

**ΕΡΓΑΣΤΗΡΙΟ 15****Design interfaces and classes. Inheritance**

1. Μελετήστε το παρακάτω πρόγραμμα:

```
class A {
    protected int x() {
        return 3;
    }
    public void printX() {
        System.out.println(x());
    }
}
class B extends A {
    public int x() {
        return 4;
    }
}
class Test {
    public static void main(String[] args) {
        A a = new A();
        System.out.println(a.x());
        a.printX();

        B b = new B();
        System.out.println(b.x());
        b.printX();

        A ab = new B();
        System.out.println(ab.x());
        ab.printX();
    }
}
```

- Ποιες είναι οι τιμές που θα τυπωθούν; Εξηγήστε γιατί.
- Όταν «είσαστε» στη κλάση B, σε πόσες μεθόδους x() έχει πρόσβαση ένα αντικείμενο της κλάσης B; Αν υπάρχουν αρκετές, δώστε για την κάθε μια τον τρόπο πρόσβασης.
- Τι θα συμβεί εάν οι μέθοδοι x() είναι στατικές μέθοδοι;

2. Γιατί ο παρακάτω κώδικας εμφανίζει σφάλμα μεταγλώττισης;

```
interface X {
    void methodX();
}
class Y implements X {
    void methodX() {
        System.out.println("Method X");
    }
}
```



3. Ποια θα είναι η έξοδος του παρακάτω προγράμματος;

```
interface X {
    char c = 'A';
    char methodX();
}

class Y implements X {
    {
        System.out.println(c);
    }

    public char methodX(){
        char c = this.c;
        return ++c;
    }
}

public class MainClass {
    public static void main(String[] args) {
        Y y = new Y();
        System.out.println(y.methodX());
        System.out.println(y.c);
        System.out.println(X.c);
    }
}
```

4. Σε αυτήν την άσκηση θα χρειαστεί να ορίσετε κλάσεις που σχεδιάζουν διακριτά σχήματα στην οθόνη.

a. Τα σχήματα μπορεί να είναι τρίγωνα ή ορθογώνια.

b. Όλα τα σχήματα έχουν κάποιες κοινές ιδιότητες. Για παράδειγμα, κάθε ένα από αυτά θα έχει ένα offset (μετατόπιση) που δείχνει πόσο μακριά θα είναι από το αριστερό άκρο της οθόνης. Μπορούμε να προσθέσουμε τις μεθόδους getters/setters για το offset. Κάθε σχήμα θα έχει επίσης ένα μέγεθος, αλλά το μέγεθος ορισμένων σχημάτων περιγράφεται από ένα μοναδικό αριθμό, ενώ το μέγεθος των άλλων καθορίζεται από διάφορους αριθμούς. Δεδομένου ότι το μέγεθος θα καθορίζεται ανάλογα με το είδος του σχήματος, αυτό δεν είναι ένα κοινό χαρακτηριστικό που όλα τα σχήματα θα έχουν (Παράδειγμα: το μέγεθος ενός ορθογώνιου καθορίζεται από το μήκος και το πλάτος ενώ το μέγεθος ενός τριγώνου από τη βάση και το ύψος). **Ορίστε την κατάλληλη δομή για να περιγράψει σωστά όλα τα παραπάνω.**

c. Αν θα ήμασταν ευχαριστημένοι να καθορίσουμε τα offset και μέγεθος σε ένα κατασκευαστή του σχήματος, τότε η δομή που γράψατε νωρίτερα θα αρκούσε. Αλλά ας υποθέσουμε ότι θέλουμε επίσης να είμαστε σε θέση να επαναπροσδιορίσουμε το μέγεθος ενός υπάρχοντος σχήματος. Δηλαδή, θέλουμε μια μέθοδο set. Δεδομένου ότι το μέγεθος ενός σχήματος εξαρτάται από το σχήμα υπό επεξεργασία, **πως μπορούμε να υλοποιήσουμε αυτή τη λειτουργία;**



- d. Χρησιμοποιήστε της κλάσεις ShapeDemo και TreeDemo, για να δοκιμάσετε τις κλάσεις που γράψατε.
5. Τροποποιήστε την κλάση GeometricObject (από το εργαστήριο 11) για να υλοποιεί τη διαπροσωπία Comparable, και ορίστε μια στατική μέθοδο max στην κλάση GeometricObject η οποία βρίσκει το μεγαλύτερο των δύο αντικειμένων GeometricObject. Γράψτε ένα πρόγραμμα δοκιμών που χρησιμοποιεί τη μέθοδο max για να βρείτε το μεγαλύτερο δύο αντικειμένων Triangle.