
Διαγράμματα Κλάσεων στη Σχεδίαση

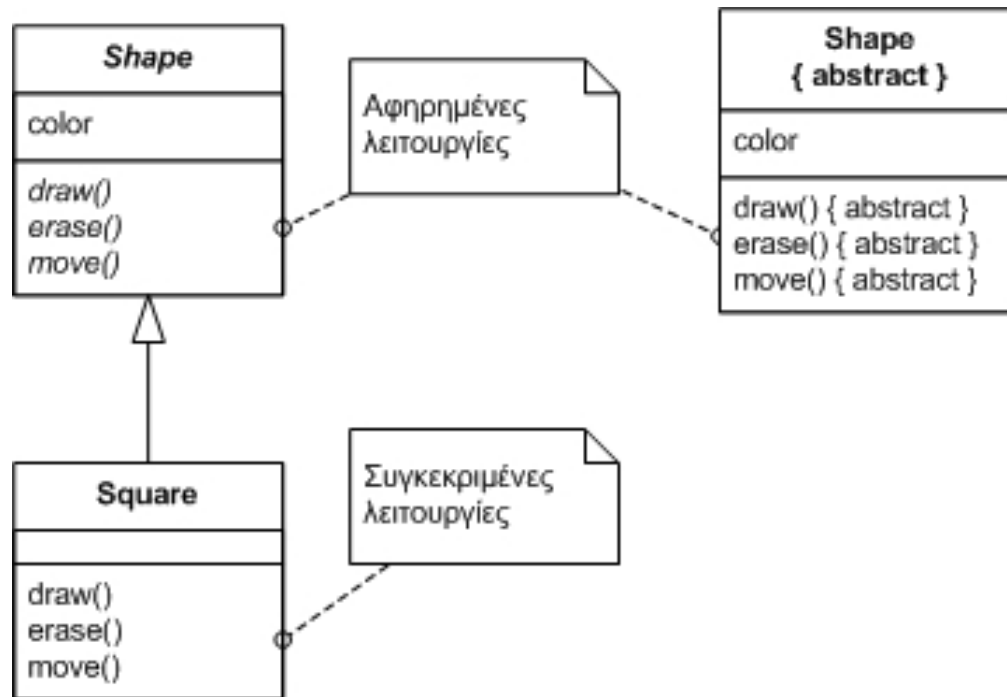
περιεχόμενα παρουσίασης

- Αφηρημένες κλάσεις
- Ιδιότητες
- Λειτουργίες
- Απλοί τύποι
- Συσχετίσεις
- Εξάρτηση
- Διεπαφές

αφηρημένες κλάσεις

- Οι αφηρημένες κλάσεις δεν μπορούν να δημιουργήσουν αντικείμενα και βασίζονται στις υποκλάσεις μέσω της γενίκευσης για τη δημιουργία αντικειμένων.
- Η συνήθης χρήση των αφηρημένων κλάσεων είναι η παροχή της διεπαφής τους στις υποκλάσεις τους.
- Οι λειτουργίες μπορεί να είναι και αυτές αφηρημένες (abstract) , δηλαδή λειτουργίες χωρίς υλοποίηση, ενώ κάποιες άλλες συγκεκριμένες (concrete).
- Η υλοποίηση των αφηρημένων λειτουργιών παρέχεται από τις υποκλάσεις της αφηρημένης κλάσης.

αφηρημένες κλάσεις



ιδιότητες

- Οι ιδιότητες μίας κλάσης αφορούν τα δεδομένα που διατηρούν τα αντικείμενά της.
- Τα δύο νέα στοιχεία των ιδιοτήτων, που είναι χρήσιμα στη σχεδίαση, είναι η **ορατότητα** (visibility) και η **πολλαπλότητα** (multiplicity).
- Έτσι, η πλήρης περιγραφή μίας ιδιότητας με τη UML είναι:
- ορατότητα / όνομα: τύπος [πολλαπλότητα] = αρχική τιμή
{συμβολοσειρά ιδιοτήτων

ορατότητα ιδιοτήτων

Η ορατότητα (visibility) συμβολίζεται ως εξής:

- Με το σύμβολο + που συμβολίζει τη δημόσια (public) ορατότητα. Οι ιδιότητες με δημόσια ορατότητα είναι προσβάσιμες σε όσες κλάσεις έχουν πρόσβαση στην κλάση που τις περιέχει.
- Με το σύμβολο ~ που συμβολίζει ορατότητα πακέτου. Οι ιδιότητες με ορατότητα πακέτου είναι προσβάσιμες στις κλάσεις που είναι στο ίδιο πακέτο με τη συγκεκριμένη κλάση.
- Με το σύμβολο # που συμβολίζει την προστατευμένη (protected) ορατότητα. Οι ιδιότητες με προστατευμένη ορατότητα είναι προσβάσιμες από τις υποκλάσεις και από την κλάση που τις περιέχει.
- Με το σύμβολο – που συμβολίζει την ιδιωτική (private) ορατότητα. Οι ιδιότητες με ιδιωτική ορατότητα είναι προσβάσιμες μόνο από την κλάση που τις περιέχει.

τύποι ιδιοτήτων

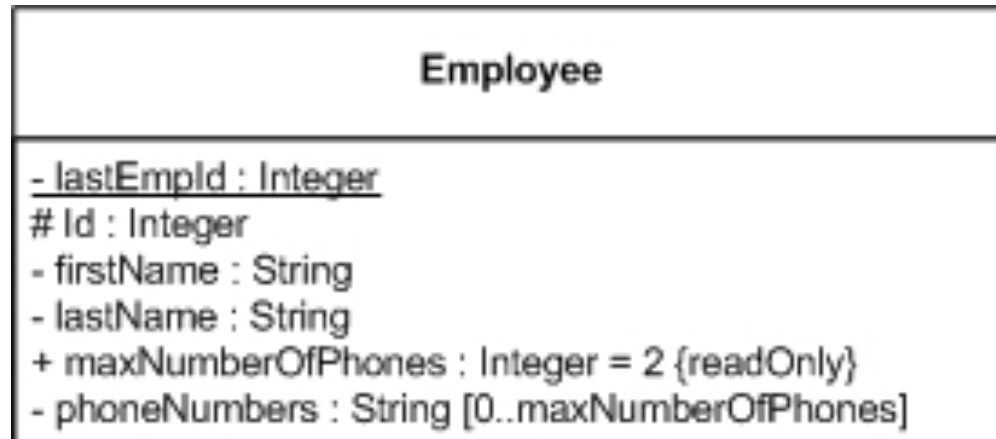


- Ο τύπος μίας ιδιότητας μπορεί να είναι:
 - κάποιος από τους πρωταρχικούς τύπους δεδομένων της UML,
 - κάποιος τύπος της γλώσσας προγραμματισμού ή κάποια
 - άλλη κλάση. Μπορεί επίσης να είναι μία διεπαφή
 - μία απαρίθμηση (enumeration).

ιδιότητες

- Η συμβολοσειρά ιδιοτήτων ρυθμίζει πρόσθετες ιδιότητες της ιδιότητας. Για παράδειγμα, η ένδειξη `readOnly` δείχνει ότι η ιδιότητα είναι μόνο για ανάγνωση και όχι για τροποποίηση.
- Η πολλαπλότητα αναφέρεται στον αριθμό των αντικειμένων της ιδιότητας. Συνήθως οι περισσότερες ιδιότητες παίρνουν μία τιμή. Χρησιμοποιούμε την πολλαπλότητα, εάν θέλουμε να δείξουμε ότι η ιδιότητα λαμβάνει περισσότερες από μία τιμές.
- Εκτός από τις απλές ιδιότητες που αφορούν τα δεδομένα των αντικειμένων μίας κλάσης, με τη UML μπορούμε να ορίσουμε και στατικές ιδιότητες (`static attributes`), οι οποίες ανήκουν στην κλάση και όχι στα αντικείμενά της. Οι στατικές ιδιότητες παραπέμπουν στα στατικά πεδία της Java. Ο συμβολισμός των στατικών ιδιοτήτων είναι όμοιος με τις απλές ιδιότητες με τη διαφορά ότι η σύνταξή τους υπογραμμίζεται.

παράδειγμα: ιδιότητες



Λειτουργίες

- Η σύνταξη μίας λειτουργίας με τη UML είναι:
ορατότητα όνομα (κατάλογος-παραμέτρων): τύπος-επιστροφής
{συμβολοσειρά ιδιοτήτων}
- Η ορατότητα των λειτουργιών συμβολίζεται όμοια με την ορατότητα των ιδιοτήτων. Μία συχνή σύμβαση που ακολουθούν οι μηχανικοί λογισμικού είναι ότι, όταν δεν ορίζεται η ορατότητα της λειτουργίας, τότε θεωρείται δημόσια. Ο κατάλογος παραμέτρων είναι οι παράμετροι της λειτουργίας οι οποίες χωρίζονται με κόμμα.
- Κάθε παράμετρος έχει την ακόλουθη σύνταξη:
κατεύθυνση όνομα: τύπος [πολλαπλότητα] = προκαθορισμένη-τιμή
{συμβολοσειρά ιδιοτήτων}

Λειτουργίες

- Η κατεύθυνση δηλώνει αν μία παράμετρος είναι παράμετρος εισόδου, παράμετρος εξόδου ή και τα δύο, και παίρνει τις τιμές in, out και inout, αντίστοιχα. Όταν η κατεύθυνση δε δηλώνεται, τότε η παράμετρος θεωρείται παράμετρος εισόδου.
- Ο τύπος της παραμέτρου μπορεί να είναι κάποιος από τους πρωταρχικούς τύπους δεδομένων ή κάποια κλάση. Η πολλαπλότητα αφορά στον αριθμό των στιγμιοτύπων για κάθε παράμετρο και συμβολίζεται όμοια με την πολλαπλότητα των ιδιοτήτων. Η προκαθορισμένη τιμή της παραμέτρου είναι η τιμή που παίρνει η παράμετρος, όταν η τιμή δεν παρέχεται κατά την κλήση της λειτουργίας.
- Η συμβολοσειρά ιδιοτήτων προσδίδει στη λειτουργία πρόσθετες ιδιότητες. Μία πιθανή ιδιότητα είναι η ιδιότητα abstract που δηλώνει ότι η λειτουργία είναι αφηρημένη. Μία άλλη είναι η query που δηλώνει ότι η λειτουργία δε μεταβάλλει την κατάσταση των αντικειμένων της κλάσης.

παράδειγμα: λειτουργίες

Employee
<ul style="list-style-type: none">- <u>lastEmpId</u> : Integer# Id : Integer- firstName : String- lastName : String+ maxNumberOfPhones : Integer = 2 {readOnly}- phoneNumbers : String [0..maxNumberOfPhones]
<ul style="list-style-type: none">+ getFirstName() : String {query}+ setFistName(firstName : String)+ getLastName() : String {query}+ setLastName(firstName : String)+ paySalary(month : Month)# calculateSalary(month : Month)+ <u>nextEmployeeId</u>() : Integer

απλοί τύποι



- Προτιμούμε τη χρήση των κλάσεων για δεδομένα με κάποια σημασία, έναντι των πρωταρχικών τύπων όπως οι συμβολοσειρές. Το πλεονέκτημα των κλάσεων είναι ότι παρέχουν ένα επίπεδο αφαίρεσης έναντι των πρωταρχικών τύπων.
- Είναι καλό να διαχωρίζονται τα αντικείμενα τιμών στα διαγράμματα κλάσεων από τα υπόλοιπα αντικείμενα της UML. Η UML δεν παρέχει κάποιο τρόπο να διαχωριστούν τα αντικείμενα τιμές. Η χρήση της λέξης κλειδί «value» στα διαγράμματα κλάσεων είναι ένας τρόπος να διαχωριστούν τα αντικείμενα τιμές, επειδή έχει επίπτωση στην υλοποίησή τους.
- Με τη χρήση της λέξης κλειδί «value» ο σχεδιαστής επικοινωνεί στους προγραμματιστές την πρόθεσή του για χρήση απλών τύπων

απλοί τύποι

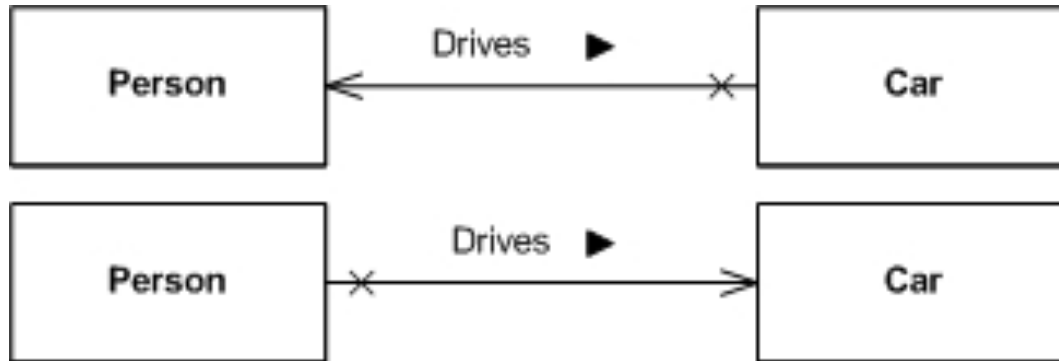
Με την εμφάνιση αντικειμένων τιμών σε ένα διάγραμμα κλάσεων επικοινωνούμε στον προγραμματιστή ότι πρέπει να λάβει υπόψη του τρία θέματα που είναι:

- Η ισότητα των αντικειμένων βασίζεται στις τιμές των ιδιοτήτων τους και όχι στην ταυτότητά τους.
- Οι κλάσεις των αντικειμένων τιμών είναι αμετάβλητες (immutable). Η κατάσταση των αντικειμένων των αμετάβλητων κλάσεων αρχικοποιείται μέσω των κατασκευαστών (constructors) και δεν αλλάζει σε όλο τον κύκλο ζωής των αντικειμένων. (Σε περίπτωση που δεν επιθυμούμε τα αντικείμενα τιμές να προέρχονται από αμετάβλητες κλάσεις, θα πρέπει να είμαστε περισσότερο προσεκτικοί, ιδιαίτερα στον τρόπο που υλοποιούμε τις συσχετίσεις)
- Εάν υπάρχουν πράξεις που θα πρέπει να μεταβάλλουν την κατάσταση των αντικειμένων, τότε και πάλι δεν αλλάζει η κατάσταση του αντικειμένου αλλά επιστρέφονται νέα αντικείμενα.

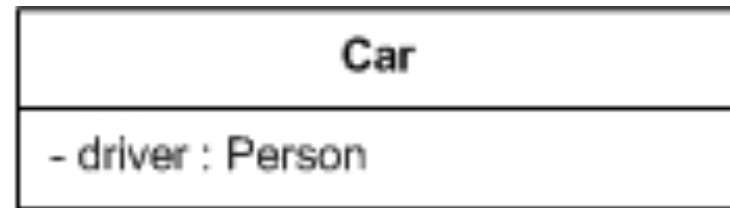
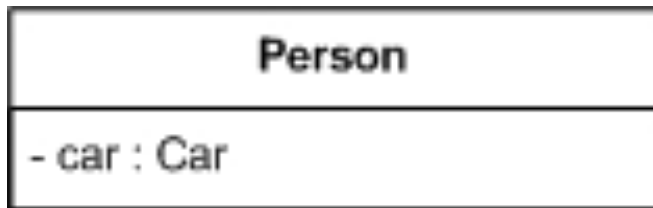
συσχετίσεις

- Γνωρίζουμε ήδη τα βασικά στοιχεία των άκρων μίας συσχέτισης που είναι το όνομα του άκρου και η πολλαπλότητα.
- Ένα τρίτο σημαντικό στοιχείο που μας απασχολεί περισσότερο στη σχεδίαση του λογισμικού είναι η πλοηγησιμότητα (navigability).
- Η πλοηγησιμότητα αναπαριστά το κατά πόσο τα αντικείμενα που συμμετέχουν στα άκρα της συσχέτισης γνωρίζουν την ύπαρξη των αντικειμένων στο άλλο άκρο

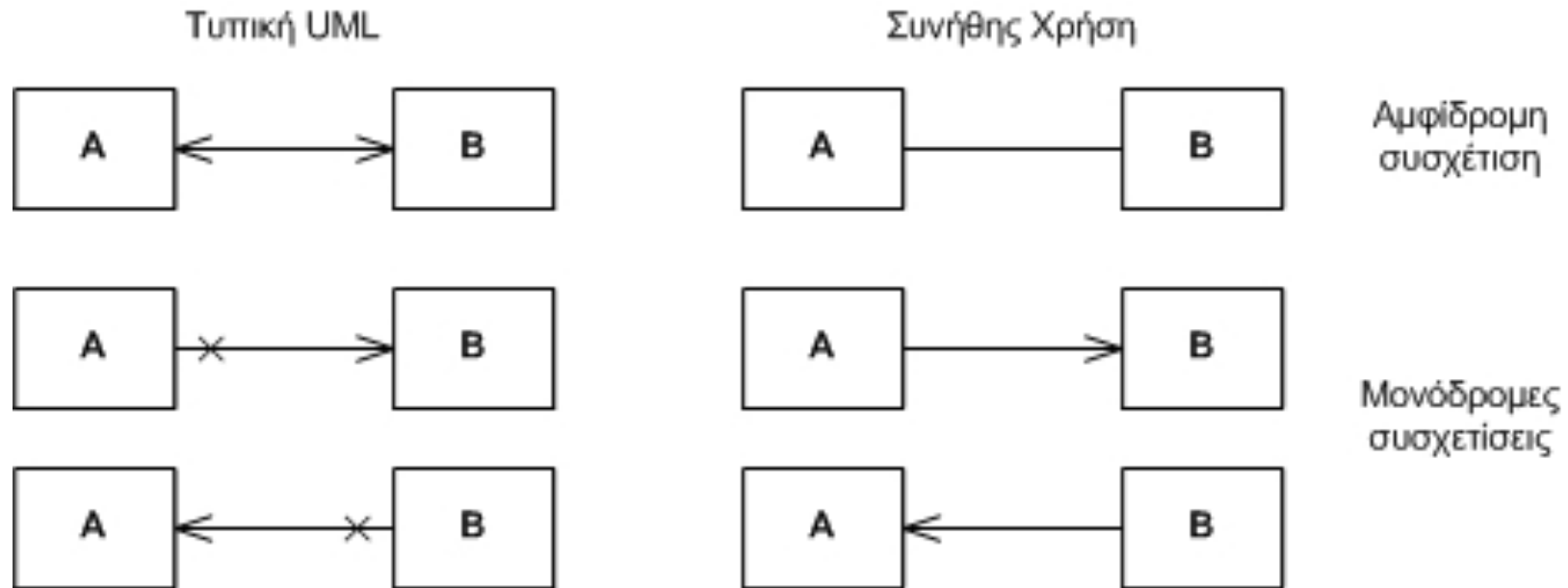
πλοηγησιμότητα συσχετίσεων



ισοδυναμία συσχετίσεων - ιδιοτήτων



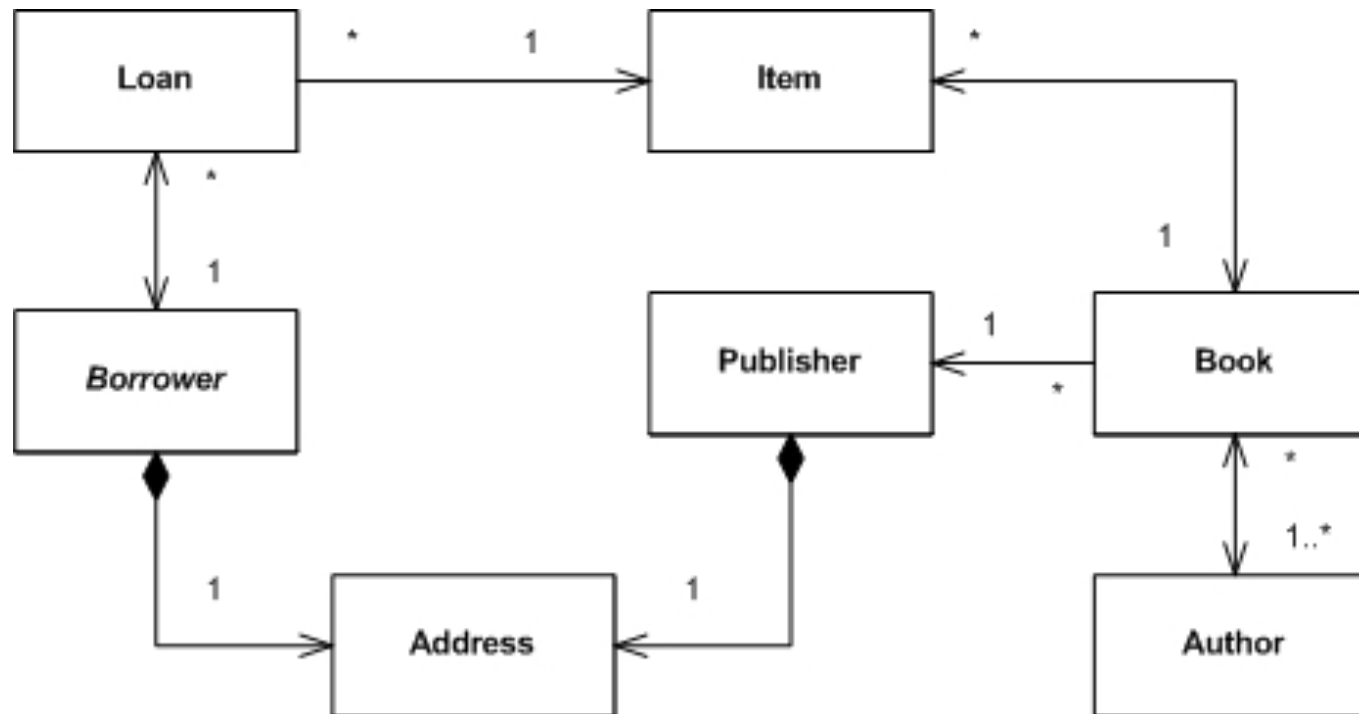
συμβολισμοί πλοηγησιμότητας



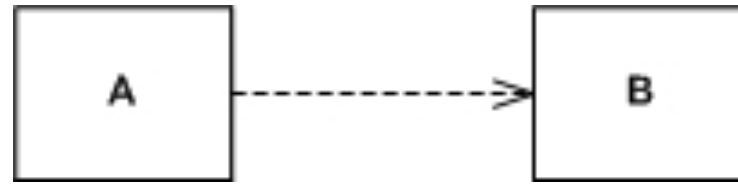
συσχετίσεις

- Στη μοντελοποίηση πεδίου η πλοηγησιμότητα δε μας απασχολεί ιδιαίτερα και μπορεί να θεωρηθεί ότι είναι αμφίδρομη.
- Στη σχεδίαση όμως οι αμφίδρομες συσχετίσεις ορίζονται μόνο όταν αυτές είναι αναγκαίες.
- Η ανάγκη για την πλοηγησιμότητα διαπιστώνεται από την εξέταση της φοράς των μηνυμάτων στα διαγράμματα επικοινωνίας και ακολουθίας, όταν μελετούμε τη συμπεριφορά των αντικειμένων.
- Αν ένα αντικείμενο της κλάσης A πρέπει να στείλει μήνυμα σε αντικείμενο της κλάσης B και υπάρχει συσχέτιση μεταξύ των κλάσεων, τότε θα πρέπει η συσχέτιση να είναι πλοηγήσιμη από την A προς τη B.
- Ο ορισμός της πλοηγησιμότητας υποδηλώνει και την εξάρτηση ή τη σύζευξη μεταξύ των κλάσεων. Όταν τα αντικείμενα μίας κλάσης δε χρειάζεται να γνωρίζουν τα αντικείμενα στο άκρο της συσχέτισης, δεν υπάρχει λόγος να εισάγουμε κάποιου είδους εξάρτηση.

πλοηγησιμότητα στο σύστημα δανεισμού



εξάρτηση



- Η εξάρτηση (dependency) μεταξύ δύο στοιχείων της UML είναι μία γενική σχέση ενός στοιχείου πελάτη, το οποίο εξαρτάται με κάποιο τρόπο με άλλο στοιχείο προμηθευτή.
- Η εξάρτηση σημαίνει ότι κάποια αλλαγή στον προμηθευτή επηρεάζει με κάποιο τρόπο τον πελάτη.
- Έχουμε ήδη συναντήσει ισχυρές μορφές εξάρτησης μεταξύ κλάσεων που είναι η συσχέτιση και η γενίκευση. Δεν είναι όμως οι μόνες περιπτώσεις από τις οποίες προκύπτει μία σχέση εξάρτησης μεταξύ δύο κλάσεων.
- Άλλες περιπτώσεις εξάρτησης είναι όταν ένα αντικείμενο της κλάσης πελάτη αποστέλλει μήνυμα σε αντικείμενο της κλάσης προμηθευτή ή όταν κάποια λειτουργία της κλάσης πελάτη έχει μία παράμετρο με τύπο την κλάση προμηθευτή.
- Με όρους κώδικα: η διαγραφή της κλάσης B από τον κώδικα προκαλεί σφάλμα μεταγλώττισης στην κλάση A.

Διεπαφές

- Οι διεπαφές (interfaces) είναι μηχανισμοί των σύγχρονων γλωσσών προγραμματισμού όπως η Java για την επίτευξη πολυμορφισμού.
- Μία διεπαφή μπορεί να θεωρηθεί ως μία κλάση χωρίς πεδία όπου όλες οι πράξεις είναι αφηρημένες.
- Οι διεπαφές της Java μοιάζουν με τις αφηρημένες κλάσεις, αλλά παρουσιάζουν ορισμένες σημαντικές διαφορές που είναι:
- Οι αφηρημένες κλάσεις μπορεί να έχουν πεδία, ενώ οι διεπαφές όχι.

διεπαφές

- Οι αφηρημένες κλάσεις μπορεί να έχουν αφηρημένες λειτουργίες, δηλαδή χωρίς υλοποίηση, αλλά μπορεί να έχουν και συγκεκριμένες λειτουργίες που κληρονομούνται από τις υποκλάσεις. Οι διεπαφές δηλώνουν τις υπογραφές των λειτουργιών, χωρίς να παρέχουν καμία υλοποίηση.
- Οι αφηρημένες κλάσεις δηλώνουν κατασκευαστές, ενώ οι διεπαφές όχι.
- Η ορατότητα μίας διεπαφής για τη Java μπορεί να είναι μόνο δημόσια.
- Οι λειτουργίες μίας αφηρημένης κλάσης μπορεί να έχουν ορατότητα προστατευμένη ή ιδιωτική, ενώ οι λειτουργίες που δηλώνει μία διεπαφή έχουν δημόσια ορατότητα

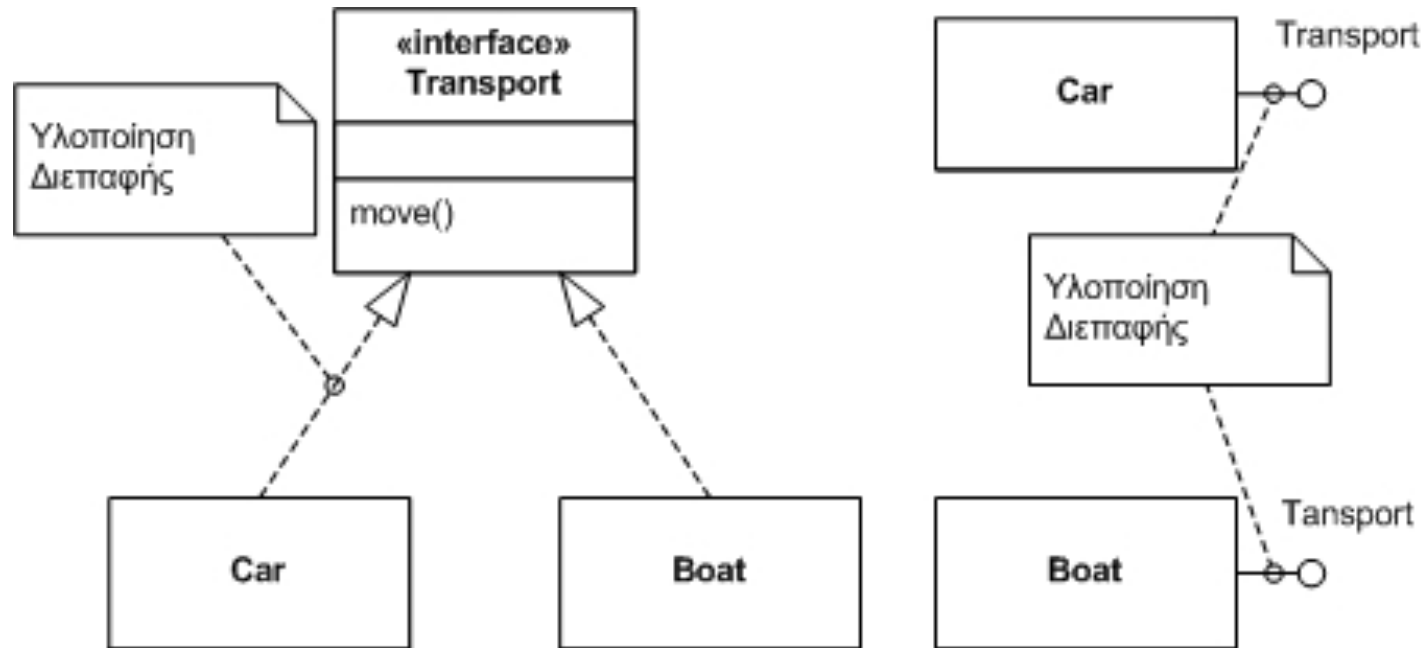
διεπαφές

- Η υποκλάση μίας αφηρημένης κλάσης κληρονομεί τη δημόσια διεπαφή και την υλοποίηση που ίσως η αφηρημένη κλάση παρέχει, έχοντας ταυτόχρονα τη δυνατότητα επαναορισμού (overriding) των μεθόδων της.
- Μία κλάση μπορεί να υλοποιεί (και δεν κληρονομεί) μία ή περισσότερες διεπαφές.
- Η κλάση θα πρέπει να παρέχει την υλοποίηση για τις μεθόδους που δηλώνει μία διεπαφή, να παρέχει δηλαδή την υλοποίηση της διεπαφής.

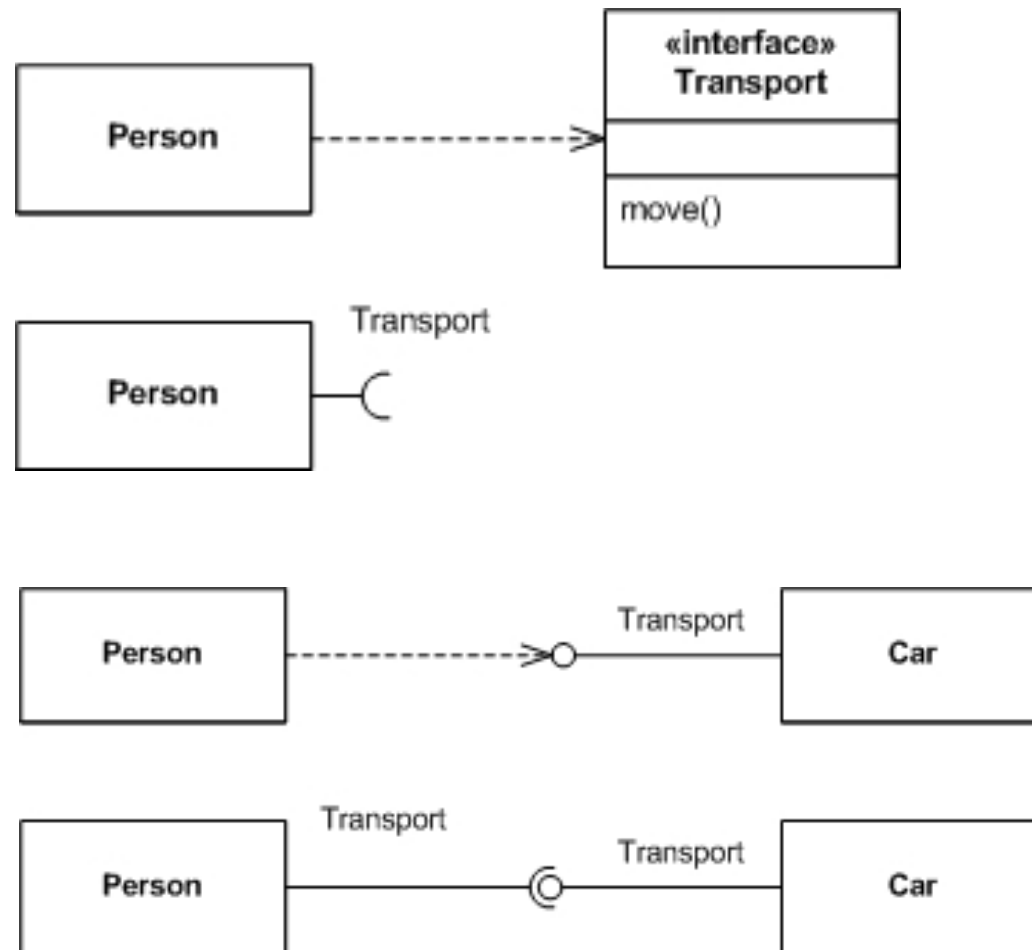
διεπαφές

- Επομένως, μία διεπαφή έχει περισσότερο το χαρακτήρα μίας δήλωσης. Δηλώνει υπογραφές (signatures) μεθόδων στις οποίες μία κλάση θα πρέπει να συμμορφώνεται σε περίπτωση που υλοποιεί τη διεπαφή.
- Οι διεπαφές είναι και το βασικό αντίδοτο για τις γλώσσες προγραμματισμού που δεν υποστηρίζουν την πολλαπλή κληρονομικότητα. Μία κλάση μπορεί να είναι υποκλάση μίας και μόνο κλάσης, ενώ μπορεί να υλοποιεί πολλές διεπαφές.

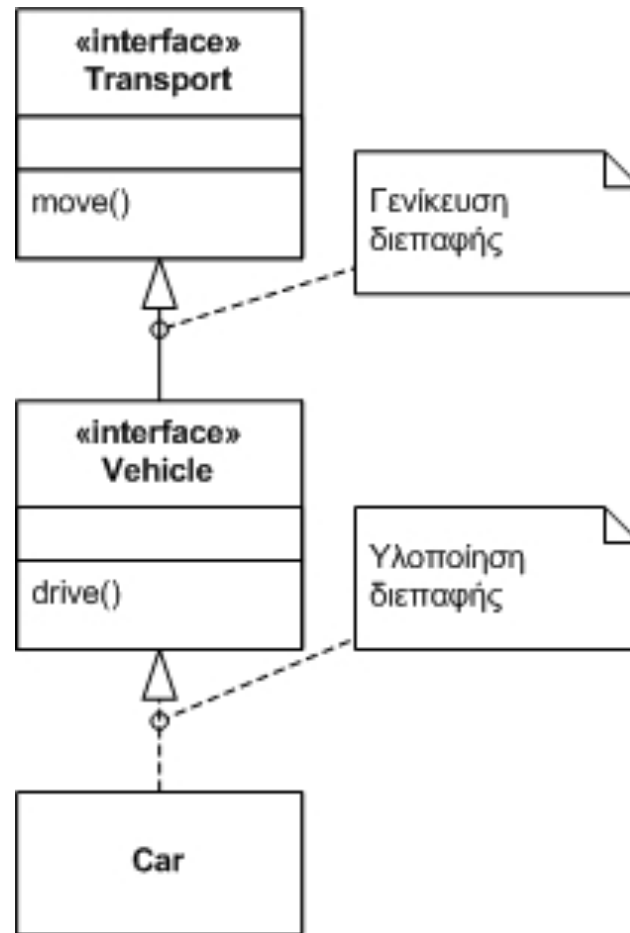
διεπαφές στη UML



παροχή και χρήση διεπαφής



γενίκευση και υλοποίηση διεπαφών



διεπαφές και αφηρημένες κλάσεις

