

Class Diagrams (διαγράμματα κλάσεων ή τάξεων) και προδιαγραφές

Γενικά

Ένα διάγραμμα τάξεων περιγράφει τις κλάσεις του συστήματος ως ένα στατικό μοντέλο.

Σε ένα διάγραμμα τάξεων διακρίνουμε:

- Τάξεις (classes):** Είναι οι τάξεις ή κλάσεις του συστήματος
- Σχέσεις (relationships):** Περιγράφουν τις σχέσεις ανάμεσα στις κλάσεις

Διαγραμματικά Στοιχεία

Τα βασικά διαγραμματικά στοιχεία που συναντάμε σε ένα διάγραμμα τάξεων είναι:

Τάξη (class)

Είναι ένα ορθογώνιο σχήμα που χωρίζεται σε τρία οριζόντια διαμερίσματα. Στο επάνω καταχωρείται το όνομα της τάξης. Στο μεσαίο οι ιδιότητες και στο τελευταίο οι λειτουργίες της κλάσης.

| ClassName |
|------------|
| attributes |
| operations |

Σχέσεις (relationships)

Είναι μια γραμμή που ενώνει δύο κλάσεις. Ανάλογα το είδος της συσχέτισης μπορεί να έχει διαφορετικές μύτες (βέλη) ή να είναι διακεκομμένη.

Δευτερεύοντα διαγραμματικά στοιχεία

Ως δευτερεύοντα διαγραμματικά στοιχεία μπορούμε να έχουμε:

Ρόλος

Είναι μία ετικέτα που μπαίνει στο πέρας της γραμμής μιας σχέσης.

Πολλαπλότητα

Είναι μια ένδειξη για το πόσα αντικείμενα μπορούν να συμμετέχουν σ' αυτή τη σχέση.

Πλοήγηση

Δείχνει ότι υπάρχει δυνατότητα πλοήγησης από την μια κατεύθυνση αλλά όχι από την άλλη. Αν για παράδειγμα μπορούμε να βρούμε σε ποια μαθήματα διδάσκει ένας καθηγητής ξεκινώντας από ένα αντικείμενο της τάξης Professor, αλλά όχι το αντίστροφο θα έχουμε δυνατότητα πλοήγησης μόνο από την τάξη Professor προς την τάξη Course. Συμβολίζεται με βέλος.

Περιγραφή τάξεων (classes)

Κάθε τάξη περιέχει **Ιδιότητες** (attributes) και **Λειτουργίες** (operations).

Οι ιδιότητες δίνουν τα χαρακτηριστικά της τάξης ενώ οι τάξεις περιγράφουν τις λειτουργίες που διαθέτουν και το αποτέλεσμα που επιστρέφουν.

Σε όρους java, η ιδιότητα αντιστοιχεί στο πεδίο (field) της τάξης και η λειτουργία στη μέθοδο (method) της τάξης. Ιδιότητες και λειτουργίες μαζί αποτελούν τα μέλη της τάξης.

Όπως αναφέρθηκε, τό όνομα της κλάσης τοποθετείται στό επάνω μέρος του ορθογωνίου με έντονα γράμματα.

Παράδειγμα

| ClassName |
|------------|
| attributes |
| operations |

Περιγραφή ιδιοτήτων

Κάθε ιδιότητα δηλώνεται με το όνομά της και τον τύπο της. Για παράδειγμα: `userid : integer`

Κάθε ιδιότητα ανάλογα την προσβασιμότητά της από εξωτερικά αντικείμενα μπορεί να είναι:

1. Ιδιωτική (private) και απεικονίζεται με το σύμβολο (-). Για παράδειγμα: `- userid : integer`
2. Δημόσια (public) και απεικονίζεται με το σύμβολο (+). Για παράδειγμα: `+ userid : integer`
3. Προστατευόμενη (protected) και απεικονίζεται με το σύμβολο (#). Για παράδειγμα: `# userid : integer`

Παράδειγμα

| Hotel |
|--|
| - name : String - address : String - numberOfRooms : integer |
| |

Περιγραφή λειτουργιών

Κάθε λειτουργία δηλώνεται με το ονομά της, τις παραμέτρους (αριθμός και τύποι) που έχει και τον τύπο δεδομένων που επιστρέφει.

Επίσης κάθε λειτουργία μπορεί να έχει δηλωθεί ως private, public ή protected και χρησιμοποιούνται τα ίδια σύμβολα όπως και στις ιδιότητες.

Μία ή περισσότερες λειτουργίες μπορεί να λειτουργούν ως δημιουργοί (constructors). Στην περίπτωση αυτή θα φέρουν το όνομα της κλάσης.

Παράδειγμα

| Hotel |
|--|
| - name : String - address : String - numberOfRooms : integer |
| + Hotel():void + getName():String + getAddress():String + getNumberOfRooms():Integer + setName(name:String):void + setAddress(address:String):void + setnumOfRooms(num:Integer):void |

Static μέλη

Αν η ιδιότητα (ή η μέθοδος) είναι static, τότε απεικονίζεται με υπογράμμιση (π.χ.: numberOfRooms:integer).

Read only

Αν η ιδιότητα είναι read only, τότε προστίθεται το στερεότυπο {readOnly} (π.χ.: numberOfRooms:integer {readOnly}).

Σχέσεις (relationships)

Οι τύποι σχέσεων μεταξύ των τάξεων μπορεί να είναι:

Συσχετίσεις (associations)

Απλές και μόνιμες (σημασιολογικές) σχέσεις μεταξύ των τάξεων.

Για παράδειγμα μια τάξη Πελάτη (Customer) μιας τράπεζας έχει μόνιμη συσχέτιση με έναν ή περισσότερους Τραπεζικούς Λογαριασμούς (Account).

Κλασική περίπτωση όταν σε μια τρίτη κλάση δημιουργούνται ή/και χρησιμοποιούνται τα συσχετισμένα αντικείμενα.



Παράδειγμα σε κώδικα java

```
class Customer {
    ...
}
class Account {
    ...
}
class BankAccount {
    ...
    Customer c = new Customer();
    Account a = new Account();
    ...
}
```

Εξαρτήσεις (dependencies)

Συνήθως εκφράζουν προσωρινές αλληλεπιδράσεις μεταξύ των αντικειμένων κάποιων τάξεων κατά τη διάρκεια εκτέλεσης ενός προγράμματος αλλά και την εξάρτηση μιας τάξης από μια άλλη.

Ειδικότερα, μπορούμε να πούμε ότι έχουμε εξάρτηση όταν μια μέθοδος μιας κλάσης καλεί μια μέθοδο μιας άλλης κλάσης ή όταν μια μέθοδος μια κλάσης δέχεται ως όρισμα αντικείμενο μιας άλλης κλάσης.

Η εξάρτηση συμβολίζεται με βέλος και διακεκομμένη γραμμή όπως στο πιο κάτω παράδειγμα.



Εδώ, η κλάση A εξαρτάται από την κλάση B.

Επίσης, σαν ένα δεύτερο παράδειγμα, σε ένα σύστημα πληρωμών μια τάξη πληρωμής λογαριασμού (Pay) καλεί μια τάξη ελέγχου εγκυρότητας πιστωτικής κάρτας (Card).



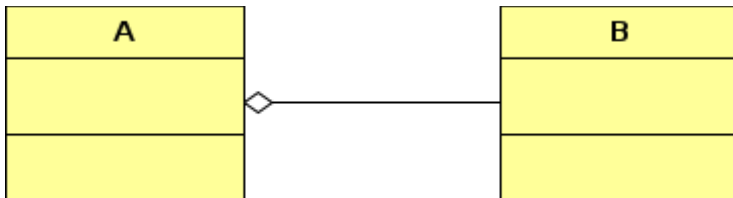
Εδώ πάλι υπάρχει μια εξάρτηση της πρώτης κλάσης από τη δεύτερη.

Συσσωματώσεις (aggregations)

Είναι μία ειδική μορφή συσχέτισης. Είναι μία συσχέτιση που συσχετίζει το "όλον" με το "μέρος" ή "τμήμα".

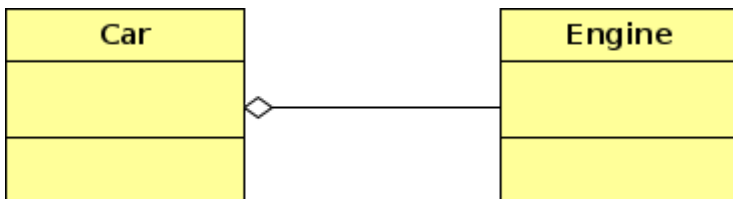
Η διάκριση από τη συσχέτιση έχει περισσότερο εννοιολογικό χαρακτήρα: το "όλον" δεν μπορεί να "λειτουργήσει" χωρίς το "μέρος".

Η συσσωμάτωση συμβολίζεται με βέλος σε σχήμα ρόμβου όπως στο σχήμα.



Εδώ, η κλάση A συσσωματώνει την κλάση B.

Επίσης, σαν ένα δεύτερο παράδειγμα, η κλάση (Car) συσσωματώνει την κλάση (Engine).



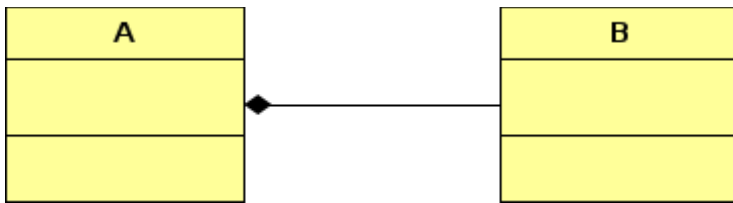
Παράδειγμα σε κώδικα java

```
class A {
    List items;
    public void AddItem(B newItem) {
        items.Add(newItem);
        ...
    }
    ...
}
class B {
    ...
}
```

Συνθέσεις (Compositions)

Είναι μία ισχυρή μορφή συσσωμάτωσης. Το "όλον" ελέγχει πλήρως τα "μέρη" του και ιδιαίτερα τον κύκλο ζωής τους. Π.χ. η διαγραφή ενός αντικειμένου "όλου" διαγράφει και τα "μέρη" του. Η πρόσβαση σε κάποιο "μέρος" γίνεται αποκλειστικά μέσω του "όλου".

Η σύνθεση συμβολίζεται με βέλος σε σχήμα μαύρου ρόμβου όπως στο σχήμα.



Παράδειγμα σε κώδικα java

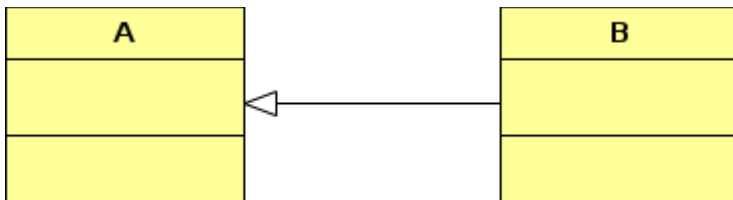
```

class A {
    B b = new B(); // Η A δημιουργεί αποκλειστικά αντικείμενα B
    ...
}
class B {
    ...
}
  
```

Γενικεύσεις (generalizations)

Σε αυτές τις σχέσεις μια υπο-τάξη είναι πιο ειδική από την υπερ-τάξη η οποία είναι πιο γενική.

Συμβολίζεται με βέλος σε σχήμα τριγώνου όπως στο πιο κάτω παράδειγμα.



Εδώ η κλάση B εξειδικεύει την πιο γενική κλάση A.

Παράδειγμα σε κώδικα java

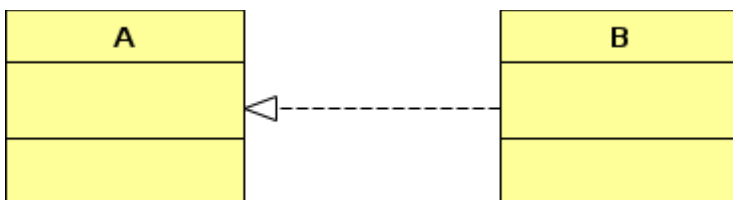
```

class A {
    ...
}
class B extends A {
    ...
}
  
```

Υλοποιήσεις ή πραγματοποιήσεις (realizations)

Εδώ, μια κλάση υλοποιεί μια άλλη (interface)

Συμβολίζεται με βέλος σε σχήμα τριγώνου και διακεκομμένη γραμμή όπως στο πιο κάτω παράδειγμα.



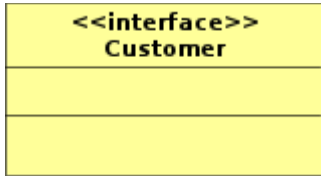
Παράδειγμα σε κώδικα java

```

interface A {
    ...
}
class B implements A {
    ...
}
  
```

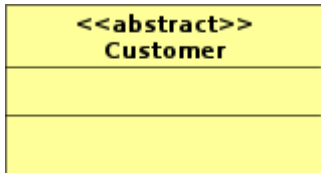
Διεπαφή (interface)

Για τις διεπαφές ο συμβολισμός γίνεται όπως παρακάτω:



Αφηρημένη κλάση (abstract)

Επίσης για τις αφηρημένες κλάσεις ο συμβολισμός είναι παρόμοιος.



Σημείωση

Οι μέθοδες τόσο στις διεπαφές όσο και στις αφηρημένες κλάσεις συμβολίζονται με πλάγια γράμματα.