



## ΕΡΓΑΣΤΗΡΙΟ 21

### Αντικειμενοστραφής Προγραμματισμός ΙΙΙ

#### Άσκηση 1.

Χρησιμοποιώντας τη κλάση `Point` που γράψατε στα προηγούμενα εργαστήρια, γράψτε μια μέθοδο που ταξινομεί ένα πίνακα αντικειμένων `Point` παίρνοντας δυο παραμέτρους: η μια είναι ο πίνακας που πρέπει να ταξινομηθεί, και η άλλη είναι μια παράμετρος τύπου `Boolean`. Αν η παράμετρος είναι `true` τότε η ταξινόμηση γίνεται με βάση το `x`. Σε αντίθετη περίπτωση η ταξινόμηση γίνεται με βάση το `y`.

Η πιο εύκολη λύση, όχι όμως και η καλύτερη, είναι να γράψουμε αυτήν την μέθοδο μέσα στην κλάση `Point`. Εξηγήστε το γιατί;

Τι πρέπει να κάνουμε εάν χρειαστεί να γράψουμε την μέθοδο έξω από την κλάση `Point` σε ένα άλλο αρχείο; Δημιουργήστε τις απαραίτητες μεθόδους.

Υπάρχουν πολλές πιθανές λύσεις, αλλά εμείς πιο κάτω δίνουμε 2.

#### Λύσεις

**Λύση 1.** Η πρώτη λύση `sort1()` δίνει αποτέλεσμα που δεν είναι ικανοποιητικό. Απαιτείται να γράψουμε τη μέθοδο αυτή μέσα στη κλάση `Point` για να μπορούμε να έχουμε πρόσβαση στις μεταβλητές `x` και `y` που είναι δηλωμένες ως `private`.

```
/**
 * Tri le tableau de points passés en argument par x croissant si b
 * est vrai et par y croissant si b est faux.
 */
public static void sort1(Point[] t, boolean b) {
    if(b) {
        for(int i = 1; i<t.length; i++) {
            for(int j = 0; j<t.length-i; j++) {
                if(t[j].x>t[j+1].x) {
                    Point tmp = t[j];
                    t[j] = t[j+1];
                    t[j+1] = tmp;
                }
            }
        }
    } else {
        for(int i = 1; i<t.length; i++) {
            for(int j = 0; j<t.length-i; j++) {
                if(t[j].y>t[j+1].y) {
                    Point tmp = t[j];
                    t[j] = t[j+1];
                    t[j+1] = tmp;
                }
            }
        }
    }
}
```



**Λύση 2.** Η δεύτερη λύση `sort2()` είναι λίγο καλύτερη, καθώς μας επιτρέπει να γράψουμε τη ζητούμενη μέθοδο, ενδεχομένως εκτός της κλάσης `Point` χωρίς να έχουμε άμεση πρόσβαση στις `private` μεταβλητές `x` και `y`. Χρειαζόμαστε 3 νέες μεθόδους μέσα στη κλάση `Point`.

// Θέλουμε μέσα στη κλάση `Point`, 3 νέες μεθόδους:

```
public boolean greater(Point p, boolean b) {
    if (b)
        return greaterInX(p);
    return greaterInY(p);
}

private boolean greaterInX(Point p) {
    if (this.x < p.x)
        return false;
    return true;
}

private boolean greaterInY(Point p) {
    if (this.y < p.y)
        return false;
    return true;
}

//Οι πιο κάτω μέθοδοι δεν είναι ανάγκη να είναι δηλωμένες μέσα στη κλάση Point
public static void sort2(Point[] t, boolean b) {
    for(int i = 1; i<t.length; i++) {
        for(int j = 0; j<t.length-i; j++) {
            if(t[j].greater(t[j+1], b)) {
                Point tmp = t[j];
                t[j] = t[j+1];
                t[j+1] = tmp;
            }
        }
    }
}

public static void main(String[] args) {
    Point t[] = new Point[]{
        new Point(2,3),
        new Point(3,2),
        new Point(1,4),
        new Point(3,5),
        new Point(4,2),
    };
    System.out.println("Taxinomisi ws pros x :");
    sort2(t,true);
    for(int i=0; i<t.length; i++)
        System.out.print(t[i]+" ");
    System.out.println();
    System.out.println("Taxinomisi ws pros y :");
    sort2(t,false);
    for(int i=0; i<t.length; i++)
        System.out.print(t[i]+" ");
}
```