ΤΕΧΝΙΚΕΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Κλάσεις και Αντικείμενα

Η εξέλιξη των γλωσσών προγραμματισμού

- Η εξέλιξη των γλωσσών προγραμματισμού είναι μια διαδικασία αφαίρεσης
  - Στην αρχή ένα πρόγραμμα ήταν μια σειρά από εντολές σε γλώσσα μηχανής.
  - Με τον Διαδικασιακό Προγραμματισμό (procedural programming), ένα πρόγραμμα έγινε μια συλλογή από διαδικασίες που η μία καλεί την άλλη.
  - Στον Συναρτησιακό Προγραμματισμό (functional programming) ένα πρόγραμμα είναι μια συλλογή από συναρτήσεις όπου η μία εφαρμόζεται πάνω στην άλλη.
  - Στον Λογικό Προγραμματισμό (logic programming) ένα πρόγραμμα είναι μια συλλογή από κανόνες και γεγονότα.
  - Στον Αντικειμενοστραφή Προγραμματισμό (object oriented programming) ένα πρόγραμμα είναι μια συλλογή από κλάσεις και αντικείμενα όπου το ένα μιλάει με το άλλο

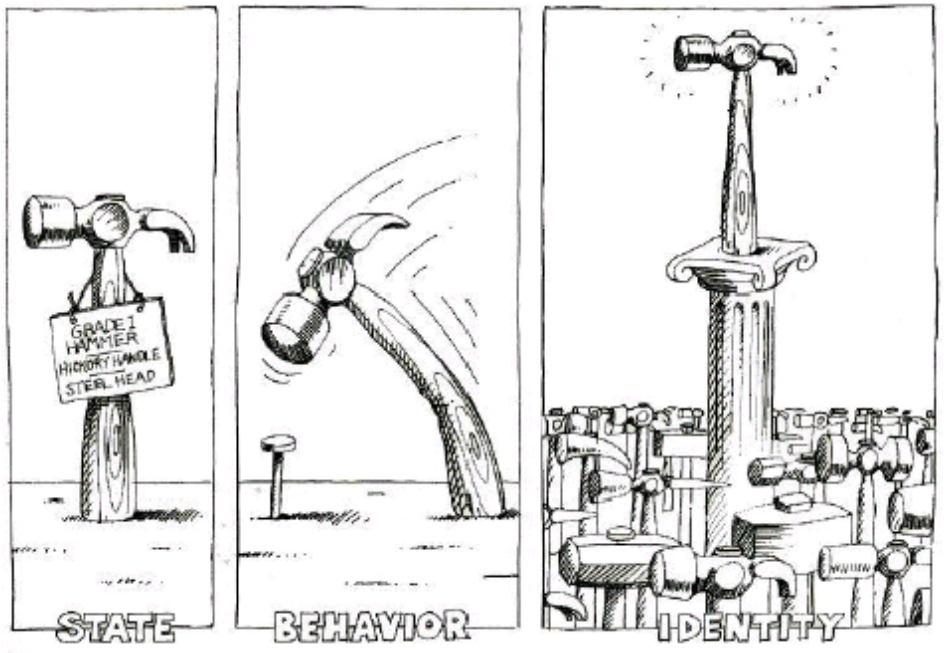
Αντικειμενοστραφής Προγραμματισμός

- Οι πέντε αρχές του Allan Kay:
  - Τα πάντα είναι αντικείμενα.
  - Ένα πρόγραμμα είναι μια συλλογή από αντικείμενα όπου το ένα λέει στο άλλο τι να κάνει.
  - Κάθε αντικείμενο έχει δικιά του μνήμη και αποτελείται από άλλα αντικείμενα.
  - Κάθε αντικείμενο έχει ένα συγκεκριμένο τύπο.
    - Τύπος = Κλάση
  - Αντικείμενα του ίδιου τύπου μπορούν να δεχτούν τα ίδια μηνύματα
    - Δηλαδή έχουν τις ίδιες λειτουργίες

Αντικείμενο

- Ένα αντικείμενο στον κώδικα αναπαριστά μια μονάδα/οντότητα/έννοια η οποία έχει:

- Μια κατάσταση, η οποία ορίζεται από ορισμένα χαρακτηριστικά
- Μια συμπεριφορά, η οποία ορίζεται από ορισμένες ενέργειες που μπορεί να εκτελέσει το αντικείμενο
- Μια ταυτότητα που το ξεχωρίζει από τα υπόλοιπα.



Παραδείγματα: ένας άνθρωπος, ένα πράγμα, ένα μέρος, μια υπηρεσία

Κλάση

- Μια κλάση είναι μία αφηρημένη περιγραφή αντικειμένων με κοινά χαρακτηριστικά και κοινή συμπεριφορά.
  - Ένα καλούπι/πρότυπο που παράγει αντικείμενα
  - Ένα αντικείμενο είναι ένα στιγμιότυπο μίας κλάσης.

- Η κλάση ορίζει τον τύπο του αντικειμένου.
  - Τα χαρακτηριστικά του αντικειμένου
  - Τις ενέργειες που μπορεί να επιτελέσει.

- Παράδειγμα
  - Η κλάση ipod ορίζει μια γενική περιγραφή που περιλαμβάνει:
    - Τα χαρακτηριστικά: μέγεθος, μνήμη, χρώμα
    - Τις ενέργειες: on, off, play
  - Το αντικείμενο ipod είναι ένα συγκεκριμένο φυσικό αντικείμενο
    - Αυτό του δίνει συγκεκριμένη ταυτότητα.

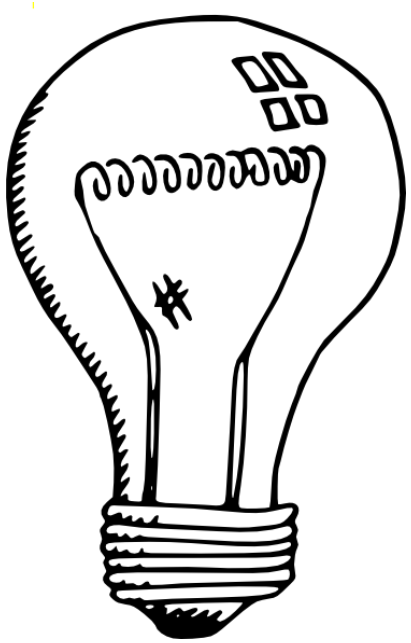
Πρακτικά στον κώδικα

- Μία κλάση Κ ορίζεται από
  - Κάποιες μεταβλητές τις οποίες ονομάζουμε πεδία
  - Κάποιες συναρτήσεις που τις ονομάζουμε μεθόδους.
    - Οι μέθοδοι «βλέπουν» τα πεδία της κλάσης

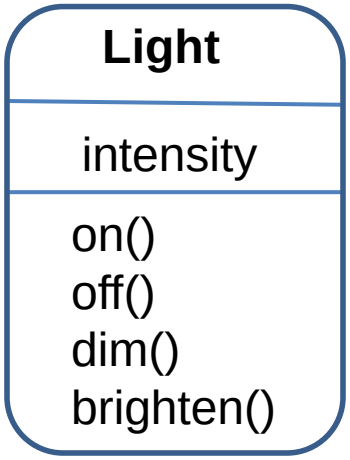
μέλη της κλάσης

- Ένα αντικείμενο ορίζεται ως μια μεταβλητή τύπου Κ
  - Στην Java (όπως και στις περισσότερες γλώσσες) όλες οι μεταβλητές έχουν ένα τύπο.
  - Το αντικείμενο που δημιουργείται παίρνει κάποιες τιμές στα πεδία της κλάσης και καταλαμβάνει κάποιο χώρο στη μνήμη.
    - Έτσι μετατρέπεται σε ένα φυσικό αντικείμενο με μοναδική ταυτότητα.

Δημιουργώντας φως



Θέλουμε να κάνουμε ένα πρόγραμμα που να διαχειρίζεται τα φώτα σε διάφορα δωμάτια και θα υλοποιεί και ένα dimmer



Όνομα κλάσης

Πεδία κλάσης

Μέθοδοι κλάσης

Αντικείμενα:

Light bedroomLight
Light kitchenLight

Τα αντικείμενα δημιουργούνται σε άλλο σημείο του κώδικα το οποίο καλεί και τις μεθόδους

Η κλήση μιας μεθόδου για ένα αντικείμενο μερικές φορές λέγεται και πέρασμα μηνύματος

Πλεονεκτήματα Αντικειμενοσταφούς

- Τα αντικείμενα και οι κλάσεις μοντελοποιούν φυσικά τα αντικείμενα του κόσμου.
- Έχοντας ένα πρόβλημα μπορούμε να δημιουργήσουμε δομές που αντιστοιχούν σε στοιχεία στην περιγραφή του προβλήματος αντί να δημιουργούμε προγραμματιστικές δομές που μετά θα προσπαθήσουμε να ταιριάξουμε στο πρόβλημα.
- Τα πλεονεκτήματα είναι ότι αυτό κάνει τον κώδικα πιο φυσικό, πιο ευανάγνωστο, πιο τμηματοποιημένο, και πιο εύκολο να συντηρηθεί.

Παράδειγμα

- Θέλουμε να προσομοιώσουμε την κίνηση ενός αυτοκινήτου το οποίο κινείται πάνω σε μία ευθεία. Αρχικά ξεκινάει από τη θέση μηδέν. Σε κάθε χρονική στιγμή διαλέγει τυχαία να κινηθεί αριστερά ή δεξιά και μετακινείται κατά μία θέση. Σε κάθε βήμα τυπώνουμε μια κουκίδα που δείχνει τη θέση του.

- Πώς θα λύσουμε αυτό το πρόβλημα?
  - Τι κλάσεις και τι αντικείμενα θα ορίσουμε?
  - Τι πεδία και τι μεθόδους θα έχουν?

Παράδειγμα

- Θέλουμε να προσομοιώσουμε την κίνηση ενός αυτοκινήτου το οποίο κινείται πάνω σε μία ευθεία. Αρχικά ξεκινάει από τη θέση μηδέν. Σε κάθε χρονική στιγμή κινείται κατά μία θέση είτε αριστερά είτε δεξιά (το επιλέγει τυχαία). Σε κάθε βήμα τυπώνεται μια κουκίδα που δείχνει τη θέση του.

- Πώς θα λύσουμε αυτό το πρόβλημα?
  - Τι κλάσεις και τι αντικείμενα θα ορίσουμε?
  - Τι πεδία και τι μεθόδους θα έχουν?

Υλοποίηση

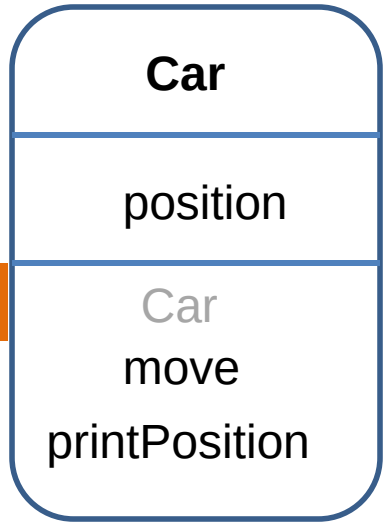
Αν έχω δύο αυτοκίνητα?

Όνομα κλάσης

Πεδία κλάσης

Μέθοδοι κλάσης

Constructor

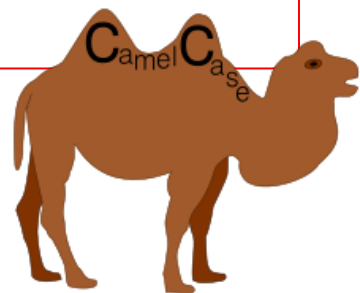


Πρόγραμμα
- Δημιούργησε το αντικείμενο myCar τύπου Car
  - Car myCar = new Car()
- myCar.printPosition()
- For i = 1…10
  - myCar.move()
  - myCar.printPosition()

Programming Style
- Τα ονόματα των κλάσεων ξεκινάνε με κεφαλαίο, τα ονόματα των πεδίων, μεθόδων και αντικειμένων με μικρό.
- Χρησιμοποιούμε ολόκληρες λέξεις (και συνδυασμούς τους) για τα ονόματα
  - Δεν πειράζει αν βγαίνουν μεγάλα ονόματα
- Χρησιμοποιούμε το CamelCase Style
  - Όταν για ένα όνομα έχουμε πάνω από μία λέξη, τις συνενώνουμε και στο σημείο συνένωσης κάνουμε το πρώτο γράμμα της λέξης κεφαλαίο
  - printPosition όχι print_position
- Χρησιμοποιούμε κεφαλαία και '_' για τις σταθερές.

Λείπει κάτι?

Αρχικοποίηση?



Αντικειμενοστραφής Σχεδίαση

- Οι οντότητες/έννοιες στον ορισμό του προβλήματος γίνονται κλάσεις και ορίζονται τα αντικείμενα που αναφέρονται στο πρόβλημα.
- Τα ρήματα γίνονται μέθοδοι
- Τα χαρακτηριστικά των αντικειμένων γίνονται πεδία
  - Τα πεδία μπορεί να είναι κι αυτά αντικείμενα.

- Δεν υπάρχει ένας μοναδικός τρόπος να μοντελοποιήσετε ένα πρόβλημα. Συνήθως όμως υπάρχει ένας που είναι καλύτερος από τους άλλους.
  - Υπάρχει ειδικό μάθημα γι αυτό το πρόβλημα.

Απόκρυψη - Ενθυλάκωση

- Στο πρόγραμμα που κάναμε πριν δεν είχαμε πρόσβαση στην θέση του αυτοκινήτου
  - Μόνο οι μέθοδοι της κλάσης μπορούν να την αλλάξουν.
  - Γιατί?
    - Αν μπορούσε να αλλάζει σε πολλά σημεία στον κώδικα τότε κάποιες άλλες μέθοδοι θα μπορούσαν να το αλλάξουν και να δημιουργηθεί μπέρδεμα
    - Τώρα αλλάζει μόνο όταν είναι λογικό να αλλάξει – όταν κινηθεί το όχημα.
- Επίσης κάποιος που χρησιμοποιεί τις μεθόδους της κλάσης δεν ξέρει πως υλοποιούνται, απλά μόνο τι κάνουν
- Αυτή η αρχή λέγεται Ενθυλάκωση – Απόκρυψη Πληροφορίας

Παράδειγμα 2

- Θέλω να δημιουργήσω ένα «τηλεφωνικό κατάλογο» στο οποίο θα αποθηκεύω ζεύγη από ονόματα και αριθμούς
  - Π.χ., τηλεφωνικός κατάλογος στο κινητό.
- Θέλω να μπορώ να προσθέτω και να αφαιρώ επαφές, και να παίρνω το τηλέφωνο μιας επαφής όταν δίνω το όνομα.

- Πιο γενικά θέλω ένα σύστημα που να αποθηκεύει (key,value) ζεύγη και να μου δίνει τις παραπάνω δυνατότητες.

- Πως θα επιλύσω αυτό το πρόβλημα?
  - Τι κλάσεις πρέπει να ορίσω?
  - Τι πεδία και τι μεθόδους θα πρέπει να έχουν?

Παράδειγμα 2

- Θέλω να δημιουργήσω ένα «τηλεφωνικό κατάλογο» στο οποίο θα αποθηκεύω ζεύγη από ονόματα και αριθμούς
  - Π.χ., τηλεφωνικός κατάλογος στο κινητό.
- Θέλω να μπορώ να προσθέτω και να αφαιρώ επαφές, και να παίρνω το τηλέφωνο μιας επαφής όταν δίνω το όνομα.

- Πιο γενικά θέλω ένα σύστημα που να αποθηκεύει (key,value) ζεύγη και να μου δίνει τις παραπάνω δυνατότητες.

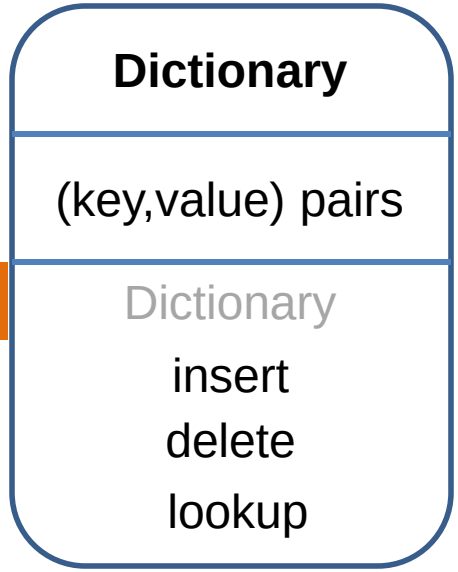
- Πως θα επιλύσω αυτό το πρόβλημα?
  - Τι κλάσεις πρέπει να ορίσω?
  - Τι πεδία και τι μεθόδους θα πρέπει να έχουν

Υλοποίηση

Όνομα κλάσης

Πεδία κλάσης

Μέθοδοι κλάσης



Πώς θα κρατάμε τα (key,value) pairs?

Υπάρχουν πολλές δομές που μπορούμε να χρησιμοποιήσουμε

Ο χρήστης της κλάσης Dictionary δεν χρειάζεται να ξέρει ποια δομή και τι αλγόριθμο χρησιμοποιούμε!

Η κλάση παρέχει ένα interface που αυτός χρησιμοποιεί.

Τι άλλες λειτουργίες θα θέλατε να έχει η κλάση Dictionary?

- size, isEmpty, contains

Τι άλλα πεδία χρειάζομαι?

- numberOfPairs

Αφηρημένοι τύποι δεδομένων

- Το προηγούμενο παράδειγμα δείχνει τη διαφορά μεταξύ Δομών Δεδομένων και Αφηρημένων Τύπων Δεδομένων (Abstract Data Types – ADTs)
- Ο Αφηρημένος Τύπος Δεδομένων ορίζει ένα σύνολο από λειτουργίες που πρέπει να υποστηρίζονται.
- Η Δομή Δεδομένων ενδιαφέρεται για ένα έξυπνο τρόπο να αποθηκεύσουμε τα δεδομένα ώστε να μπορούμε να κάνουμε τις παραπάνω λειτουργίες

Παράδειγμα 3

- Θέλω να δημιουργήσω μια μηχανή αναζήτησης η οποία θα παίρνει ένα ερώτημα και θα μου τυπώνει τα κείμενα που περιέχουν το ερώτημα.

- Πως θα επιλύσω αυτό το πρόβλημα?
  - Τι κλάσεις πρέπει να ορίσω?
  - Τι πεδία και τι μεθόδους θα πρέπει να έχουν?

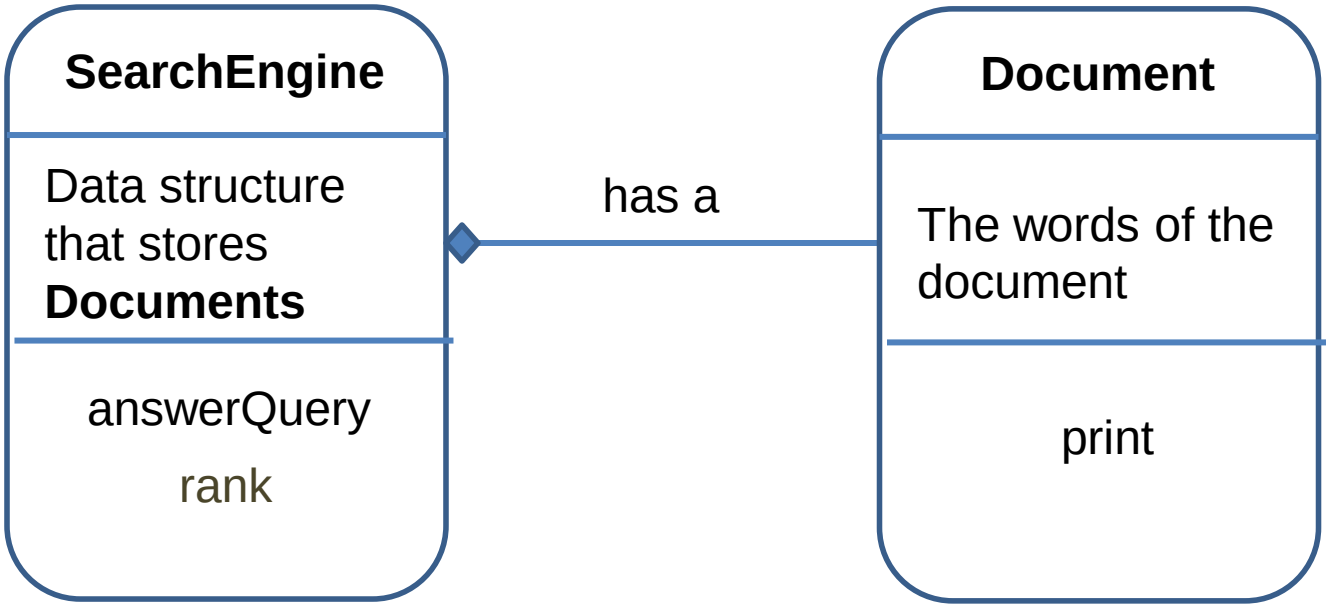
Παράδειγμα 3

- Θέλω να δημιουργήσω μια μηχανή αναζήτησης η οποία θα παίρνει ένα ερώτημα και θα μου τυπώνει τα κείμενα που περιέχουν το ερώτημα.

- Πως θα επιλύσω αυτό το πρόβλημα?
  - Τι κλάσεις πρέπει να ορίσω?
  - Τι πεδία και τι μεθόδους θα πρέπει να έχουν?

Υλοποίηση

Σύνθεση



Λείπει κάτι?

Οι μηχανές αναζήτησης επιστρέφουν τα κείμενα ταξινομημένα (ranked)

Προσθέτουμε μια μέθοδο rank
Η μέθοδος αυτή είναι ιδιωτική (private), δεν φαίνεται εξωτερικά

Θα μπορούσαμε να έχουμε μία κλάση Ranker και ένα αντικείμενο που να κάνει το ranking

Παράδειγμα 4

- Τι γίνεται αν στο αρχικό μας παράδειγμα εκτός από αυτοκίνητο είχαμε και μία μηχανή? Η μηχανή κινείται με τον ίδιο τρόπο αλλά όταν τυπώνεται η θέση της αντί για κουκίδα (.) τυπώνεται ένα αστέρι (*).

- Τι κλάσεις πρέπει να ορίσουμε?

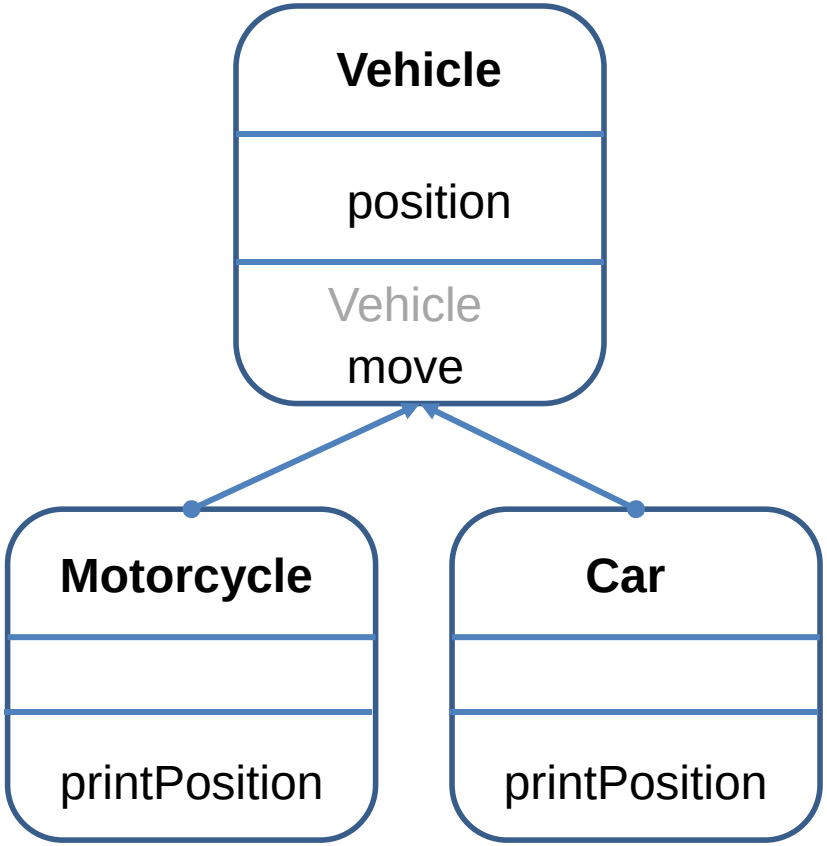
Μία λύση

- Μπορούμε να ορίσουμε ξανά από την αρχή μια καινούρια κλάση για τη μηχανή που να κάνει την ίδια κίνηση με το αυτοκίνητο (ίδια μέθοδο move) αλλά τυπώνεται διαφορετικά (διαφορετική μέθοδο printPosition)

- Τι προβλήματα έχει αυτό?
  - Επανάληψη του κώδικα (μπορεί να είναι μεγάλος και δύσκολος)
  - Πιθανότητα για λάθη
  - Πολύ δύσκολο να γίνουν αλλαγές

Κληρονομικότητα

- Ορίζουμε μια κλάση Vehicle η οποία έχει θέση, και έχει κίνηση (μέθοδο move)




```
Πρόγραμμα
Car myCar = new Car()
Motorcycle myMoto = new Motorcycle()
myCar.printPosition()
myMoto.printPosition()
For i = 1…10
        myCar.move()
        myCar.printPosition()
        myMoto.move()
        myMoto.printPosition()
```