



Ονοματεπώνυμο: Α.Μ.:

[54] **Θέμα 1**

Μία κλάση Pixel αναπαριστά τις συντεταγμένες ενός εικονοστοιχείου:

```
public class Pixel
{
    public static int MAX_X = 1280;
    public static int MAX_Y = 800;

    public int x, y;
}
```

Υποθέτουμε ότι το πρόγραμμά μας χειρίζεται εικόνες συγκεκριμένων διαστάσεων $MAX_X \times MAX_Y$ (π.χ. 1280×800), επομένως οι συντεταγμένες x και y κάθε σημείου πρέπει να είναι στα όρια $0 \leq x < MAX_X$ και $0 \leq y < MAX_Y$.

- [4] **α)** Τι σημαίνει το προσδιοριστικό `static` στις παραπάνω δηλώσεις;
Τι σημαίνει η απουσία του προσδιοριστικού `final` στις παραπάνω δηλώσεις;
- [5] **β)** Με βάση τον κώδικα που δίνεται, γράψτε κώδικα που να δημιουργεί ένα Pixel με όνομα `p` και συντεταγμένες (300, 200).
Επίσης γράψτε κώδικα που να αλλάζει το μέγιστο όριο της συντεταγμένης x σε 1024.

- [4] **γ)** Δίνεται η ακόλουθη μέθοδος (που υπάρχει εκτός της κλάσης Pixel):

```
public void setCoords(Pixel pix)
{
    // Δημιουργία ενός νέου Pixel και απόδοση σε αυτό συντεταγμένων
    // x=400, y=800
    Pixel temp;
    // ...

    pix = temp;
}
```

και η κλήση αυτής για το Pixel του προηγούμενου ερωτήματος

```
setCoords(p);
```

Ποιες οι τιμές των συντεταγμένων του `p` μετά από αυτή την κλήση, και γιατί;

- [5] **δ)** Υλοποιήστε μέθοδο `toString()` και χρησιμοποιήστε την για να εμφανίσετε στην οθόνη το παραπάνω εικονοστοιχείο στη μορφή «(300, 200)».

[5] **ε)** Προσθέστε στην κλάση `Pixel` μία μέθοδο

```
void setX(int x)
```

η οποία να αναθέτει την τιμή που δίνεται ως παράμετρος στη συντεταγμένη `x` του εικονοστοιχείου, μόνο εφόσον η τιμή αυτή είναι εντός ορίων, διαφορετικά να προκαλεί μία ελεγχόμενη εξαίρεση του τύπου `CoordinateOutOfBoundsException`.

[13] **στ)** Υποθέτοντας ότι στην κλάση `Pixel` υπάρχει και μία αντίστοιχη μέθοδος `setY()`, γράψτε πλήρη μέθοδο `main()` (εκτός της κλάσης `Pixel`) που να διαβάζει από το πληκτρολόγιο τις συντεταγμένες 10 (έγκυρων) εικονοστοιχείων και τα καταχωρίζει σε (απλό) πίνακα. Η εγκυρότητα των συντεταγμένων θα πρέπει να ελέγχεται μέσω των μεθόδων `setX()` και `setY()`.

[4] **ζ)** Αν εκτελεστεί ο κώδικας που γράψατε για το προηγούμενο ερώτημα και ο χρήστης δώσει ως είσοδο μία συμβολοσειρά αντί για κάποιον από τους ακέραιους που διαβάζονται, τι θα συμβεί, και γιατί; Αν υπάρξει πρόβλημα, περιγράψτε πώς θα μπορούσατε να το αποφύγετε.

[6] **η)** Υλοποιήστε την κλάση **ελεγχόμενης** εξαίρεσης `CoordinateOutOfBoundsException`. Τι θα έπρεπε να αλλάξετε για να γίνει **μη ελεγχόμενη** εξαίρεση; Ποια επίπτωση θα είχε αυτή η αλλαγή σε κώδικα που χρησιμοποιεί τη μέθοδο `setX()` και στην εκτέλεση του προγράμματος;

[8] **θ)** Γράψτε κώδικα με τον οποίο τα αντικείμενα του πίνακα του παραπάνω ερωτήματος εγγράφονται σε **δυναμικό σειριακό** αρχείο. Υπενθυμίζονται οι διάφορες κλάσεις εγγραφής σε αρχεία: `FileOutputStream`, `PrintWriter`, `ObjectOutputStream`, `RandomAccessFile`. Επιστημάνετε αν χρειάζονται αλλαγές στην κλάση `Pixel`.

[46] **Θέμα 2**

Δίνονται οι ακόλουθες δύο κλάσεις: η `Item` αντιπροσωπεύει ένα προϊόν που αγοράζει ο πελάτης ενός καταστήματος, ενώ η `Basket` αντιπροσωπεύει όλα τα προϊόντα που έχει αγοράσει.

[8] **α)** Κάποια στοιχεία της κλάσης `Basket` δεν έχουν χαρακτηρισμό `public` (γραμμές 25 και 39). Τι τύπου πρόσβαση υπάρχει σε αυτά; Συγκρίνετε αυτού του τύπου την πρόσβαση με αυτή που προσδίδουν τα χαρακτηριστικά `public`, `private` και `protected` και κατατάξτε τα τέσσερα είδη πρόσβασης ως προς το πόσο περιοριστικά είναι.

[8] **β)** Οι δύο σταθερές στην αρχή της κλάσης `Item` χρησιμοποιούνται για να διακριθεί το είδος του προϊόντος. Αντικαταστήστε τις σταθερές αυτές με μία απαρίθμηση (`enum`) και αναφέρετε αναλυτικά τις αλλαγές που πρέπει να γίνουν σε όλο τον κώδικα εξαιτίας αυτής της αντικατάστασης (δώστε αριθμούς γραμμών του κώδικα και τον νέο κώδικα ή επαρκή περιγραφή του).

[30] **γ)** Ο αρχικός κώδικας δεν ακολουθεί τις αρχές του αντικειμενοστρεφούς προγραμματισμού και μπορεί να βελτιωθεί χρησιμοποιώντας κληρονομικότητα. Ξαναγράψτε τον κώδικα

- αποκρύπτοντας τα πεδία των κλάσεων,
- προσθέτοντας στην κλάση `Item` μία μέθοδο `value()` που να κάνει, για κάθε **υποκλάση** της που θα δημιουργήσετε, τους υπολογισμούς που τώρα γίνονται στη μέθοδο `itemValue()` της κλάσης `Basket` επιστρέφοντας την αξία του αντίστοιχου προϊόντος,
- αντικαθιστώντας τον απλό πίνακα `items` της κλάσης `Basket` από κατάλληλο `ArrayList<>` και εκμεταλλευόμενοι τις **δυνατότητές** του.

```

1 public class Item
2 {
3     public static final int UNIT_ITEM = 0;
4     public static final int WEIGHT_ITEM = 1;
5
6     public int type;
7     public String description;
8     public double quantity;
9     public double price;
10
11     public Item(int type, String description, double quantity, double price)
12     {
13         this.type = type;
14         this.description = description;
15         this.quantity = quantity;    // Only for WEIGHT_ITEMS, see Basket.itemValue()
16         this.price = price;
17     }
18 }
19
20 //////////////////////////////////////
21
22 public class Basket
23 {
24     Item[] items;
25     int itemCount;
26
27     public Basket()
28     {
29         items = new Item[100];    // Big enough to hold everything, hopefully!
30         itemCount = 0;
31     }
32
33     public void addItem(Item item)
34     {
35         items[itemCount] = item;
36         itemCount++;
37     }
38
39     double itemValue(Item item)
40     {
41         if (item.type == item.UNIT_ITEM) {
42             return item.price;
43         }
44         else {    // in this case, item.type == item.WEIGHT_ITEM
45             return item.price * item.quantity;
46         }
47     }
48 }

```

```
49     public double totalValue()
50     {
51         double sum = 0.0;
52         for (int i = 0; i < itemCount; i++) {
53             sum += itemValue(items[i]);
54         }
55
56         return sum;
57     }
58 }
```