

## Αναγκαιότητα Κλάσεων

Θεωρούμε το κάτωθι απλό πρόγραμμα, το οποίο, αν και γραμμένο στη γλώσσα προγραμματισμού Java δεν έχει **καθόλου αντικειμενοστρεφή χαρακτηριστικά**. Όλος ο κώδικας βρίσκεται μέσα στη μέθοδο main (στατική μέθοδος που πρέπει να υφίσταται σε κάθε πρόγραμμα Java δεδομένου ότι από αυτήν ξεκινά πάντοτε η εκτέλεση του προγράμματος).

Το πρόγραμμα λαμβάνει για ένα πλήθος υποθετικών προϊόντων (για όσα θέλει να καταχωρήσει ο χρήστης), το όνομά τους, την τιμή τους και το σκορ τους, βάσει κάποιας κλίμακας αξιολόγησης. Μετά το πέρας της καταχώρησης των στοιχείων το πρόγραμμα εκτυπώνει το "βέλτιστο" προϊόν, δηλαδή αυτό που επιτυγχάνει τον καλύτερο λόγο σκορ/τιμή.

Από πλευράς μεταβλητών, χρησιμοποιεί 3 μεταβλητές για την καταχώρηση του τρέχοντος προϊόντος σε κάθε επανάληψη (next\_name, next\_price, next\_score) και 3 μεταβλητές για την αποθήκευση των στοιχείων του βέλτιστου προϊόντος σε κάθε επανάληψη (best\_name, best\_price, best\_score).

### Πρώτη εκδοχή προγράμματος (μη αντικειμενοστρεφής)

```
import java.util.Scanner;

public class Main {

    /**
     * @param args
     */
    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        String best_name = "";
        double best_price = 1;
        int best_score = 0;

        boolean more = true;
        while(more) {

            String next_name;
            double next_price;
            int next_score;

            System.out.println("Please enter the product name: ");
            next_name = in.nextLine();
            System.out.println("Please enter the product price: ");
            next_price = in.nextDouble();
            System.out.println("Please enter the product score: ");
            next_score = in.nextInt();

            if(next_score/next_price > best_score/best_price) {
                best_name = next_name;
            }
        }
    }
}
```

```
        best_price = next_price;
        best_score = next_score;
    }

    System.out.println("More data ? (1=YES, 2=NO)");
    int answer = in.nextInt();
    if(answer != 1)
        more = false;
    in.nextLine();
}

System.out.println("The best product is: " + best_name);
System.out.println("The best price is: " + best_price);
System.out.println("The best score is: " + best_score);
}
}
```

Αν εξετάσουμε το πρόγραμμα προσεκτικά, και σε ένα πιο αφαιρετικό επίπεδο, θα διαπιστώσουμε ότι στο πρόγραμμα υποκρύπτονται δύο "έννοιες", αυτή του τρέχοντος προϊόντος και του βέλτιστου προϊόντος. Από πλευράς χαρακτηριστικών, και το τρέχον και το βέλτιστο προϊόν έχουν τις ίδιες ιδιότητες (όνομα, τιμή και σκορ), κατά περίπτωση όμως διαφορετικές τιμές σε αυτές τις ιδιότητες (σε κάποιο δεδομένο σημείο της εκτέλεσης το βέλτιστο μπορεί να είναι διαφορετικό από το τρέχον προϊόν που καταχωρήθηκε).

Επομένως, στο συγκεκριμένο πρόγραμμα, υπονοείται η ύπαρξη μιας "κατηγορίας" αντικειμένων τύπου "Προϊόν", έννοια που στις αντικειμενοστρεφείς γλώσσες προγραμματισμού μοντελοποιείται από μια **κλάση**. Μια κλάση αποτελεί την γενική κατηγορία, την "μήτρα" από την οποία μπορούν να παραχθούν διάφορα στιγμιότυπα – αντικείμενα. Όλα τα αντικείμενα μιας κλάσης έχουν τις ίδιες ιδιότητες (και την ίδια συμπεριφορά), λαμβάνουν όμως διαφορετικές τιμές για τις ιδιότητες αυτές.

Στην προκειμένη περίπτωση υπονοείται η ύπαρξη της κλάσης "Προϊόν" με ιδιότητες το όνομα, σκορ και τιμή, ενώ το τρέχον και το βέλτιστο προϊόν αποτελούν στιγμιότυπα της κλάσης αυτής.

Για την εύρεση των λειτουργιών/μεθόδων που πρέπει να περιλαμβάνει η κλάση "Προϊόν" διερευνούμε τον υπάρχοντα κώδικα (1η εκδοχή) για να δούμε τις λειτουργίες στις οποίες εμπλέκεται ένα αντικείμενο τύπου "Προϊόν". Προκύπτει ότι:

- ένα προϊόν πρέπει να μπορεί να αρχικοποιήσει τις τιμές των ιδιοτήτων του. Αυτό επιτυγχάνεται με την προσθήκη ενός κατασκευαστή στην κλάση "Προϊόν". Ο κατασκευαστής αποτελεί ειδική μέθοδο μιας κλάσης που καλείται οποτεδήποτε δημιουργούμε ένα αντικείμενο της κλάσης. Στους κατασκευαστές συνήθως συμπεριλαμβάνουμε και την αρχικοποίηση των τιμών των ιδιοτήτων που επιθυμούμε.
- για ένα προϊόν θα πρέπει να μπορούμε να "διαβάσουμε" τιμές από το πληκτρολόγιο και να τις καταχωρήσουμε στις αντίστοιχες ιδιότητές του. Επομένως "εφοδιάζουμε" την κλάση "Προϊόν" και με μια μέθοδο read().
- για κάθε προϊόν, θα πρέπει να μπορούμε να συγκρίνουμε τον εν λόγω προϊόν με κάποιο άλλο προϊόν για να βρούμε ποιο από τα 2 είναι το καλύτερο (με βάση το λόγο σκορ/τιμή). Επομένως

"εφοδιάζουμε" την κλάση "Προϊόν" και με μια μέθοδο `is_better_than(Product other)`. Η μέθοδος καλείται επί ενός αντικειμένου (έστω `X`) μεταβιβάζοντας ως παράμετρο ένα άλλο αντικείμενο τύπου Προϊόν (έστω `other`). Η μέθοδος επιστρέφει `true` αν το `X` είναι καλύτερο από το `other`.

- τέλος, για ένα προϊόν θα πρέπει να μπορούμε να εκτυπώσουμε τις τιμές των ιδιοτήτων τους. Επομένως "εφοδιάζουμε" την κλάση "Προϊόν" και με μια μέθοδο `printData()`.

Με βάση τα ανωτέρω, ο ορισμός της κλάσης `Product` είναι ως εξής:

### Δεύτερη εκδοχή προγράμματος (αντικειμενοστρεφής)

```
import java.util.Scanner;

public class Product {

    private String name;
    private int score;
    private double price;

    //κατασκευαστής
    public Product() {
        name = "";
        score = 0;
        price = 1;
    }

    //μέθοδος ανάγνωσης τιμών από το πληκτρολόγιο
    public void read() {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter product name: ");
        name = in.nextLine();
        System.out.println("Enter product score: ");
        score = in.nextInt();
        System.out.println("Enter product price: ");
        price = in.nextDouble();
    }

    //μέθοδος σύγκρισης προϊόντος με ένα άλλο
    public boolean is_better_than(Product other) {
        if(score/price > other.score/other.price)
            return true;
        return false;
    }

    //μέθοδος εκτύπωσης τιμών ιδιοτήτων
    public void printData() {
        System.out.println("Product Name: " + name);
        System.out.println("Price: " + price);
        System.out.println("Score: " + score);
    }
}
```

Έχοντας πλέον την κλάση Προϊόν στη διάθεσή μας, μπορούμε να την αξιοποιήσουμε κατασκευάζοντας αντικείμενα της κλάσης στη συνάρτηση main και καλώντας τις κατάλληλες μεθόδους:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        //κατασκευή βέλτιστου προϊόντος
        Product best = new Product();

        boolean flag = true;
        while(flag) {

            //κατασκευή τρέχοντος προϊόντος
            Product current = new Product();

            current.read(); //αποστολή μηνύματος read στο αντικείμενο current
                           //κλήση της μεθόδου read επί του αντικειμένου current

            if(current.is_better_than(best))
                best = current;

            System.out.println("CONTINUE? (1=YES, 2=NO)");
            int answer = in.nextInt();
            if(answer != 1)
                flag = false;
            in.nextLine(); //flushes the input buffer
        }

        System.out.println("Best Product: ");
        best.printData();
    }
}
```

Στη δεύτερη εκδοχή του προγράμματος, αν και με στοιχειώδη τρόπο, η συμπεριφορά είναι αντικειμενοστρεφής, δηλαδή η λειτουργικότητα επιτυγχάνεται μέσω της αλληλεπίδρασης των αντικειμένων που δημιουργούνται (η αλληλεπίδραση συνίσταται στην σύγκριση των δύο αντικειμένων στην προκειμένη περίπτωση).