

## Κλάσεις, Μέθοδοι και Αντικείμενα

Η αντικειμενοστραφής πλευρά της Java φαίνεται με την χρήση των αντικειμένων και των κλάσεων στις οποίες ανήκουν τα αντικείμενα.

Όλα τα αντικείμενα που έχουν κοινά χαρακτηριστικά ανήκουν στην ίδια κλάση και κάθε ένα από αυτά λέγεται ότι είναι “στιγμιότυπο” (instance) της κλάσης.

Για παράδειγμα, η κλάση “αυτοκίνητο” θεωρεί μια γενική εικόνα του αυτοκινήτου με κάποια χαρακτηριστικά που υπάρχουν σε όλα τα στιγμιότυπα.

Συγκεκριμένα, η κλάση αυτοκίνητο πιθανώς να ορίζει ότι είναι τετράτροχο, έχει τιμόνι, ταχύτητες, πεντάλ και καθίσματα για τους οδηγούς αλλά δεν ορίζει με σαφήνεια το σχήμα ή τους μηχανισμούς όλων των εξαρτημάτων, ούτε τα χαρακτηριστικά της μηχανής (κυβικά, ίπποι, κύλινδροι, κλπ).

Αυτά είναι χαρακτηριστικά που αφορούν το κάθε αντικείμενο ξεχωριστά (ή κάποια υποκατηγορία/υποκλάση της κλάσης “αυτοκίνητο”).

Στη Java (και στη C++ στην οποία βασίστηκε η Java) η κλάση δηλώνεται με τη λέξη class. Στο παράδειγμα του αυτοκινήτου, έστω ότι θέλουμε να δηλώσουμε μια κλάση με το όνομα Car:

```
class Car
{
    (δεδομένα/μεταβλητές)
    (συναρτήσεις/μέθοδοι)
}
```

Τα χαρακτηριστικά του αντικειμένου είναι τα δεδομένα γύρω από τα οποία πρέπει να αναπτύξουμε το πρόγραμμά μας. Στο παράδειγμα με το αυτοκίνητο, τα δεδομένα αυτά είναι: η γωνία στρέψης του τιμονιού, η πίεση που ασκούμε στα πεντάλ, η θέση του μοχλού ταχυτήτων και άλλα.

Στον προγραμματισμό, αυτά τα δεδομένα θα πρέπει να μοντελοποιηθούν, δηλαδή να γίνει η αντιστοιχία τους σε μεταβλητές τις οποίες μπορούμε να επεξεργαστούμε στο πρόγραμμά μας. Για την κλάση Car, μπορούμε να έχουμε:

```
class Car
{
    // Γωνία στρέψης του τιμονιού
    float steering_angle;
```

```

// Ποσοστό πατήματος του γκαζιού (0 = καθόλου, 100 = τερματισμένο!)
float gas_pedal;

// Ποσοστό πατήματος του φρένου (0 = καθόλου, 100 = τερματισμένο!)
float break_pedal;

// Ποσοστό πατήματος του συμπλέκτη (0 = καθόλου, // 100 = τερματισμένο!)
float clutch;

// Θέση της τρέχουσας ταχύτητα (πιθανές τιμές: 0, 1,2,3,4,5, // 0 = νεκρό, -1 = όπισθεν)
int gear;

// μεταβλητές που καθορίζουν την επιτάχυνση, την ταχύτητα του
// αυτοκινήτου και τις στροφές του κινητήρα
float acceleration, speed, rpm;
}

```

Εδώ πρέπει να σημειωθεί ότι οι μεταβλητές της κλάσης έχουν διαφορετικές τιμές για κάθε αντικείμενο, και κάθε αντικείμενο έχει πρόσβαση μόνο στις δικές του μεταβλητές. Επίσης, ο τύπος δεδομένων που επιλέχθηκε για κάθε μεταβλητή εξαρτάται από το είδος της πληροφορίας που θα κρατάει αυτή η μεταβλητή. Για παράδειγμα, εφόσον η γωνία στρέψης του τιμονιού θα είναι σε μοίρες, είναι λογικό να χρησιμοποιήσουμε ένα τύπο που θα μπορεί να κρατήσει δεκαδικούς αριθμούς, όπως ο float.

## Μέθοδοι

Οι μέθοδοι (methods) σε μια κλάση, δεν είναι τίποτε παραπάνω από συναρτήσεις (υπορουτίνες) που προσφέρουν πρόσβαση στα δεδομένα του εκάστοτε αντικειμένου της κλάσης. Η αλληλεπίδραση του χρήστη με κάθε αντικείμενο γίνεται μέσω των μεθόδων της κλάσης, οι οποίες καθορίζουν και τον τρόπο χειρισμού των μεταβλητών του αντικειμένου.

Στο παράδειγμά μας, οι μέθοδοι θα καθορίζουν με ποιον τρόπο θα στρίβουμε το τιμόνι, θα αυξάνουμε το πάτημα στο γκάζι ή στο φρένο, θα αλλάζουμε ταχύτητες αλλά και πώς θα μπορούμε να γνωρίζουμε την ταχύτητα του αυτοκινήτου, τις στροφές του κινητήρα δηλαδή πληροφορίες που δε μπορούμε να ελέγξουμε άμεσα.

Πιθανές μέθοδοι για την κλάση "Car" θα μπορούσαν να είναι οι εξής:

```

// Αλλαγή της γωνία στρέψης του τιμονιού, μοίρες

// σε σχέση με την τρέχουσα γωνία.

```

```

void turn_wheel(float relative_angle);

// Πάτημα πεντάλ γκαζιού

void press_gas_pedal(float amount);

// Πάτημα πεντάλ φρένου

void press_break_pedal(float amount);

// Πάτημα πεντάλ συμπλέκτη

void press_clutch_pedal(float amount);

// Αλλαγή της ταχύτητας. Επιστρέφει true αν η αλλαγή ήταν επιτυχής
// ή false αν ήταν ανεπιτυχής (π.χ. από 5 σε όπισθεν).

bool change_gear(int new_gear);

// προβολή της τρέχουσας ταχύτητας, επιτάχυνσης και στροφών του
// κινητήρα

float get_acceleration();

float get_speed();

float get_rpm();

```

### **Δημιουργία αντικειμένων με τη new**

Έχοντας ορίσει την κλάση μας, θα πρέπει να δημιουργήσουμε τα αντικείμενα -τα στιγμιότυπα (instances) της κλάσης.

Για το σκοπό αυτό, χρησιμοποιούμε την εντολή new της Java. Η new δημιουργεί μια φυσική αναπαράσταση της κλάσης, ένα μοναδικό στιγμιότυπο, και επιστρέφει ένα δείκτη αναφοράς (reference) σε αυτό.

Με αυτό το δείκτη αναφοράς μπορούμε να προσπελάσουμε το αντικείμενο με οποιόν τρόπο θέλουμε (και εφόσον το επιτρέπει το ίδιο το αντικείμενο).

Ο τρόπος χρήσης της new είναι ο εξής:

```
Car acar = new Car();
```

```
Car anothercar = new Car();
```

Οι μεταβλητές acar και anothercar είναι οι δείκτες αναφοράς στα αντίστοιχα αντικείμενα που δημιουργούνται με τη new. Η δημιουργία του αντικείμενου δεν είναι αναγκαίο να γίνει κατά την δήλωσή του. Το ίδιο αποτέλεσμα μπορούμε να έχουμε και με τις εντολές:

```
Car acar; acar = new Car();
```

Τα δεδομένα του κάθε αντικειμένου (οι μεταβλητές) αλλά και οι μέθοδοι της κλάσης που αντιστοιχούν στο αντικείμενο, μπορούν να προσπελαστούν ως εξής:

```
// Η γωνία στρέψης του τιμονιού του
acar acar.steering_angle

// Η γωνία στρέψης του τιμονιού του
anothercar anothercar.steering_angle

// Εντολή στο acar να στρίψει δεξιά 13.4 μοίρες.
acar.turn(13.4);

// Επιστρέφει την τρέχουσα ταχύτητα του
acar float speed = acar.get_speed();

// Εντολή στο anothercar να στρίψει αριστερά 32 μοίρες
anothercar.turn(-32.0);

// Εντολή στο anothercar να βάλει όπισθεν
bool result = anothercar.gchange_gear(-1);
```

Όπως βλέπουμε ορισμένοι μέθοδοι δέχονται κάποιες παραμέτρους (εντός παρενθέσεων). Οι παράμετροι μπορεί να είναι μεταβλητές οποιουδήποτε αποδεκτού τύπου στη Java ή ακόμη και άλλα αντικείμενα. Αν έχουμε περισσότερες από μία παραμέτρους τις διαχωρίζουμε με κόμμα ','.

## Κατασκευαστές

Όταν δημιουργείται ένα αντικείμενο με την εντολή new, στην πραγματικότητα η Java, αφού δεσμεύσει την απαραίτητη μνήμη για το αντικείμενο, εκτελεί μια συγκεκριμένη μέθοδο της κλάσης, τον δημιουργό (constructor).

Ο δημιουργός πραγματοποιεί τις απαραίτητες ενέργειες για να καταστεί κάποιο αντικείμενο έτοιμο προς χρήση. Αυτές μπορεί να είναι κάτι απλό όπως η ρύθμιση κάποιων μεταβλητών με αρχικές τιμές, ή πιο περίπλοκο όπως η δημιουργία σύνδεσης με μια βάση δεδομένων, η αρχικοποίηση των πινάκων SQL, η δέσμευση κάποιων δικτυακών θυρών (sockets) για κάποιο server ή ακόμη και το άνοιγμα κάποιου παραθύρου για εμφάνιση γραφικής πληροφορίας.

Ο δημιουργός έχει τη μορφή μιας μεθόδου με το όνομα της κλάσης και χωρίς τύπο (δηλαδή δεν δηλώνεται ο τύπος δεδομένων που επιστρέφει, το οποίο είναι διαφορετικό από το να δηλωθεί ως void).

Για παράδειγμα, στην κλάση "Car", ένας πιθανός δημιουργός μπορεί να είναι:

```
Car()
{
    steering_wheel = 0.0;
    gas_pedal = 0.0;
    break_pedal = 0.0;
    float clutch = 0.0;
    int gear = 0;
    acceleration = 0.0;
    speed = 0.0;
    rpm = 0.0;
}
```

Δηλαδή, ο δημιουργός πραγματοποιεί την αρχικοποίηση (initialization) των μεταβλητών του αντικειμένου ώστε αυτό να είναι έτοιμο προς χρήση. Μπορούμε να έχουμε περισσότερους από έναν δημιουργούς, οι οποίοι να δέχονται διαφορετικές παραμέτρους ο καθένας.

Για παράδειγμα, αν υποθέσουμε ότι μπορούσαμε να ορίσουμε χαρακτηριστικά του κινητήρα (engine\_cc: κυβικά, engine\_hp: ίπποι) στη δημιουργία του αντικειμένου, θα μπορούσαμε να έχουμε έναν επιπλέον δημιουργό:

```
Car(int cc, int hp)
{
    engine_cc = cc;
    engine_hp = hp;
    // Ακολουθούν οι υπόλοιπες εντολές αρχικοποίησης του αντικειμένου
}
```

επιπλέον του αρχικού δημιουργού. Η δημιουργία περισσότερων από έναν δημιουργούς καλείται constructor overloading και είναι υποπερίπτωση του χαρακτηριστικού της Java, method overloading (σε επόμενη παράγραφο).

