

Δημιουργία Κλάσεων/Αντικειμένων/Συσχετίσεων/Συνδέσεων

Δημιουργία κλάσεων και αντικειμένων

Θεωρούμε ένα υποθετικό σύστημα που αναφέρεται σε Φοιτητές και Μαθήματα. Για τη μοντελοποίηση των Φοιτητών δημιουργούμε την κλάση Student. Η κλάση διαθέτει ως ιδιότητες το όνομα και τον αριθμό μητρώου του φοιτητή, ενώ από πλευράς μεθόδων πέραν των κατασκευαστών και των μεθόδων πρόσβασης διαθέτει και μια μέθοδο εκτύπωσης των τιμών των ιδιοτήτων.

```
public class Student {  
  
    private String name;  
    private String id;  
  
    public Student(String text1, String text2) {  
        name = text1;  
        id = text2;  
    }  
  
    public Student(String text1) {  
        name = text1;  
        id = "9999";  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getID() {  
        return id;  
    }  
  
    public void printInfo() {  
        System.out.println("Name: " + name);  
        System.out.println("AM: " + id);  
    }  
  
}
```

Η κατασκευή αντικειμένων της κλάσης Student μπορεί να πραγματοποιηθεί είτε στη main είτε σε οποιαδήποτε μέθοδο κλάσης:

```

public class Main {

    public static void main(String[] args) {
        Student S1 = new Student("John", "1209");
        Student S2 = new Student("Nick");

        S1.printInfo();

        S2.printInfo();
    }
}

```

Το τμήμα της εντολής `Student S1 = new Student("John", "1209");` μετά τον τελεστή "=", δηλαδή:

```
new Student("John", "1209");
```

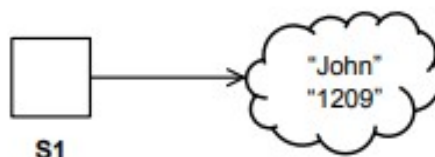
δημιουργεί ένα αντικείμενο της κλάσης `Student` καλώντας τον αντίστοιχο κατασκευαστή και μεταβιβάζοντας ως ορίσματα το όνομα και τον ΑΜ του φοιτητή.

Το τμήμα της εντολής πριν από τον τελεστή "=", δηλαδή

`Student S1`

δημιουργεί μια αναφορά τύπου `Student`, δηλαδή μια μεταβλητή της οποίας το περιεχόμενο μπορεί να λάβει ως τιμή τις διευθύνσεις αντικειμένων τύπου `Student` (ουσιαστικά πρόκειται για έναν "δείκτη").

Μετά την εκτέλεση της εντολής, ανατίθεται στην αναφορά `S1` η διεύθυνση του αντικειμένου που δημιουργήθηκε και κατά συνέπεια η αναφορά "δείχνει" πλέον στο δημιουργηθέν αντικείμενο. Αυτό μπορεί να παρασταθεί διαγραμματικά ως εξής:



Επειδή όλες οι ενέργειες επί του αντικειμένου που δημιουργήθηκε πραγματοποιούνται μέσω της αναφοράς S1, καταχρηστικά αναφερόμαστε συχνά στο ίδιο το αντικείμενο ως 'S1'. Ωστόσο θα πρέπει να γίνει κατανοητή η διαφορά μεταξύ μιας αναφοράς ενός αντικειμένου (η οποία προτού αρχικοποιηθεί μπορεί να μην "δείχνει" πουθενά) και του ίδιου του αντικειμένου.

Η αποστολή ενός μηνύματος στο αντικείμενο ισοδυναμεί με την κλήση μιας μεθόδου του αντικειμένου. Για παράδειγμα, η αποστολή του μηνύματος `printInfo()` στο αντικείμενο προς το οποίο δείχνει η αναφορά S1 με σκοπό να εκτυπωθούν στην κονσόλα οι τιμές των ιδιοτήτων του πραγματοποιείται με την εντολή:

```
S1.printInfo();
```

Δημιουργία συσχετίσεων μεταξύ κλάσεων

Για τη μοντελοποίηση ενός μαθήματος δημιουργούμε την κλάση `Course`:

```
public class Course {  
  
    private String name;  
    private int credits;  
  
    public Course(String text, int number) {  
        name = text;  
        credits = number;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getCredits() {  
        return credits;  
    }  
  
}
```

Για να μοντελοποιήσουμε το γεγονός ότι ένας φοιτητής παρακολουθεί ένα μάθημα (και άρα σχετίζεται με αυτό) δημιουργούμε μια συσχέτιση μεταξύ των κλάσεων Student και Course με κατεύθυνση από την πρώτη προς τη δεύτερη (μονόδρομη συσχέτιση). Μια τέτοια συσχέτιση σε ένα διάγραμμα κλάσεων απεικονίζεται ως εξής:



Η συσχέτιση υποδηλώνει ότι ένα αντικείμενο τύπου Student "γνωρίζει" το αντικείμενο Course με το οποίο σχετίζεται και επιτρέπει την αποστολή μηνυμάτων από το αντικείμενο Student προς το αντικείμενο Course (όχι όμως και το ανάποδο, δεδομένου ότι η συσχέτιση έχει μονόδρομη κατεύθυνση). Η συσχέτιση υλοποιείται στην κλάση Student ως μια ιδιότητα που είναι αναφορά τύπου Course:

```
private Course myCourse;
```

προφανώς, αν η αναφορά αυτή δεν λάβει συγκεκριμένη τιμή δεν "δείχνει" πουθενά. Μπορούμε να αναθέσουμε τιμή σε μια αναφορά όπως και σε οποιαδήποτε άλλη ιδιότητα. Αν υποθέσουμε ότι η αναφορά λαμβάνει τιμή μέσω κατάλληλης μεθόδου set, η αντίστοιχη μέθοδος θα ήταν ως εξής:

```
public void setCourse(Course c) {
    myCourse = c;
}
```

Για να συσχετίσουμε ένα συγκεκριμένο αντικείμενο Student με ένα αντικείμενο Course θα πρέπει, αφού δημιουργηθούν και τα 2 αντικείμενα, να καλέσουμε τη μέθοδο setCourse() επί του αντικειμένου Student μεταβιβάζοντας ως όρισμα το αντικείμενο Course. Για παράδειγμα στη main:

```

public class Main {

    public static void main(String[] args) {
        Student S1 = new Student("John", "1209");
        Student S2 = new Student("Nick");

        Course C1 = new Course("Java", 5);

        S1.setMyCourse(C1); //δημιουργία σύνδεσης Student S1->Course C1
        S2.setMyCourse(C1);

        S2.printInfo();
        S2.printInfo();
    }
}

```

Μετά την εκτέλεση της εντολής `S1.setMyCourse(C1)`; η αναφορά `myCourse` του αντικειμένου `S1` "δείχνει" πλέον στο αντικείμενο `C1`. Από το σημείο αυτό και μετά υφίσταται σύνδεση (link) μεταξύ των δύο αντικειμένων. Επισημαίνεται ότι οι συσχετίσεις υφίστανται στατικά μεταξύ κλάσεων και δημιουργούν την "υποδομή" για τη σύνδεση αντικειμένων. Απλώς και μόνο η ύπαρξη μια συσχέτισης δεν οδηγεί αυτομάτως και σε δημιουργία συνδέσεων μεταξύ των αντικειμένων. Οι συνδέσεις δημιουργούνται δυναμικά κατά τη διάρκεια της εκτέλεσης και μπορούν να μεταβάλλονται ή να καταστρέφονται. (για παράδειγμα η αναφορά `myCourse` του αντικειμένου `S1` μπορεί να τροποποιηθεί στη συνέχεια ώστε να "δείχνει" σε άλλο μάθημα).

Ένα αντικείμενο τύπου `Student` μπορεί να αξιοποιήσει τη γνώση του συσχετισμένου μαθήματος ώστε να αποστείλει μηνύματα σε αυτό είτε για την εκτέλεση λειτουργιών είτε για την άντληση δεδομένων. Στον παρακάτω πλήρη κώδικα της κλάσης `Student` η μέθοδος `printInfo` έχει τροποποιηθεί ώστε μαζί με τα στοιχεία του φοιτητή να εκτυπώνονται και στα στοιχεία του συσχετιζόμενου μαθήματος.

```

public class Student {

    private String name;
    private String id;
    private Course myCourse; //αναφορά προς αντικείμενο τύπου Course

    public Student(String text1, String text2) {
        name = text1;
        id = text2;
    }

    public Student(String text1) {
        name = text1;
        id = "9999";
    }
}

```

```
}  
  
public void setMyCourse(Course c) {  
    myCourse = c;  
}  
  
public String getName() {  
    return name;  
}  
  
public String getID() {  
    return id;  
}  
  
public void printInfo() {  
    System.out.println("Name: " + name);  
    System.out.println("AM: " + id);  
    System.out.println("Course: " + myCourse.getName());  
    System.out.println("Credits: " + myCourse.getCredits());  
}  
}
```