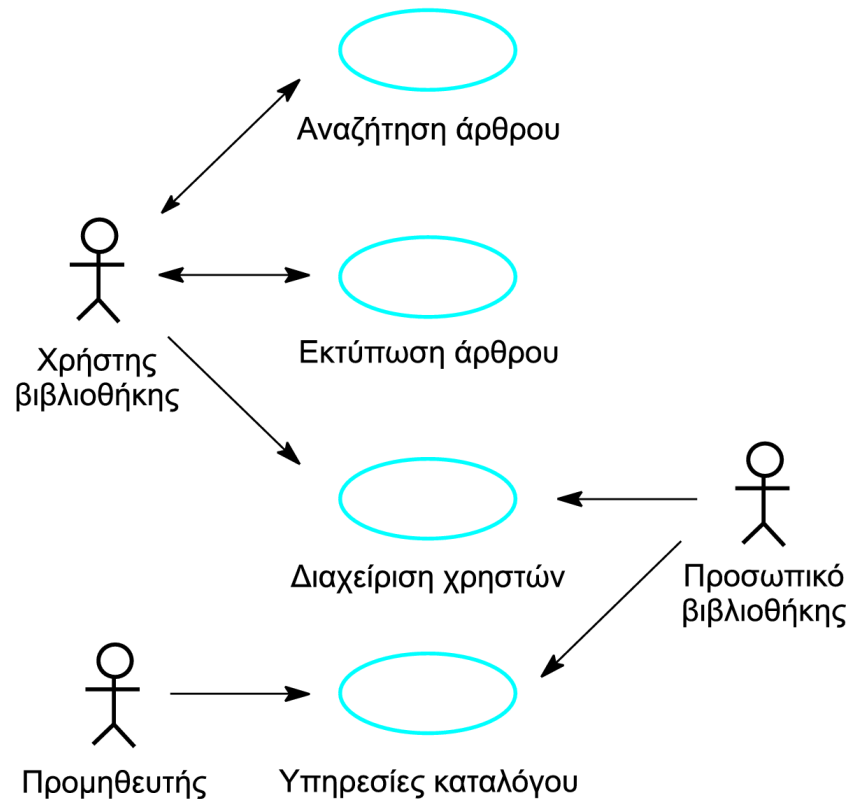
Περιπτώσεις χρήσης

- Οι περιπτώσεις χρήσης είναι μια τεχνική εξαγωγής απαιτήσεων βάσει σεναρίων σε γλώσσα UML που προσδιορίζει τους συμμετέχοντες σε μια αλληλεπίδραση και περιγράφει την ίδια την αλληλεπίδραση.
- Ένα σύνολο περιπτώσεων χρήσης πρέπει να περιγράφει όλες τις πιθανές αλληλεπιδράσεις με το σύστημα.
- Για την προσθήκη λεπτομερειών στις περιπτώσεις χρήσης μπορούν να χρησιμοποιηθούν διαγράμματα ακολουθίας τα οποία εμφανίζουν την ακολουθία της επεξεργασίας γεγονότων στο σύστημα.

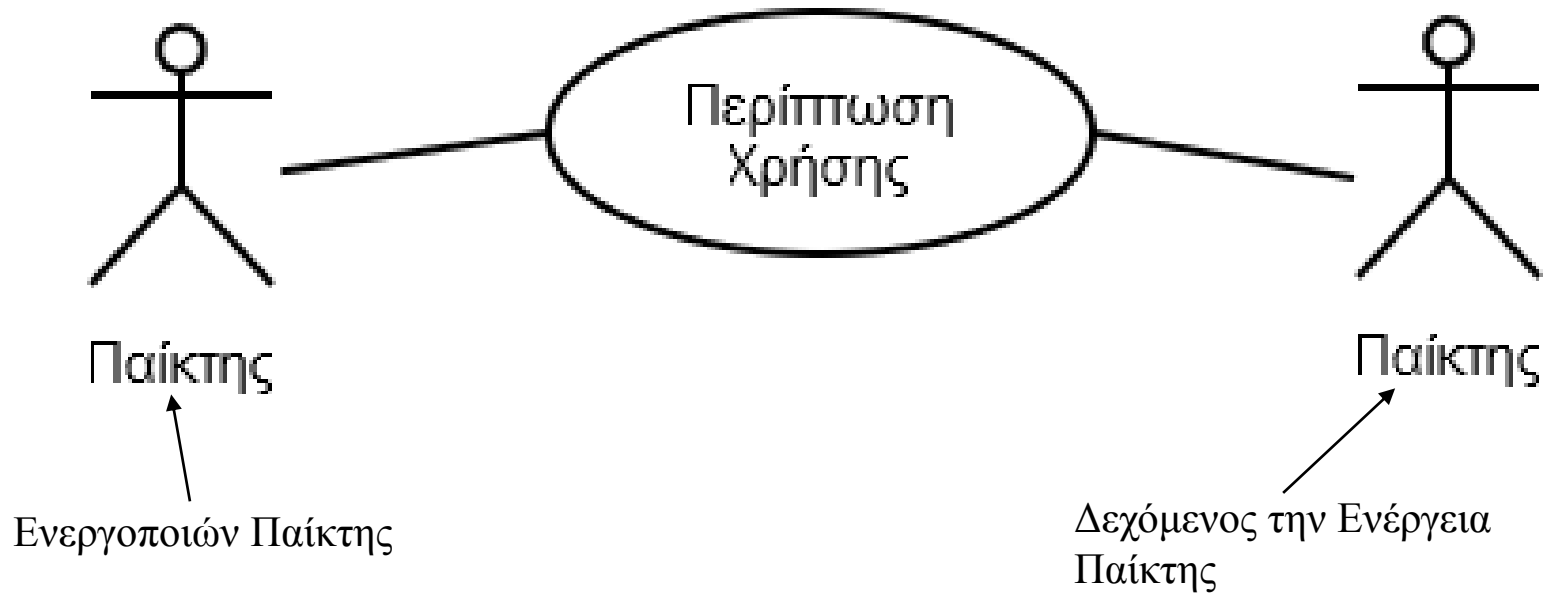
Περίπτωση χρήσης: Εκτύπωση ενός άρθρου



Περιπτώσεις χρήσης του συστήματος LIBSYS



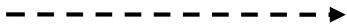
Διαγράμματα Περιπτώσεων Χρήσης: Εισαγωγή



- **Συνοδευτικό κείμενο**
 - Παίκτης που ενεργοποιεί
 - Προϋποθέσεις
 - Βήματα
 - Συνθήκες μετά το πέρας
 - Δεχόμενος την ενέργεια παίκτης

Διαγράμματα Περιπτώσεων Χρήσης: Επαναχρησιμοποίηση

<<include>>



- Συμπερίληψη (inclusion)

- Νέο σενάριο χρήσης ως υπεर्सύνολο υπάρχοντος

<<extends>>



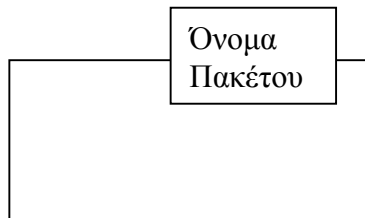
- Επέκταση (extension)

- Προθήκη βημάτων σε υπάρχον σενάριο χρήσης



- Γενίκευση

- Κληρονομικότητα: Το παιδί κληρονομεί από τον πατέρα και προσθέτει και τη δική του συμπεριφορά



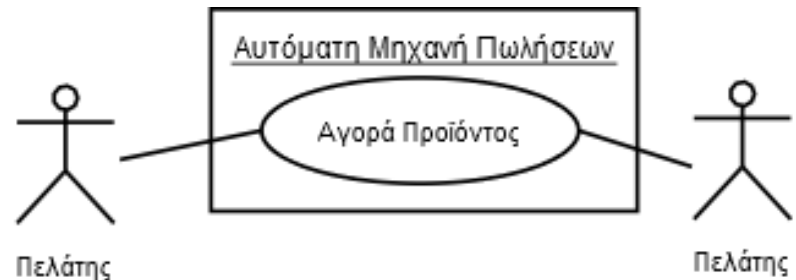
- Ομαδοποίηση

- Παρόμοιες περιπτώσεις χρήσης

Διαγράμματα Περιπτώσεων

Χρήσης: Παράδειγμα

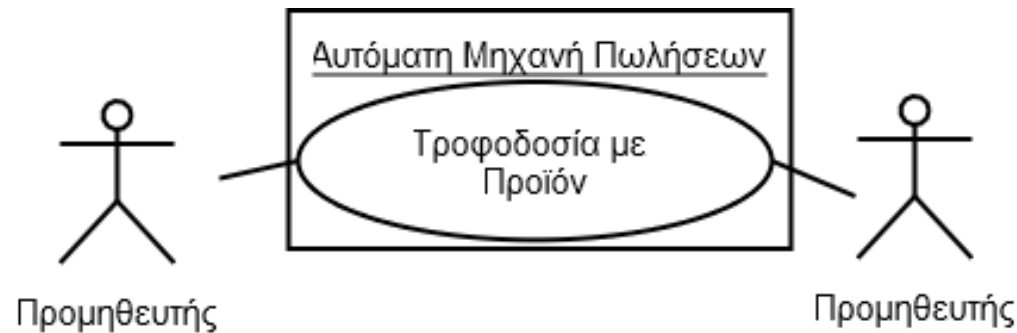
- Αυτόματη μηχανή πωλήσεων
- Αγορά προϊόντος
 - Παίκτης: Αγοραστής
 - Βάζει χρήματα, επιλέγει προϊόν, η μηχανή του το δίνει
 - Αν δεν έχει προϊόν;
 - Επιστροφή χρημάτων ή επιλογή άλλου προϊόντος
 - Λάθος ποσό χρημάτων;
 - Επιστροφή χρημάτων στον αγοραστή
 - Προϋπόθεση
 - Πεινασμένος ή διψασμένος πελάτης
 - Συνθήκες με το πέρας
 - Προϊόν από τη μηχανή ή επιστροφή χρημάτων



Διαγράμματα Περιπτώσεων

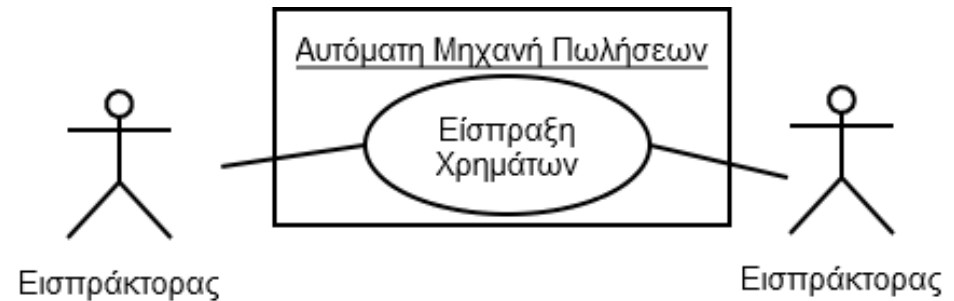
Χρήσης: Παράδειγμα

- Τροφοδοσία με προϊόντα
 - Παίκτης: Προμηθευτής
 - Διαδικασία: Απασφάλιση μηχανής, επανατροφοδοσία κάθε προϊόντος (γέμισμα), κλείσιμο και ασφάλιση μηχανής
- Προϋπόθεση: Εκπνοή του προκαθορισμένου χρονικού διαστήματος μεταξύ επανατροφοδοσιών
- Συνθήκες μετά το πέρας: Πιθανές νέες πωλήσεις για τον προμηθευτή

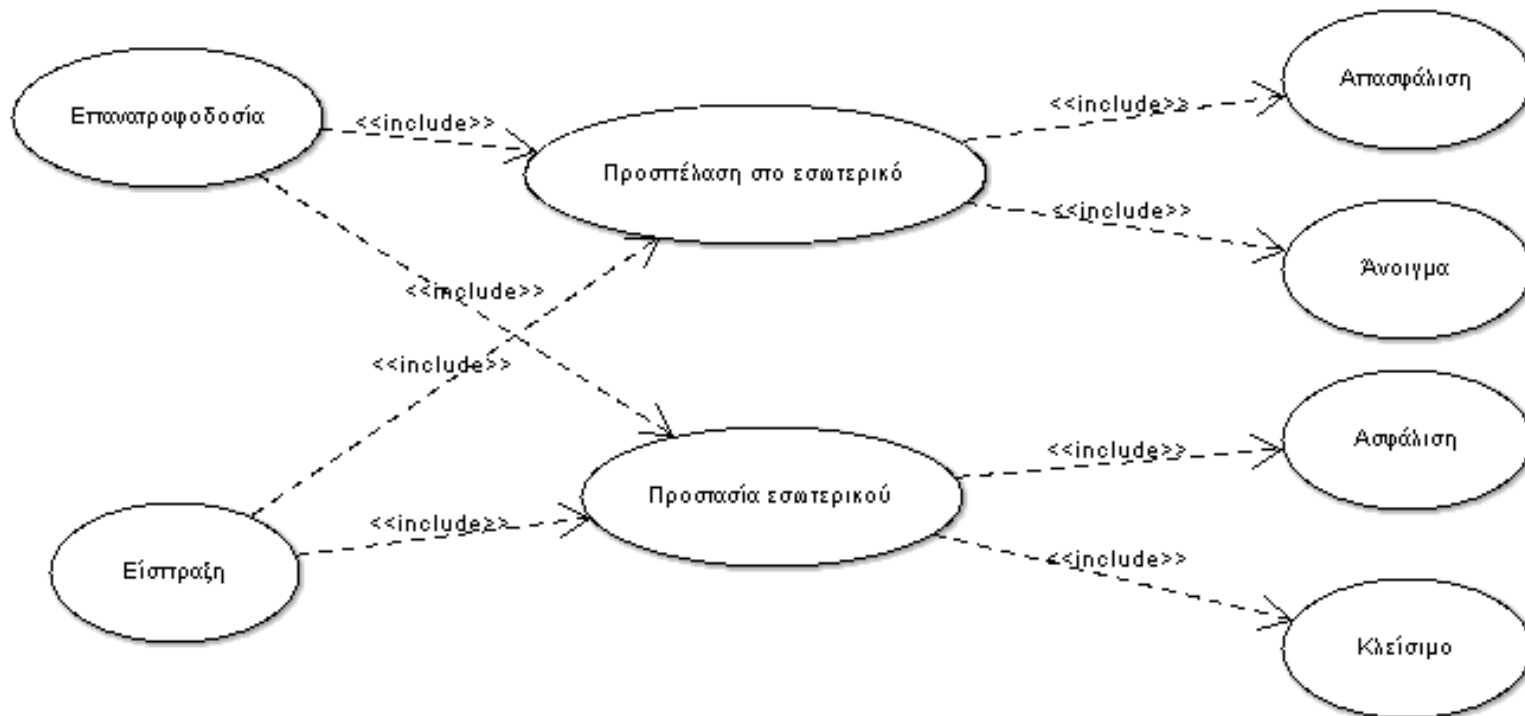


Διαγράμματα Περιπτώσεων Χρήσης: Παράδειγμα

- Είσπραξη χρημάτων
 - Ίδια βήματα με τον προμηθευτή
 - Χρήματα αντί προϊόντων
 - Συνθήκη με το πέρας: Χρήματα στα χέρια του εισπράκτορα χρημάτων

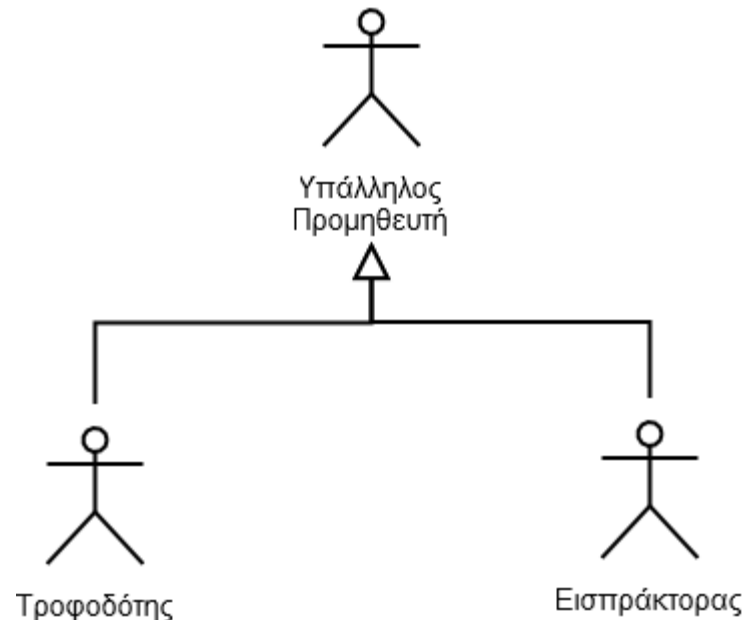


Διαγράμματα Περιπτώσεων Χρήσης: Παράδειγμα

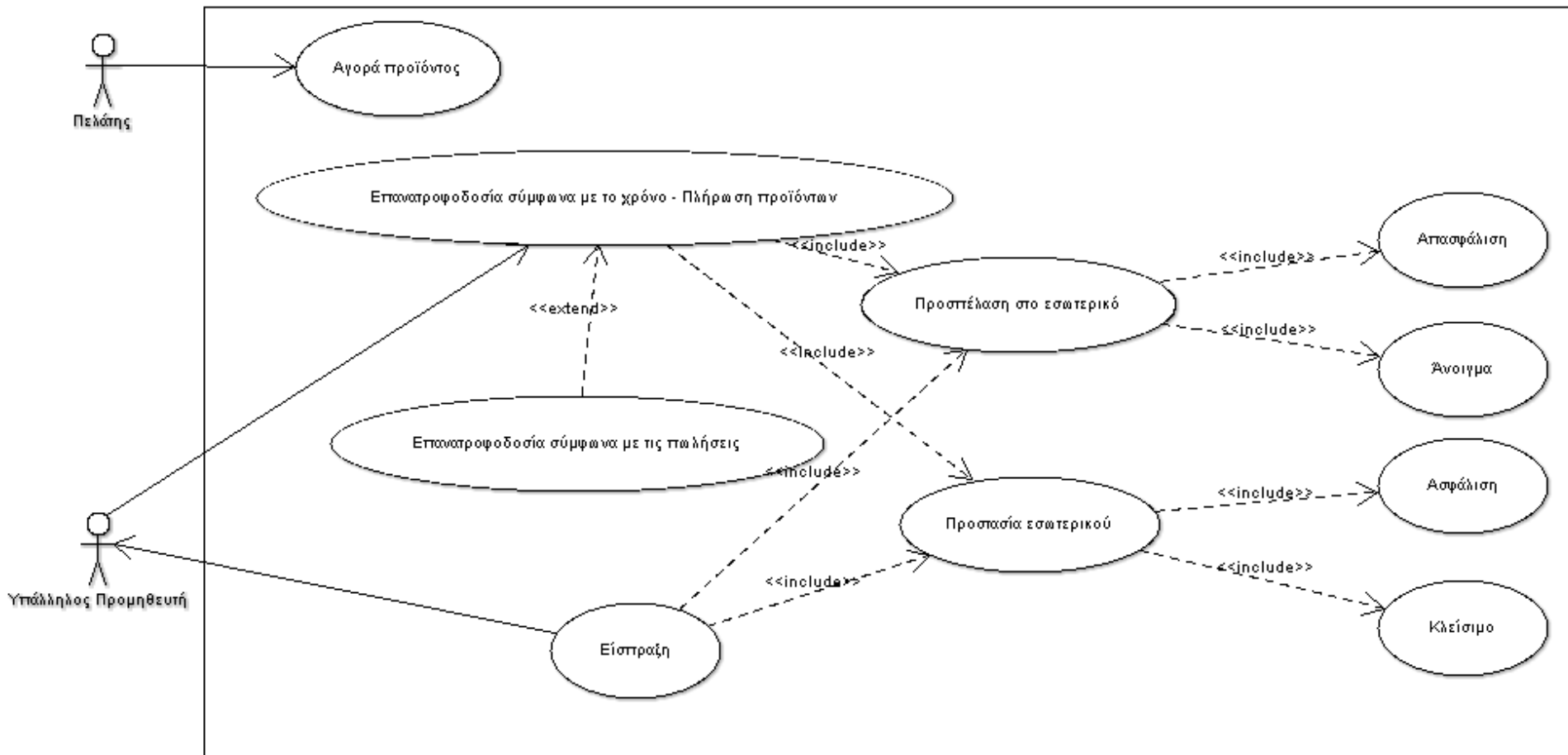


Διαγράμματα Περιπτώσεων Χρήσης: Παράδειγμα

- Γενίκευση:
 - Εισπράκτορας & Τροφοδότης το ίδιο πρόσωπο (Υπάλληλος Προμηθευτή)



Διαγράμματα Περιπτώσεων Χρήσης: Παράδειγμα

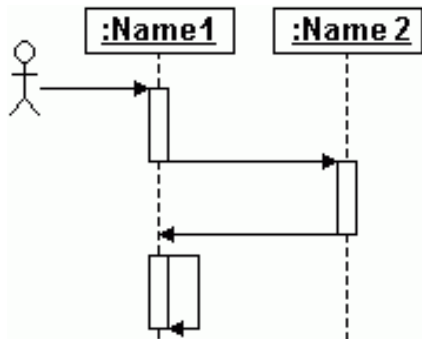
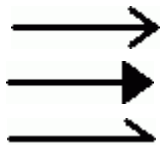
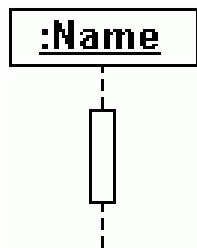


Διαγράμματα Ακολουθιών

Τα διαγράμματα ακολουθιών δείχνουν **πώς** τα αντικείμενα επικοινωνούν μεταξύ τους με το πέρασμα του **χρόνου**

- Χρόνος
- Απεικόνιση
- Αντικείμενα
- Μηνύματα

Διαγράμματα Ακολουθιών: Απεικόνιση



- Αντικείμενο
 - Όνομα
 - Γραμμή ζωής (διακεκομμένη)
 - Ενεργοποίηση (διάρκεια εκτέλεσης λειτουργίας)
- Μήνυμα
 - Απλό (μεταφορά ελέγχου)
 - Σύγχρονο
 - Ασύγχρονο
- Χρόνος
 - Από την κορυφή προς τα κάτω

Διαγράμματα Ακολουθιών:

Λεπτομέρειες

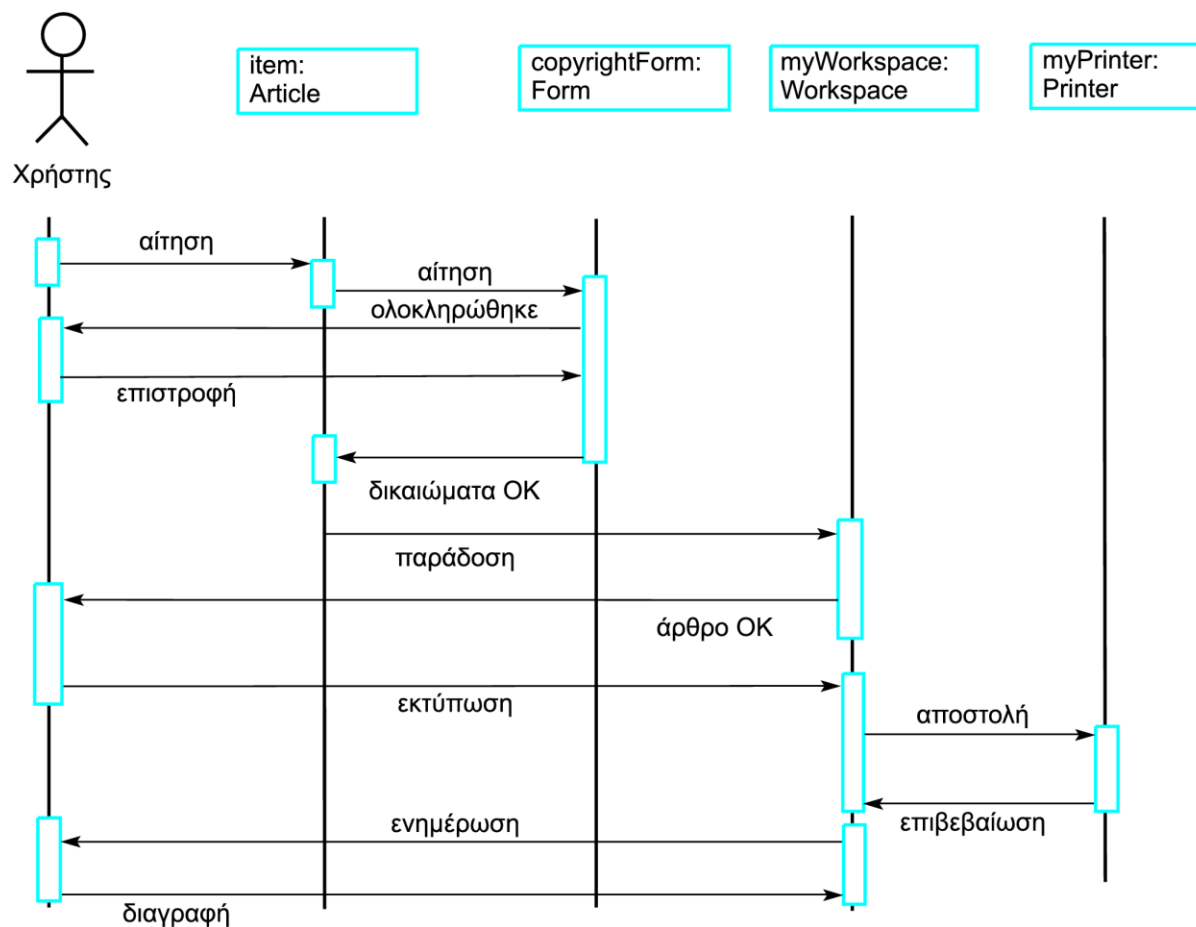
- Μπορεί να αναφέρεται είτε σε **στιγμιότυπο** (περίπτωσης χρήσης) είτε να είναι **γενικό**
 - Γενικά διαγράμματα: Δυνατότητα έκφρασης if και while loops
 - Συνθήκες if και while σε []
- Δυνατότητα **αναδρομής**
 - Λειτουργία που καλεί τον εαυτό της
- Δυνατότητα **δημιουργίας** αντικειμένου
 - Μήνυμα με όνομα **Create()**
 - Το νέο αντικείμενο δεν εμφανίζεται στην κορυφή αλλά στο σημείο που δημιουργήθηκε

Διαγράμματα Ακολουθιών:

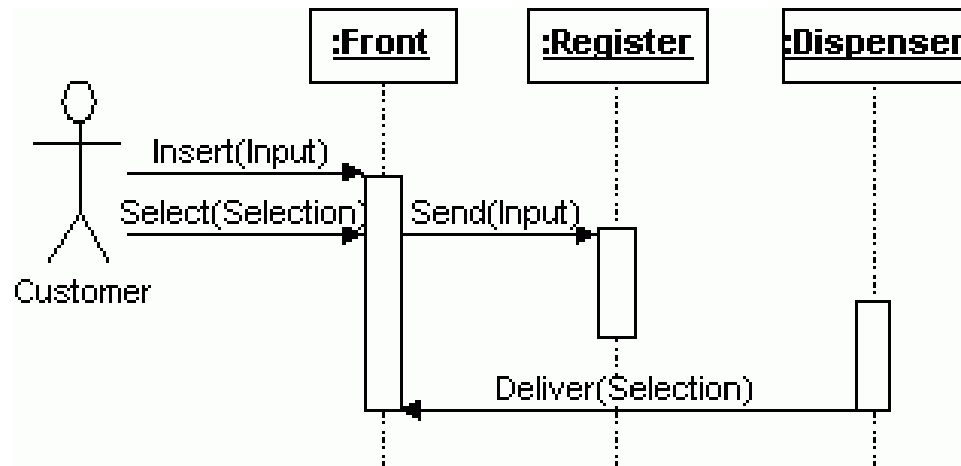
Παράδειγμα

- Αυτόματος Πωλητής
 - Διεπαφή προς πελάτη
 - Ταμείο
 - Παροχέας προϊόντων
- Διάγραμμα ακολουθιών (στιγμιότυπο)
 - Πελάτης βάζει χρήματα στο μηχάνημα
 - Πελάτης επιλέγει προϊόν
 - Τα χρήματα πηγαίνουν στο ταμείο
 - Το ταμείο ελέγχει αν το προϊόν βρίσκεται στον παροχέα
 - Το ταμείο ανανεώνει το ποσό του
 - Το ταμείο δίνει εντολή στον παροχέα να παραδώσει το προϊόν στη διεπαφή με τον πελάτη

Ακολουθία εκτύπωσης άρθρου



Διαγράμματα Ακολουθιών: Παράδειγμα

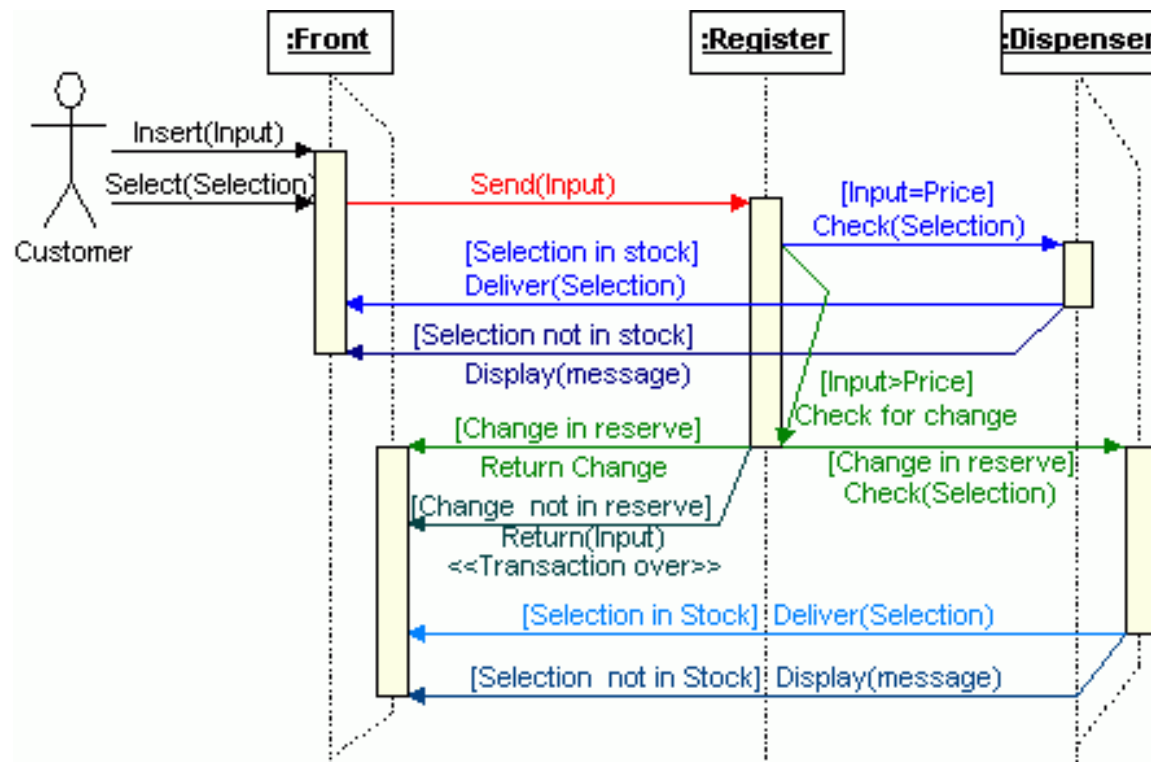


Διαγράμματα Ακολουθιών:

Παράδειγμα

- Δεν υπάρχει το προϊόν
 - Μετά την επιλογή προϊόντος που έχει εξαντληθεί εμφανίζεται σχετικό μήνυμα στην οθόνη
 - Προτροπή για άλλη επιλογή
 - Επιλογή επιστροφής χρημάτων
 - Αν ο πελάτης διαλέξει νέο προϊόν που υπάρχει προχωράμε όπως στο «ιδανικό» σενάριο, αν τα χρήματα είναι σωστά. Διαφορετικά, ακολουθείται το σενάριο λάθος χρημάτων
 - Αν ο πελάτης διαλέξει νέο προϊόν που επίσης δεν υπάρχει, η διαδικασία επαναλαμβάνεται έως ότου επιλεγεί προϊόν που υπάρχει ή ζητηθεί επιστροφή χρημάτων

Διαγράμματα Ακολουθιών: Παράδειγμα

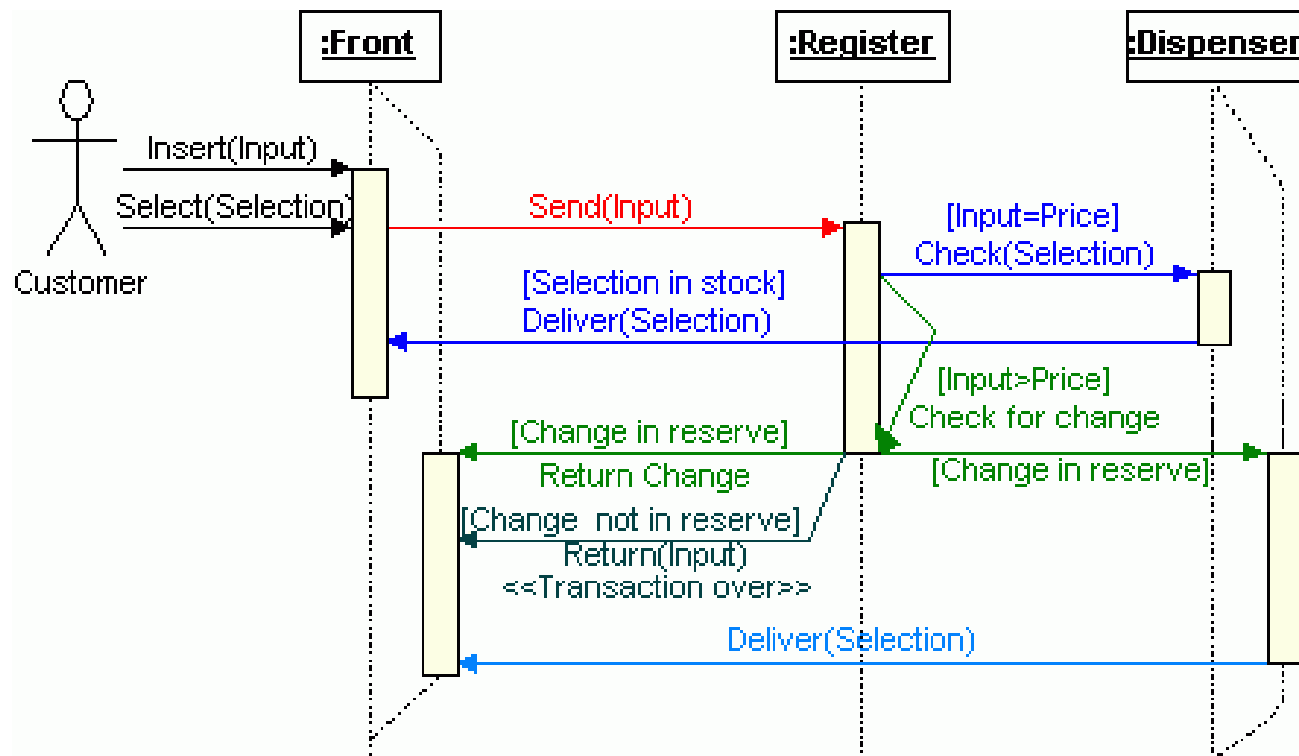


Διαγράμματα Ακολουθιών:

Παράδειγμα

- Λάθος ποσό χρημάτων
 - Το ταμείο ελέγχει το ποσό που έβαλε ο πελάτης και το συγκρίνει με την τιμή του προϊόντος
 - Αν το ποσό είναι μεγαλύτερο από την τιμή του προϊόντος, υπολογίζεται η διαφορά και ελέγχεται το ταμείο για μετρητά
 - Αν η διαφορά υπάρχει, το ταμείο επιστρέφει τα ρέστα και συνεχίζουμε κανονικά
 - Διαφορετικά το ταμείο επιστρέφει το ποσό και εκδίδει μήνυμα που προτρέπει να εισαχθεί το ακριβές ποσό
 - Αν το ποσό είναι μικρότερο από την τιμή το ταμείο δεν κάνει τίποτα και περιμένει για περισσότερα χρήματα

Διαγράμματα Ακολουθιών: Παράδειγμα

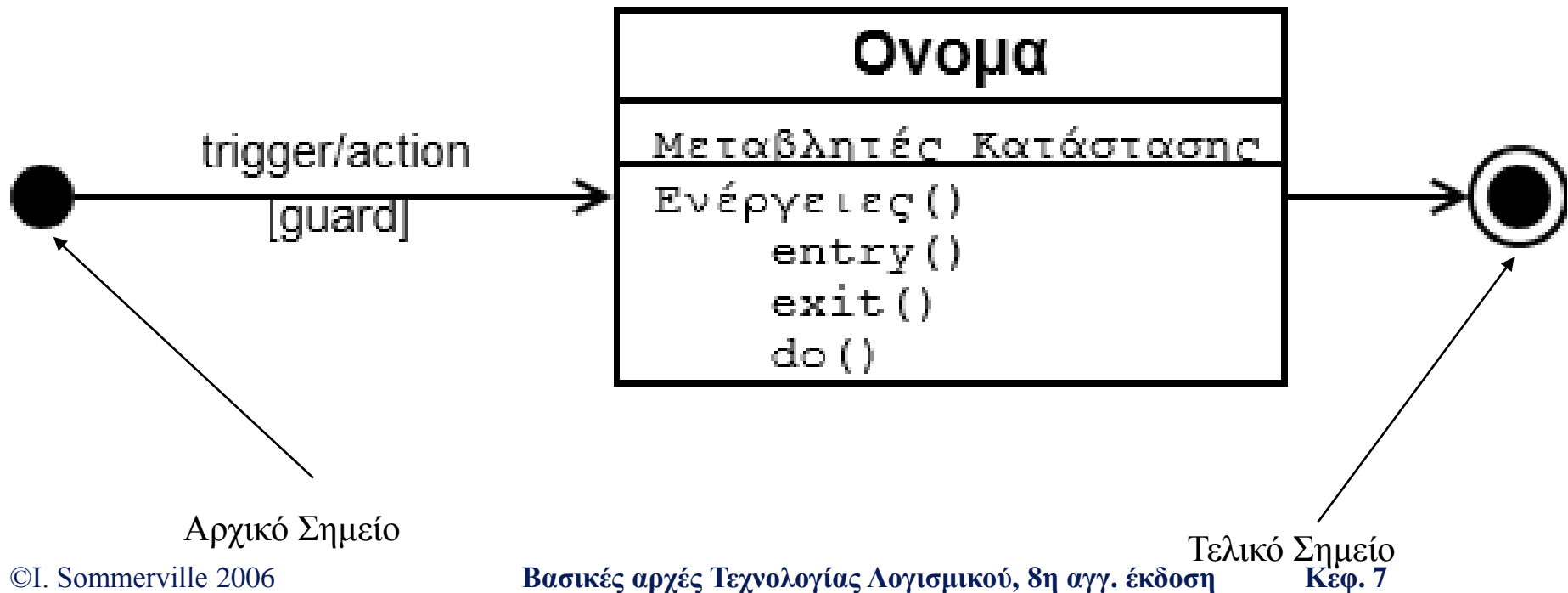


Μοντέλα μηχανής καταστάσεων

- Μοντελοποιούν την απόκριση του συστήματος σε εξωτερικά και εσωτερικά συμβάντα.
- Δείχνουν την απόκριση του συστήματος σε ερεθίσματα, γι' αυτό και συχνά χρησιμοποιούνται για τη μοντελοποίηση συστημάτων πραγματικού χρόνου.
- Στα μοντέλα μηχανών καταστάσεων οι καταστάσεις του συστήματος είναι οι κόμβοι και τα συμβάντα είναι τα βέλη ή τόξα μεταξύ των κόμβων. Όταν λαμβάνει χώρα ένα συμβάν, το σύστημα μεταβαίνει από μία κατάσταση σε μια άλλη.
- Τα διαγράμματα καταστάσεων είναι αναπόσπαστο κομμάτι της Ενοποιημένης Γλώσσας Μοντελοποίησης και χρησιμοποιούνται για την αναπαράσταση μοντέλων μηχανών καταστάσεων.

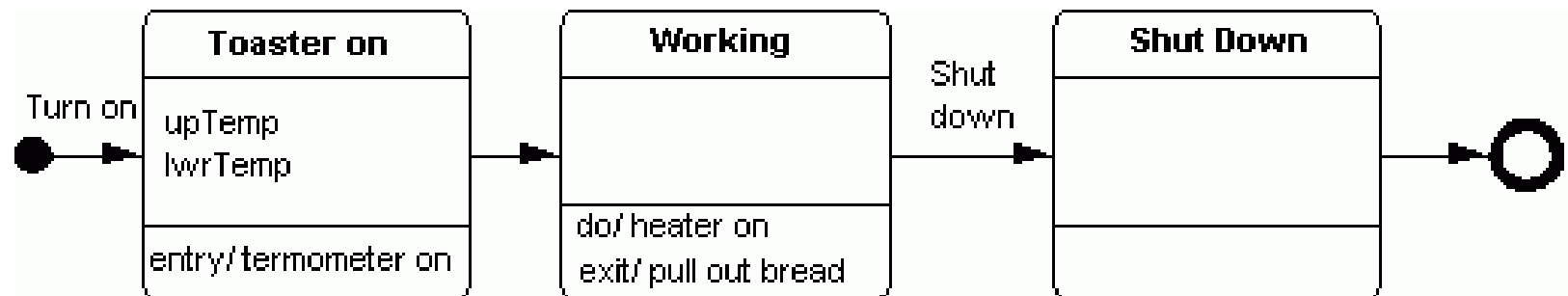
Διαγράμματα Καταστάσεων

- Δείχνει τις **καταστάσεις** ενός αντικειμένου
 - Και τις **μεταβάσεις** μεταξύ καταστάσεων στο πέρασμα του **χρόνου**
 - Π.χ. για άνθρωπο: νεογέννητο, μωρό, παιδί, έφηβος, νέος, ενήλικας, μεσήλικας, ηλικιωμένος.
 - Δυνατές υποκαταστάσεις (μέσα σε μια κατάσταση)



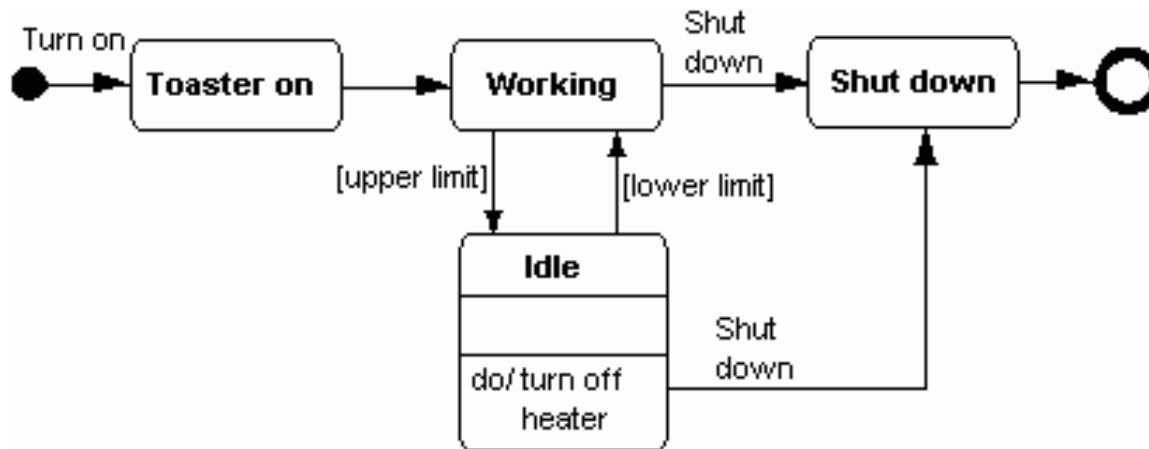
Διαγράμματα Καταστάσεων: Παράδειγμα

- Διαδικασία:
 - Ενεργοποίηση τοστιέρας
 - Τοποθέτηση ψωμιού
 - Αναμονή
 - Ψημένο ψωμί

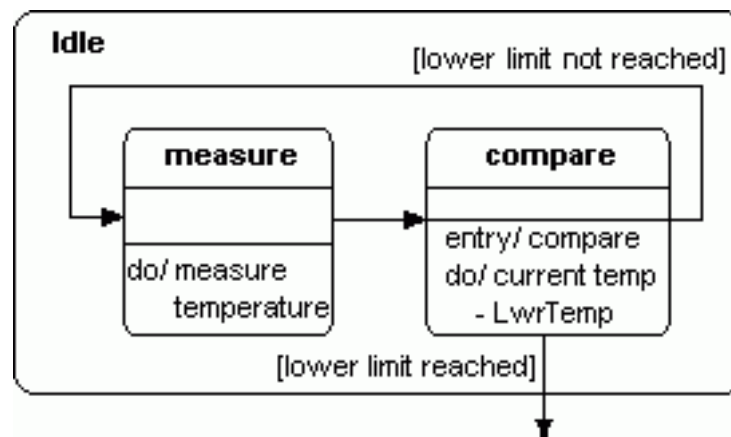
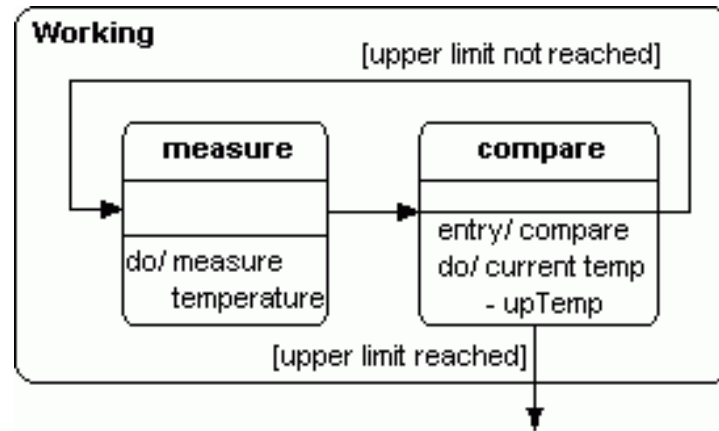


Διαγράμματα Καταστάσεων: Παράδειγμα

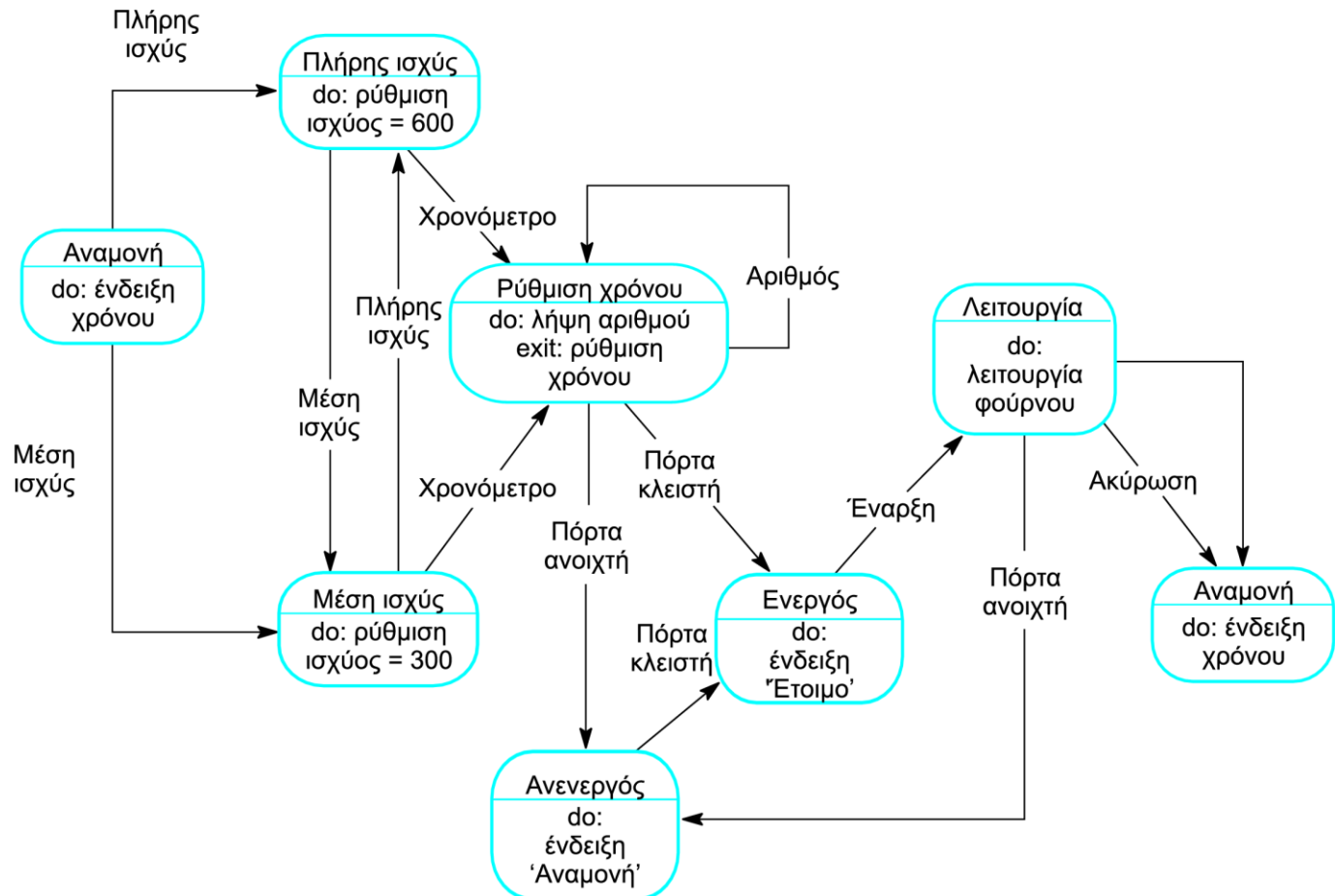
- Θερμοστάτης
 - Για τη διατήρηση ορίων θερμοκρασίας



Διαγράμματα Καταστάσεων: Παράδειγμα – Περισσότερη Λεπτομέρεια



Μοντέλο φούρνου μικροκυμάτων



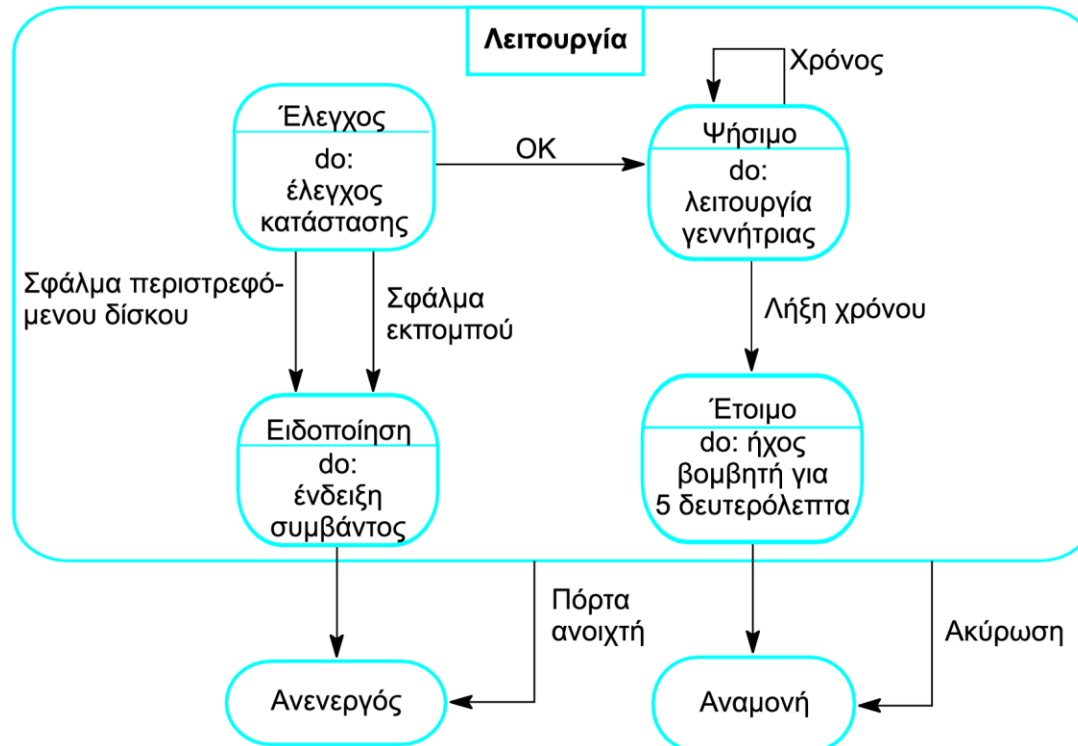
Περιγραφή καταστάσεων φούρνου μικροκυμάτων

Κατάσταση	Περιγραφή
Αναμονή	Ο φούρνος αναμένει είσοδο. Η οθόνη δείχνει την τρέχουσα ώρα.
Μέση ισχύς	Η ισχύς του φούρνου ρυθμίζεται στα 300 watt. Η οθόνη δείχνει 'Μέση ισχύς'.
Πλήρης ισχύς	Η ισχύς του φούρνου ρυθμίζεται στα 600 watt. Η οθόνη δείχνει 'Πλήρης ισχύς'.
Ρύθμιση χρόνου	Ρυθμίζεται ο χρόνος ψησίματος σύμφωνα με την τιμή εισόδου του χρήστη. Η οθόνη εμφ τον επιλεγμένο χρόνο ψησίματος και ενημερώνεται καθώς ρυθμίζεται ο χρόνος.
Ανενεργός	Η λειτουργία του φούρνου έχει απενεργοποιηθεί για λόγους ασφάλειας. Ο εσωτερικός φω του φούρνου είναι αναμμένος. Η οθόνη δείχνει 'Όχι έτοιμο'.
Ενεργός	Η λειτουργία του φούρνου είναι ενεργοποιημένη. Ο εσωτερικός φωτισμός του φούρνου ε σβηστός. Η οθόνη δείχνει 'Έτοιμο για ψήσιμο'.
Λειτουργία	Ο φούρνος λειτουργεί. Ο εσωτερικός φωτισμός του φούρνου είναι αναμμένος. Η οθόνη δ την αντίστροφη μέτρηση του χρόνου. Με την ολοκλήρωση του ψησίματος, ηχεί ο βομβη δευτερόλεπτα. Ο φωτισμός του φούρνου είναι αναμμένος. Η οθόνη δείχνει 'Ολοκλήρωσι

Ερεθίσματα φούρνου μικροκυμάτων

Ερέθισμα	Περιγραφή
Μέση ισχύς	Ο χρήστης έχει πατήσει το κουμπί της μέσης ισχύος.
Πλήρης ισχύς	Ο χρήστης έχει πατήσει το κουμπί της πλήρους ισχύος.
Χρονοδιακόπτης	Ο χρήστης έχει πατήσει ένα από τα κουμπιά του χρονοδ
Αριθμός	Ο χρήστης έχει πατήσει ένα αριθμητικό πλήκτρο.
Πόρτα ανοιχτή	Ο διακόπτης της πόρτας το φούρνου δεν είναι κλειστός.
Πόρτα κλειστή	Ο διακόπτης της πόρτας το φούρνου είναι κλειστός.
Έναρξη	Ο χρήστης έχει πατήσει το κουμπί της έναρξης.
Ακύρωση	Ο χρήστης έχει πατήσει το κουμπί της ακύρωσης.

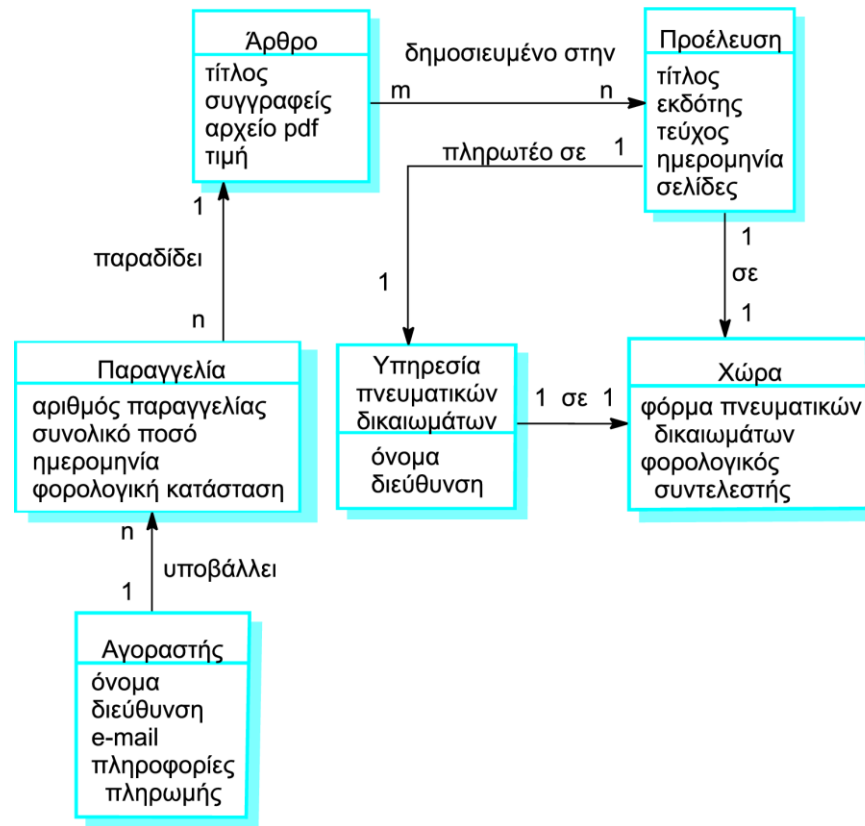
Λειτουργία φούρνου μικροκυμάτων



Σημασιολογικά μοντέλα δεδομένων

- Χρησιμοποιούνται για την περιγραφή της λογικής δομής των δεδομένων που επεξεργάζεται το σύστημα.
- Τα μοντέλα οντοτήτων-σχέσεων-γνωρισμάτων καθορίζουν τις οντότητες του συστήματος, τις σχέσεις μεταξύ τους και τα γνωρίσματά τους.
- Χρησιμοποιούνται ευρέως στο σχεδιασμό βάσεων δεδομένων. Υλοποιούνται εύκολα με χρήση σχεσιακών βάσεων δεδομένων.
- Δεν υπάρχει συγκεκριμένη σημειογραφία στη Ενοποιημένη Γλώσσα Μοντελοποίησης, αλλά μπορούν να χρησιμοποιηθούν αντικείμενα και συσχετίσεις.

Σημασιολογικό μοντέλο βιβλιοθήκης



Μοντέλα αντικειμένων

- Τα μοντέλα αντικειμένων περιγράφουν το σύστημα με βάση τις κλάσεις αντικειμένων και τις συσχετίσεις των κλάσεων.
- Κλάση αντικειμένων ονομάζεται μια αφαιρετική αναπαράσταση ενός συνόλου αντικειμένων με κοινά γνωρίσματα και των υπηρεσιών (λειτουργιών) που παρέχει κάθε αντικείμενο.
- Μπορούν να παραχθούν διάφορα μοντέλα αντικειμένων
 - Μοντέλα κληρονομικότητας
 - Μοντέλα συνάθροισης
 - Διαδραστικά μοντέλα

Μοντέλα αντικειμένων

- Φυσικός τρόπος αναπαράστασης των οντοτήτων του πραγματικού κόσμου που χειρίζεται το σύστημα
- Πιο αφηρημένες οντότητες είναι δυσκολότερο να μοντελοποιηθούν με αυτήν την προσέγγιση
- Ο προσδιορισμός κλάσεων αντικειμένων αναγνωρίζεται ως δύσκολη διαδικασία που απαιτεί σε βάθος κατανόηση του πεδίου εφαρμογής
- Οι κλάσεις αντικειμένων που αναπαριστούν οντότητες πεδίων εφαρμογών μπορούν να επαναχρησιμοποιηθούν και σε άλλα συστήματα

Μοντέλα κληρονομικότητας

- Οι κλάσεις αντικειμένων του πεδίου εφαρμογής είναι οργανωμένες σε μια ιεραρχία.
- Οι κλάσεις που βρίσκονται στην κορυφή της ιεραρχίας απεικονίζουν τα κοινά χαρακτηριστικά όλων των κλάσεων.
- Οι κλάσεις αντικειμένων κληρονομούν γνώσματα και υπηρεσίες από μία ή περισσότερες υπερκλάσεις. Κατόπιν μπορούν να εξειδικευθούν κατά περίπτωση.
- Ο σχεδιασμός της ιεραρχίας κλάσεων είναι δύσκολη διαδικασία επειδή πρέπει να αποφευχθεί η επανάληψη κλάσεων μεταξύ διαφορετικών κλάδων.

Τα μοντέλα αντικειμένων και η Ενοποιημένη Γλώσσα Μοντελοποίησης

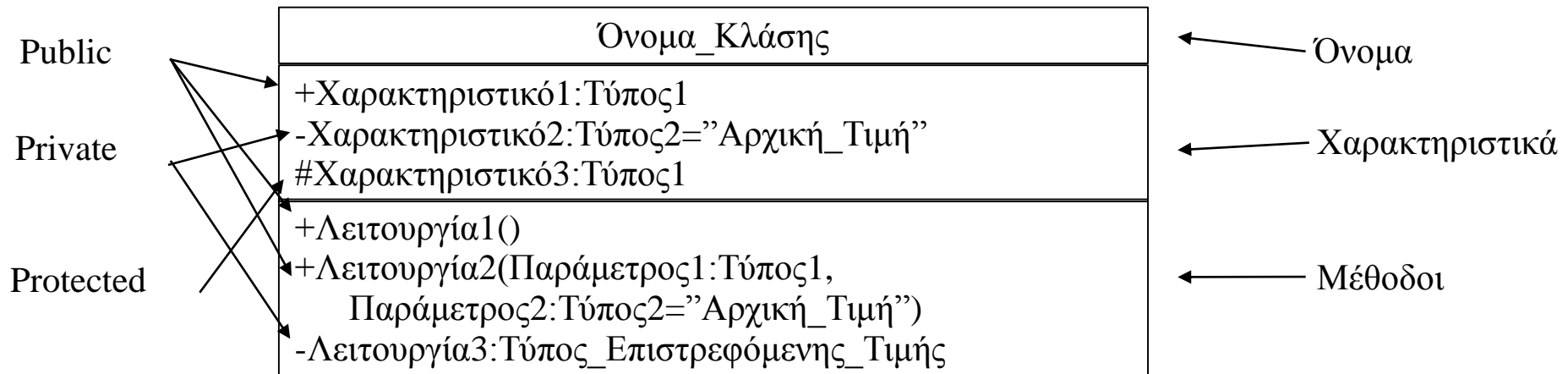
- Η Ενοποιημένη Γλώσσα Μοντελοποίησης (UML) είναι μια τυποποιημένη αναπαράσταση την οποία επινόησαν οι προγραμματιστές των ευρέως χρησιμοποιούμενων αντικειμενοστρεφών μεθόδων ανάλυσης και σχεδιασμού.
- Έχει εξελιχθεί σε ένα αποδοτικό πρότυπο για την αντικειμενοστρεφή μοντελοποίηση.
- Σημειογραφία
 - Οι κλάσεις αντικειμένων αναπαρίστανται με παραλληλόγραμμα τα οποία φέρουν την ονομασία των κλάσεων στην κορυφή, τα γνωρίσματα στο μέσο τους και τις λειτουργίες στο κάτω μέρος.
 - Οι σχέσεις μεταξύ των κλάσεων αντικειμένων (ονομάζονται συσχετίσεις) απεικονίζονται ως γραμμές που συνδέουν αντικείμενα
 - Η κληρονομικότητα αναφέρεται με τον όρο "γενίκευση" και απεικονίζεται στην ιεραρχία με κατεύθυνση προς τα επάνω και όχι προς τα κάτω.

Διάγραμμα κλάσεων: Απεικόνιση

Διάγραμμα κλάσεων

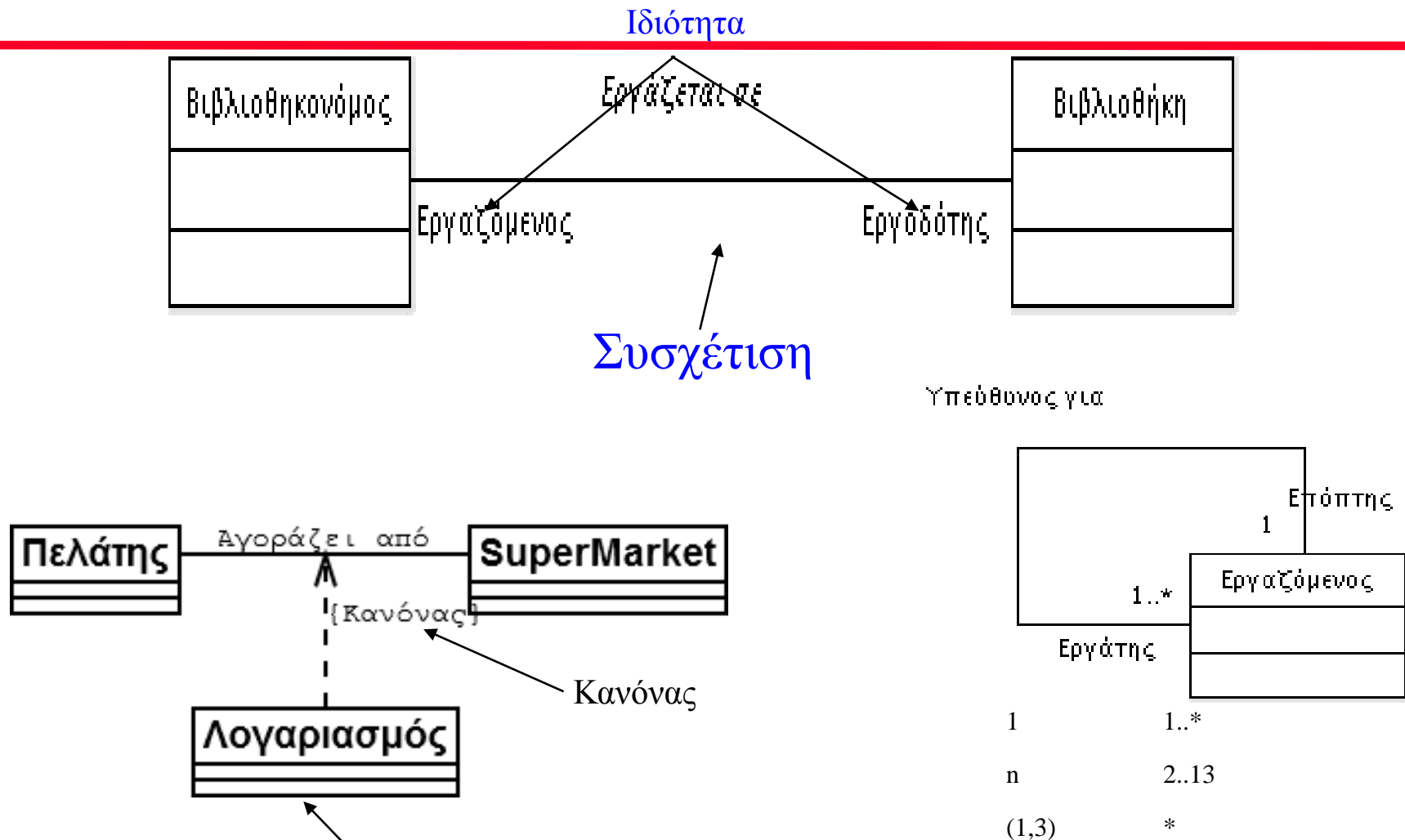
Κλάση: Κατηγορία ή ομάδα «πραγμάτων» με παρόμοια χαρακτηριστικά/πεδία και κοινή συμπεριφορά.

Διαγράμματα κλάσεων: Απεικονίσεις που αξιοποιούνται από την ομάδα ανάπτυξης



- Επιπλέον στοιχεία:
 - Περιορισμοί
 - Κανόνες – Κείμενο μέσα σε {}
 - Σημειώσεις
 - Ελεύθερο κείμενο, γραφικά κλπ.

Διάγραμμα Κλάσεων: Συσχετισμοί



Διαγράμματα Κλάσεων:

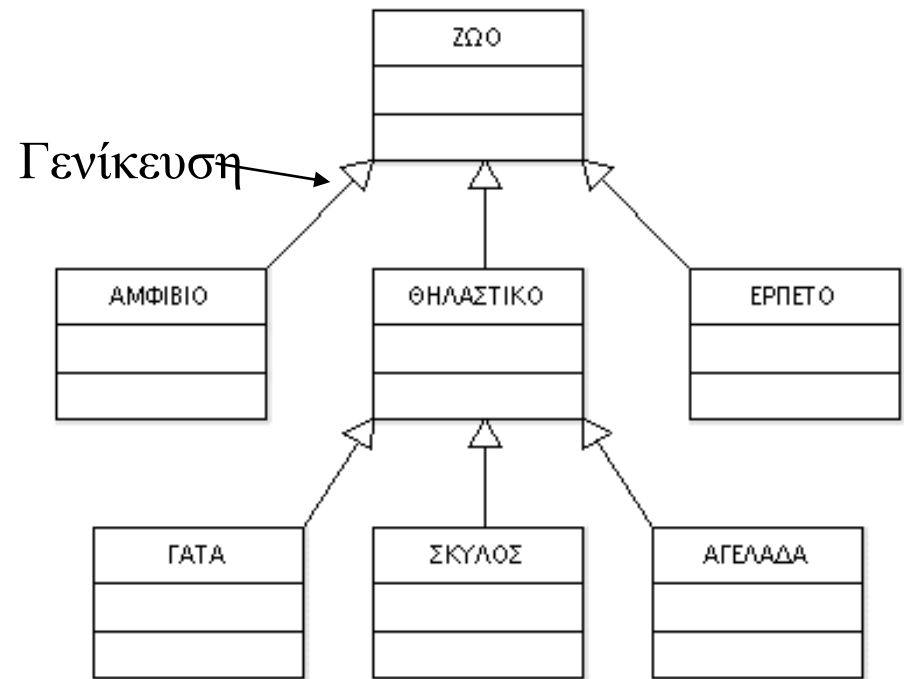
Κληρονομικότητα & Γενίκευση

- **Κληρονομικότητα:**

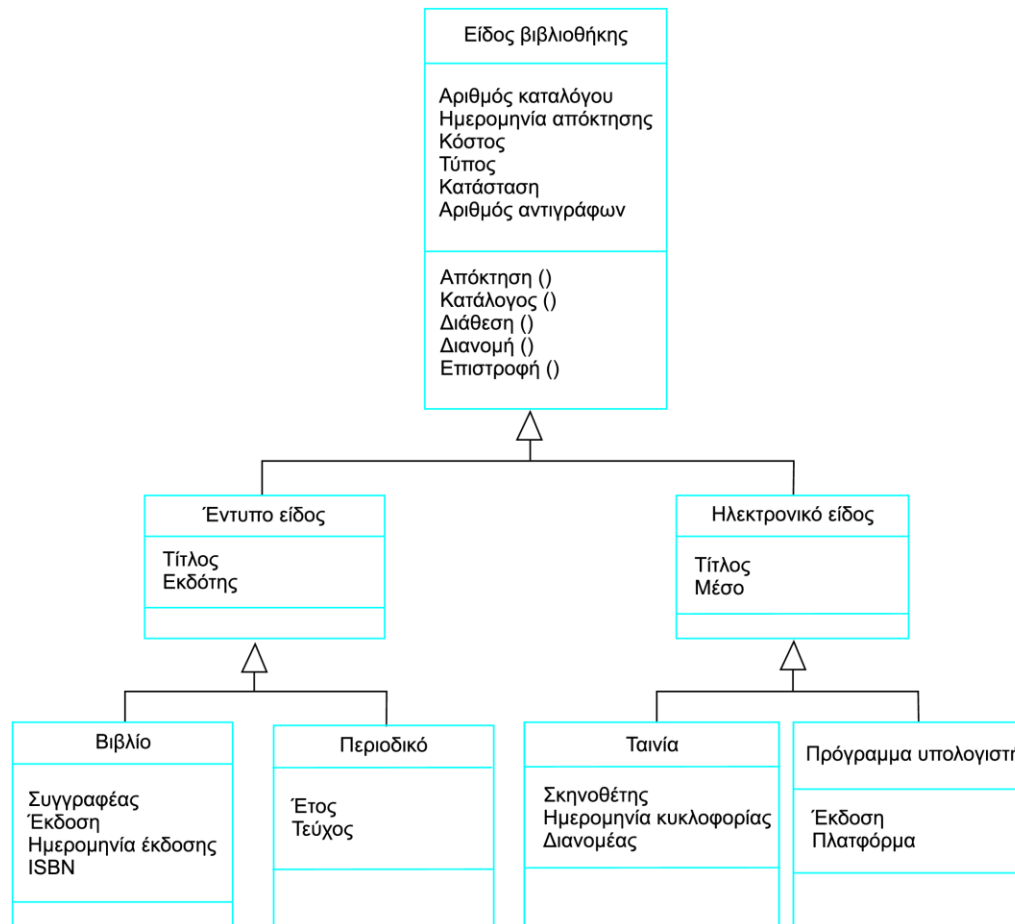
- Κλάση κληρονομεί ιδιότητες & μεθόδους από άλλη (γονική κλάση)

- **Γενίκευση:**

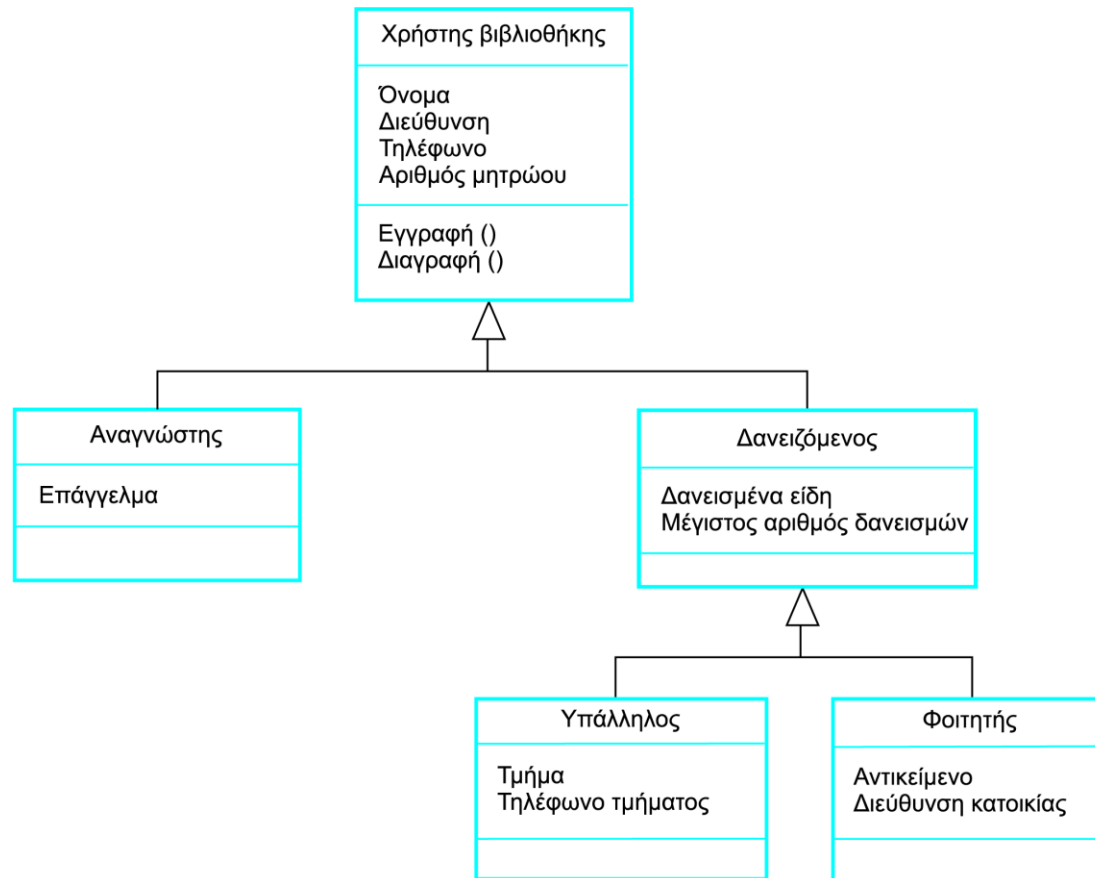
- Το παιδί είναι ειδίκευση του γονέα
 - Μπορεί να εμφανίζεται «αντί» του γονέα
 - Το αντίθετο δεν ισχύει



Ιεραρχία κλάσεων για μια βιβλιοθήκη



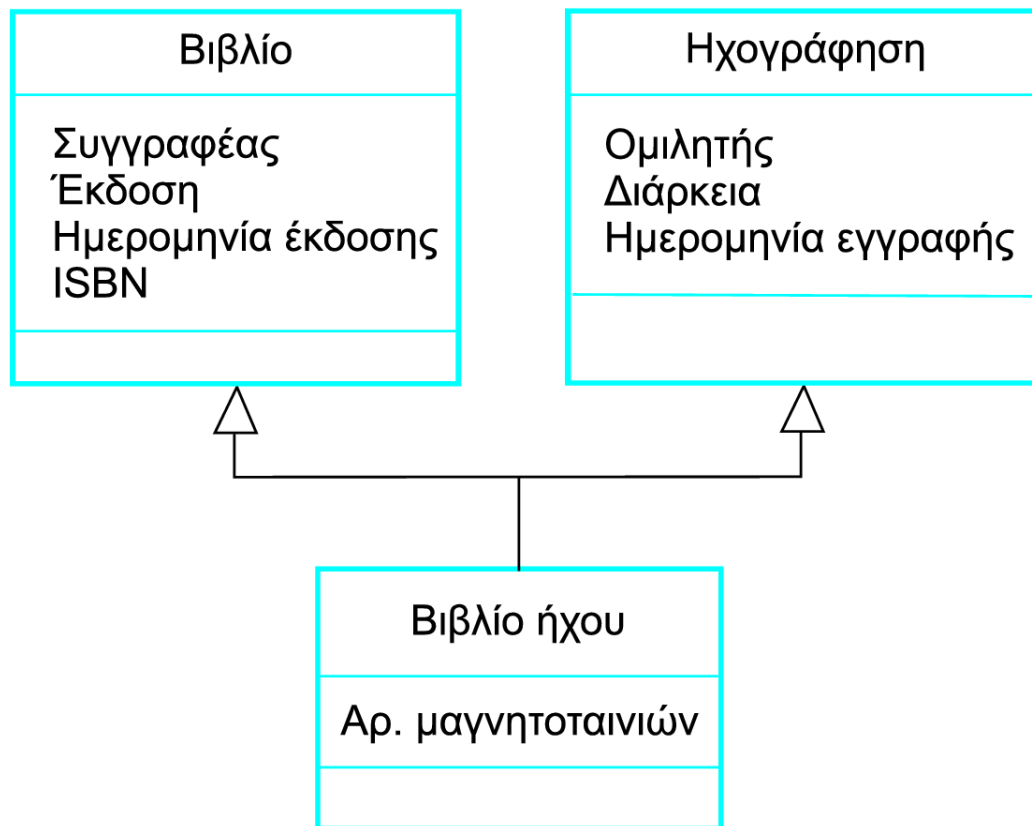
Ιεραρχία κλάσεων χρηστών



Πολλαπλή κληρονομικότητα

- Ένα σύστημα που υποστηρίζει την πολλαπλή κληρονομικότητα επιτρέπει στις κλάσεις αντικειμένων να κληρονομούν γνωρίσματα και υπηρεσίες από πολλές υπερκλάσεις και όχι από μία μόνο γονική κλάση.
- Αυτό μπορεί να οδηγήσει σε σημασιολογικές διενέξεις όταν γνωρίσματα ή υπηρεσίες με το ίδιο όνομα σε διαφορετικές υπερκλάσεις έχουν διαφορετική σημασία.
- Η πολλαπλή κληρονομικότητα περιπλέκει τη διαδικασία αναδιοργάνωσης της ιεραρχίας των κλάσεων.

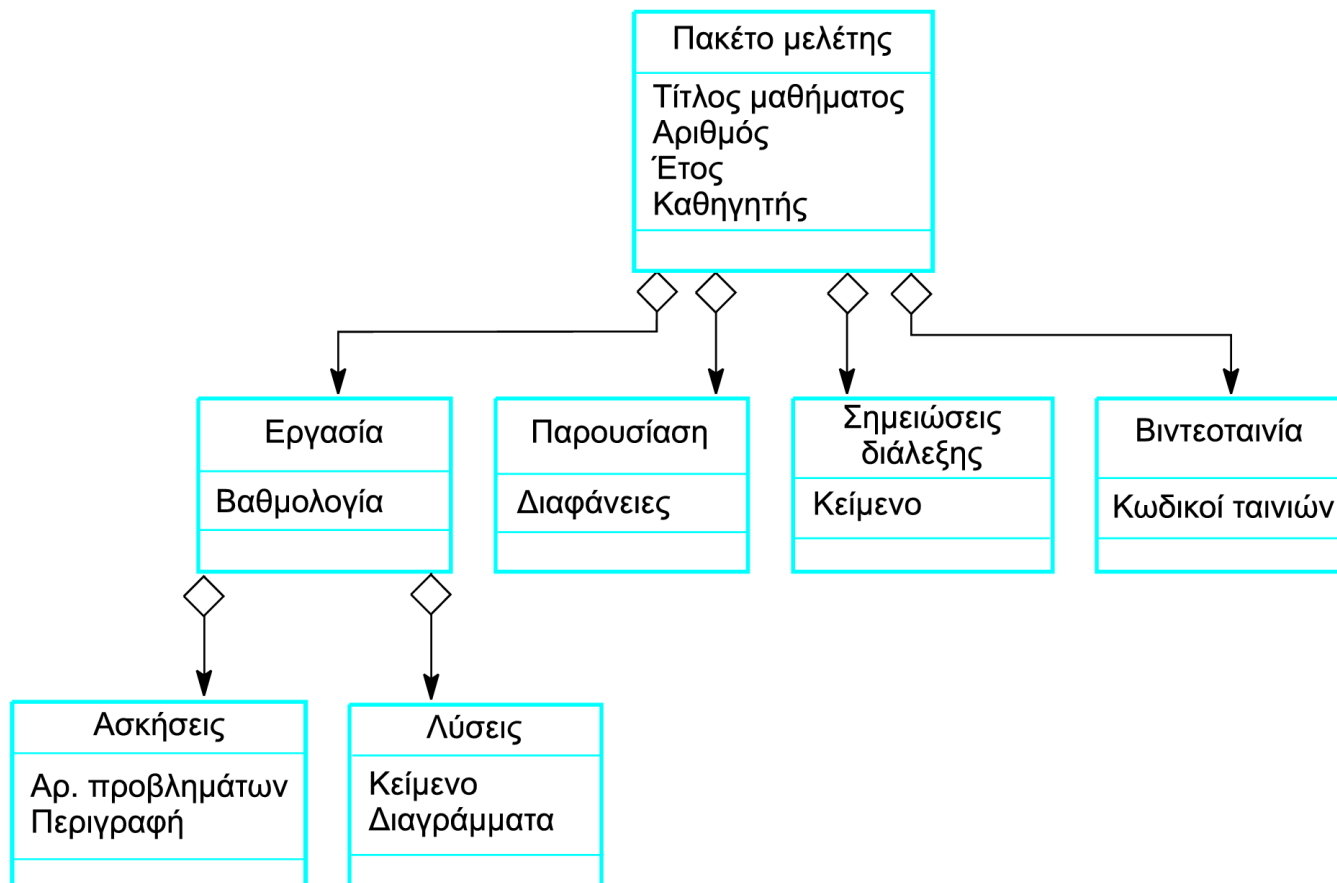
Πολλαπλή κληρονομικότητα



Συνάθροιση αντικειμένων

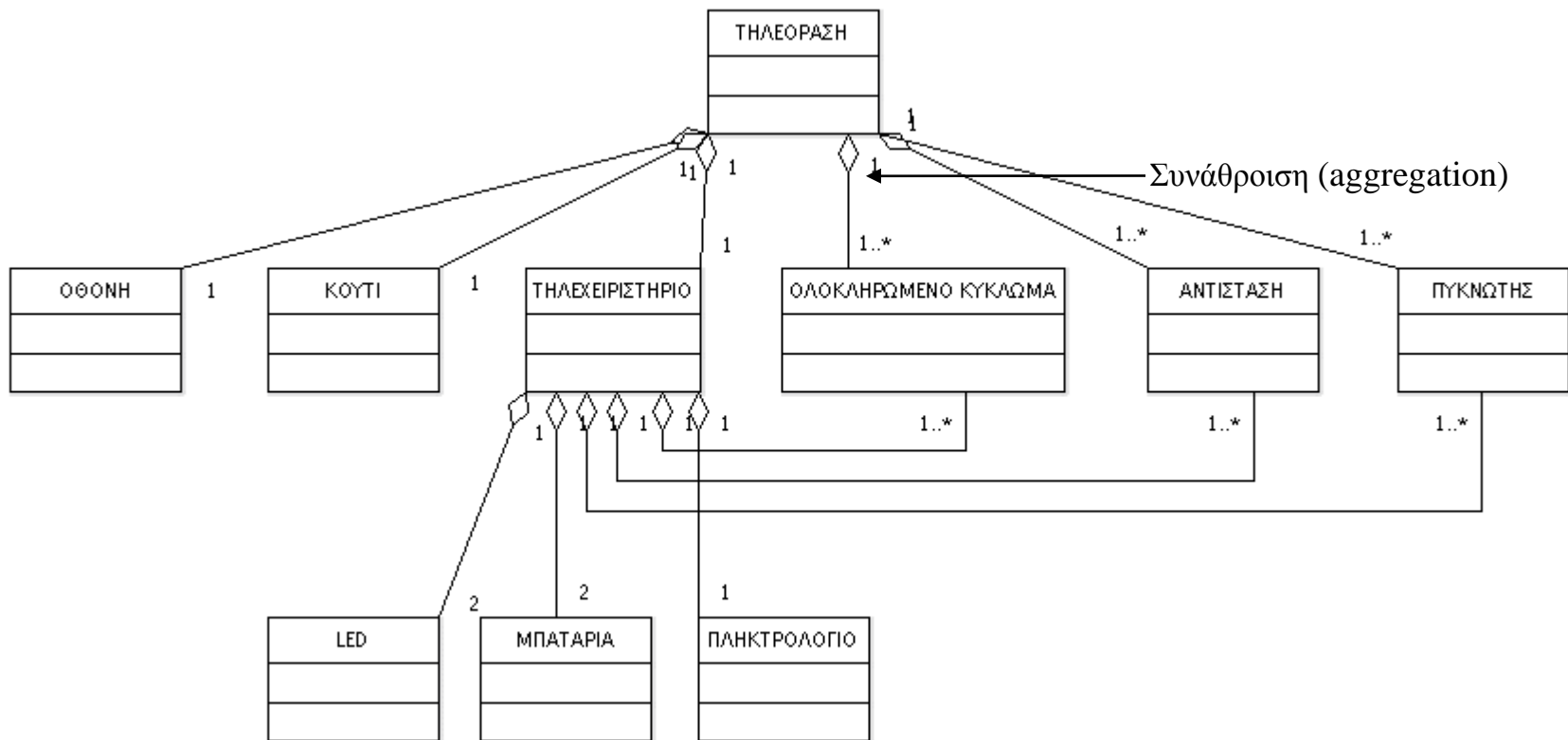
- Τα μοντέλα συνάθροισης δείχνουν πώς οι κλάσεις-συλλογές αποτελούνται από άλλες κλάσεις.
- Τα μοντέλα συνάθροισης μοιάζουν με τη σχέση "ανήκει σε" των σημασιολογικών μοντέλων δεδομένων.

Συνάθροιση αντικειμένων

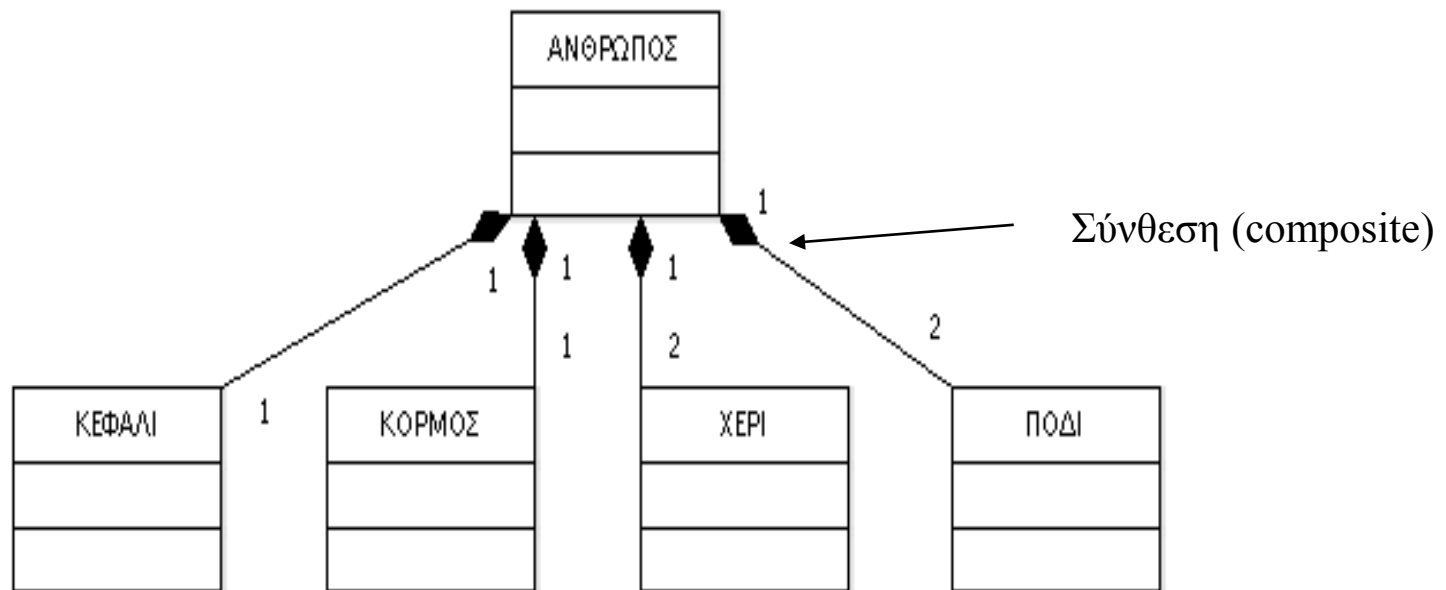


Διαγράμματα Κλάσεων:

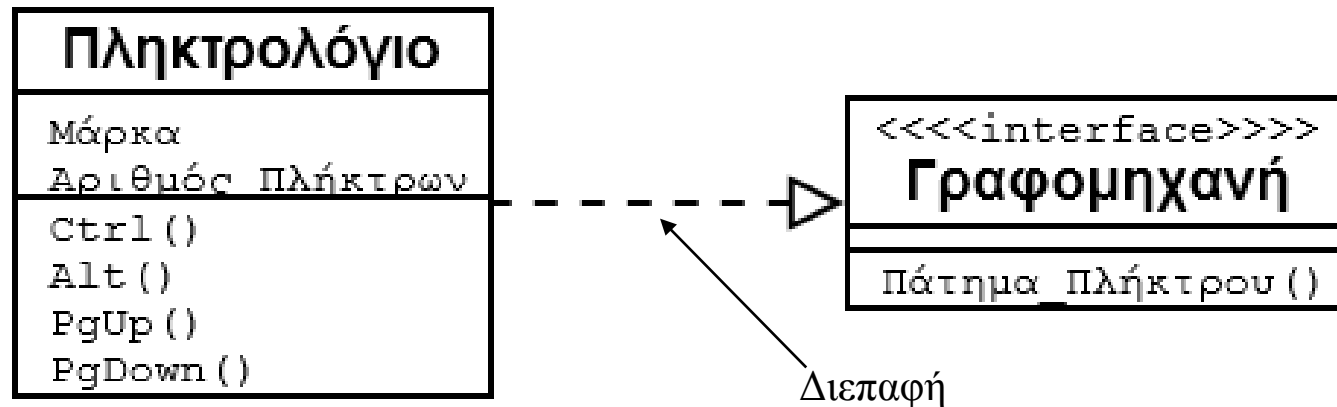
Συναθροίσεις



Συνθέσεις αντικειμένων



Διεπαφές

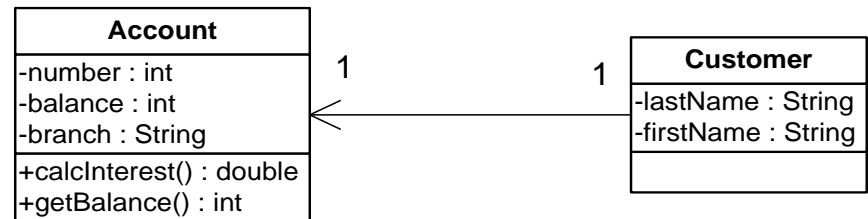


- **Διεπαφή:**
 - Σύνολο μεθόδων
 - Ορίζουν κάποια πλευρά της συμπεριφοράς της κλάσης
 - Είναι διαθέσιμες σε άλλες κλάσεις
 - Προσδιοριστικό `<<interface>>` πάνω από το όνομα της κλάσης
 - Δεν υπάρχουν ιδιότητες

C++ σε UML

Class **Customer**

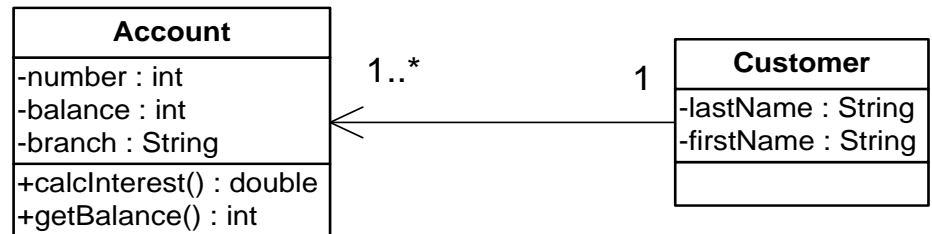
```
{  
    public:  
    Customer();  
    Account* getAccount()  
        {return theAccount;}  
    void setAccount(Account  
        *value)  
        {theAccount=value;}  
    private:  
    string lastName;  
    string firstName;  
    Account* theAccount;  
}
```



C++ σε UML

```
class Customer
{
    public:
        Customer();
        Account* getAccount (int index )
            {return theAccounts[index];}
        void setAccount(int index,
            Account
            *value){relatedAccount=value
            ;}

    private:
        string lastName;
        string firstName;
        Account* theAccounts[];
}
```



C++ σε UML

class **Car**

{

public:

Car();

Engine* getEngine () ;

void setEngine(Engine
*value);

private:

string model;

int serialNo;

Engine* theEngine;

}



C++ σε UML

class **Car**

{

public:

Car();

Engine getEngine () ;

void setEngine(Engine
value);

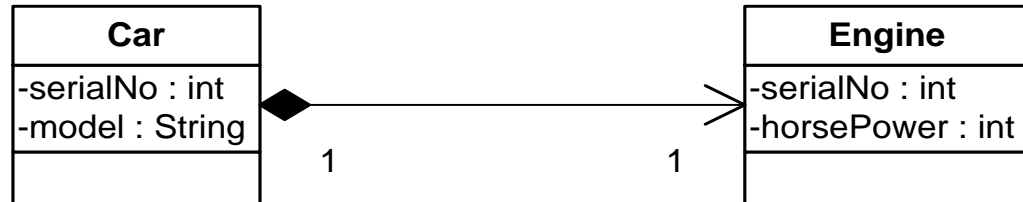
private:

string model;

int serialNo;

Engine theEngine;

}



C++ σε UML

class **Flight**

{

public:

 Flight();

 bool addPassenger (Passenger p) ;

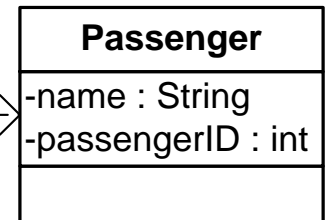
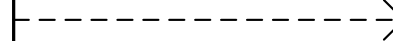
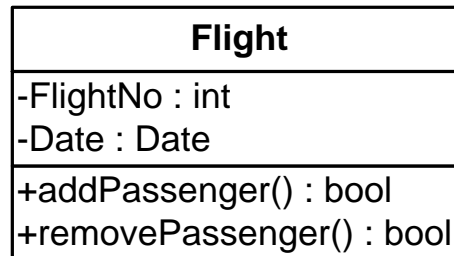
 bool removePassenger (Passenger p);

private:

 int flightNo;

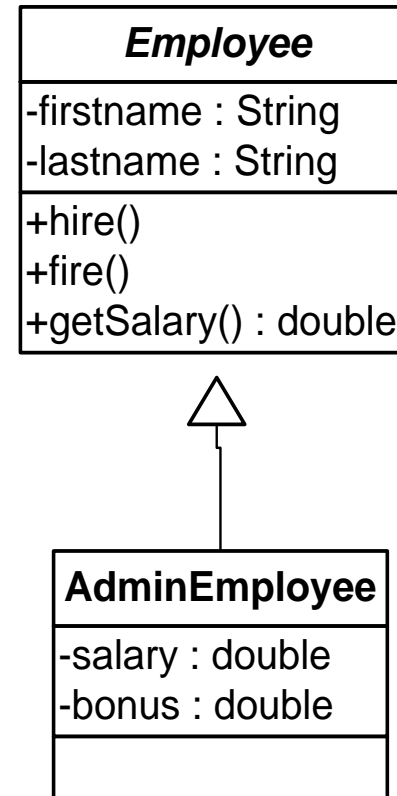
 date flightdate;

}



C++ σε UML

```
class Employee
{
    public:
    Employee();
    void Hire() ;
    void Fire();
    virtual double getSalary();
    private:
    string firstname;
    string lastname;
}
```



C++ σε UML

```
class AdminEmployee:
    public Employee
{
    public:
        AdminEmployee();
        double getSalary();
    private:
        double salary;
        double bonus;
}
```

