

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής
15 Ιουλίου 2015

- Διάρκεια 2:00 ώρες
- Να απαντηθούν **ΟΛΑ** (5) τα θέματα.
- Καλή επιτυχία.

Θέμα 1^ο

Να υλοποιηθεί η στατική μέθοδος `blackWhiteTable(int r, int c)` η οποία δημιουργεί και επιστρέφει ένα δισδιάστατο διάνυσμα (r γραμμών και c στηλών) από `Strings`. Το στοιχείο σε κάθε θέση να προκύπτει με βάση το άθροισμα της γραμμής και της στήλης: εάν το άθροισμα είναι άρτιο έχει την τιμή "white", εάν είναι περιττό την τιμή "black"

Θέμα 2^ο

Να υλοποιηθεί η στατική μέθοδος `smallestElements(int[] a, int k)` η οποία δέχεται ως παραμέτρους ένα διάνυσμα από διακριτούς ακεραίους (έστω a) και έναν ακέραιο (έστω k) και επιστρέφει ένα διάνυσμα αποτελούμενο από τα k μικρότερα στοιχεία του διανύσματος a .

Θέμα 3^ο (Κλάσεις και υλοποιήσεις τους)

Δίνεται η κλάση `Point` η οποία μοντελοποιεί ένα σημείο στο επίπεδο (καθορισμένο από δύο ακέραιες συντεταγμένες) και υποστηρίζει τις μεθόδους:

- | | |
|--|---|
| 1. <code>Point(int x, int y)</code> | Κατασκευάζει το σημείο (x,y) |
| 2. <code>void setX(int x)</code> | Θέτει/μετατρέπει την X-συντεταγμένη του σημείου |
| 3. <code>void setY(int y)</code> | Θέτει/μετατρέπει την Y-συντεταγμένη του σημείου |
| 4. <code>int getX()</code> | Επιστρέφει την X-συντεταγμένη του σημείου |
| 5. <code>int getY()</code> | Επιστρέφει την Y-συντεταγμένη του σημείου |
| 6. <code>double distanceFrom(Point p)</code> | Επιστρέφει την απόσταση από το σημείο p . |

Να υλοποιηθεί η κλάση `LS` (ευθύγραμμο τμήμα) η οποία έχει σκοπό να μοντελοποιήσει ένα ευθύγραμμο τμήμα. Το κάθε ευθύγραμμο τμήμα ορίζεται από τα δύο άκρα του (σημεία στο επίπεδο, τύπου `Point` στις δύο διαστάσεις).

Αντικείμενα της κλάσης `LS` υποστηρίζουν τις παρακάτω μεθόδους:

1. <code>LS(Point a, Point b)</code>	Κατασκευάζει ένα ευθύγραμμο τμήμα με άκρα τα σημεία a και b .
2. <code>getFirstEndPoint()</code>	Επιστρέφει το πρώτο άκρο (δηλ. το σημείο a) του ευθύγραμμου τμήματος.
3. <code>getSecondEndPoint()</code>	Επιστρέφει το δεύτερο άκρο (δηλ. το σημείο b) του ευθύγραμμου τμήματος.
3. <code>length()</code>	Επιστρέφει το μήκος του ευθύγραμμου τμήματος.
4. <code>isParallelToAxes()</code>	Επιστρέφει <code>true</code> εάν το ευθύγραμμο τμήμα είναι παράλληλο με τον άξονα X ή άξονα Y, <code>false</code> διαφορετικά.

Θέμα 4^ο (Διαπροσωπείες και υλοποιήσεις τους)

Η διαπροσωπεία `IntQueue` μοντελοποιεί μία ουρά ακεραίων, δηλαδή μία δομή δεδομένων που αποτελείται από μία ακολουθία ακεραίων (τον ένα πίσω από τον άλλο, τον πρώτο στην θέση "0", τον δεύτερο στη θέση "1", κ.ο.κ.) και υποστηρίζει τις παρακάτω μεθόδους:

<code>void add(int elem, int pos)</code>	Εισάγει τον στοιχείο <code>elem</code> στη θέση <code>pos</code> της ουράς, μετατοπίζοντας τα στοιχεία που βρίσκονταν πριν την εισαγωγή στην θέση <code>pos</code> και μετά, κατά μία θέση. Υποθέτει ότι η ουρά δεν είναι γεμάτη και ότι $0 \leq pos \leq n$ όπου n το πλήθος των στοιχείων της ουράς.
<code>int remove()</code>	Διαγράφει και επιστρέφει το πρώτο στοιχείο, δηλαδή αυτό που βρίσκεται στην <u>αρχή</u> της ουράς. Υποθέτει ότι η ουρά δεν είναι άδεια.
<code>boolean isFull()</code>	Ελέγχει εάν η ουρά είναι γεμάτη.
<code>boolean isEmpty()</code>	Ελέγχει εάν η ουρά είναι άδεια.
<code>int size()</code>	Επιστρέφει το πλήθος των στοιχείων στην ουρά.

Να αναπτυχθεί κώδικας για την κλάση `ArrayIntQueue` η οποία υλοποιεί τη διαπροσωπεία `IntQueue` ως ένα διάνυσμα από ακεραίους. Το μέγεθος της ουράς, και κατά συνέπεια του διανύσματος, να δοθεί ως παράμετρος στον κατασκευαστή της `ArrayIntQueue`.

Θέμα 5^ο

Δίνεται η κλάση `Student` (φοιτητής) η οποία χρησιμοποιείται στην μοντελοποίηση ενός εργαστηριακού τμήματος κάποιου μαθήματος. Κάθε αντικείμενο της κλάσης `Student` υλοποιεί τις μεθόδους:

<code>Student(String name, String id)</code>	Κατασκευαστής. Θέτει το όνομα και τον αριθμό μητρώου κάθε επαφής.
<code>void setName(String newName)</code>	Θέτει το όνομα του φοιτητή.
<code>void setID(String newID)</code>	Θέτει τον αριθμό μητρώου του φοιτητή.
<code>String getName()</code>	Επιστρέφει το όνομα του φοιτητή.
<code>String getID()</code>	Επιστρέφει τον αριθμό μητρώου του φοιτητή.
<code>String toString()</code>	Εκτυπώνει τον φοιτητή (σε μία γραμμή εξόδου)

Δίνεται ο (όχι πλήρης) κώδικας για την κλάση `LabGroup` η οποία υλοποιεί το εργαστηριακό τμήμα (με το πολυμορφικό) χρησιμοποιώντας ένα μονοδιάστατο διάνυσμα. Η κλάση `LabGroup` έχει τις παρακάτω μεθόδους:

<code>LabGroup()</code>	Κατασκευαστής. Δημιουργεί ένα εργαστηριακό τμήμα.
<code>void insert(Student s)</code>	Εισάγει τον φοιτητή <code>s</code> στο εργαστηριακό τμήμα.
<code>void printStudents()</code>	Τυπώνει τους φοιτητές στο παράθυρο εξόδου
<code>int size()</code>	Επιστρέφει το μέγεθος του τμήματος.
<code>boolean isFull()</code>	Ελέγχει εάν το τμήμα είναι πλήρες.
<code>boolean isInLabGroup(String id)</code>	Ελέγχει εάν ο φοιτητής ανήκει στο τμήμα.
<code>Iterator iter();</code>	Επιστρέφει έναν "προσπελαστή" (iterator) προς τους φοιτητές του εργαστηριακού τμήματος. ΝΑ ΥΛΟΠΟΙΗΘΕΙ.

Να υλοποιηθεί η μέθοδος `iter()`. Να συμπληρωθεί ο υπάρχων κώδικας με ότι επιπλέον στοιχεία χρειάζονται. Δίνεται η διαπροσωπεία `Iterator` (στο τέλος της σελίδας).

```
01 public class LabGroup {
02     private static int groupNumber=0;
03     private String groupID;
04     private Student data[];
05     private int size;
06
07     public LabGroup() {
08         groupNumber++;
09         groupID="Group"+groupNumber;
10         data = new Student[25];
11         size=0;
12     }
13
14     public void insert(Student s) {
15         data[size]=s;
16         size++;
17     }
18
19     public int size() {
20         return size;
21     }
22
23     public boolean isFull() {
24         return (size==25);
25     }
26
27     public void printStudents() {
28         System.out.println("Lab Group:" + groupID);
29         System.out.println();
30         for (int i=0; i<size; i++)
31             System.out.println(data[i]);
32     }
33
34     public boolean isInLabGroup(String id) {
35         for (int i=0; i<size; i++)
36             if (data[i].getID().equals(id))
37                 return true;
38         return false;
39     }
40
41 }//class LabGroup
```

```
interface Iterator
```

```
{
    boolean hasNext() //Returns true if the iteration has more elements, false otherwise.
    Object next() //Returns the next element of the Iteration
}
```