

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Τομέας Μαθηματικών

## Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής 7 Οκτωβρίου 2009

- Διάρκεια 2:30 ώρες
- Να απαντηθούν **ΟΛΑ** (5) τα θέματα.
- Δίνεται η περιγραφή των κλάσεων:
- Καλή επιτυχία.

### Θέμα 1<sup>ο</sup>

Να υλοποιηθεί η στατική μέθοδος `log10Approx()` η οποία δέχεται ως παράμετρο ένα θετικό ακέραιο  $x$  και «προσεγγίζει» τον λογάριθμό του με βάση το 10, υπολογίζει και επιστρέφει δηλαδή έναν ακέραιο  $y$  τέτοιον ώστε να ισχύει ότι  $10^y \leq x < 10^{y+1}$ . Να μην χρησιμοποιηθούν μέθοδοι βιβλιοθήκης. Εάν χρησιμοποιήσετε κάποια άλλη μέθοδο (πχ. ύψωση σε δύναμη) πρέπει δώσετε και τη υλοποίησή της.

### Θέμα 2<sup>ο</sup>

Να υλοποιηθεί η κλάση `Triangle` η οποία έχει σκοπό να μοντελοποιήσει ένα τρίγωνο. Το κάθε ευθύγραμμο τμήμα ορίζεται από τα σημεία των άκρων του (τύπου `Point` στις δύο διαστάσεις). Αντικείμενα της κλάσης `Triangle` υποστηρίζουν τις παρακάτω μεθόδους:

1. <code>Triangle(Point a, Point b, Point c)</code>	Κατασκευάζει ένα τρίγωνο με κορυφές τα σημεία $a$ , $b$ και $c$ .
2. <code>perimeter()</code>	Επιστρέφει την περίμετρο του τριγώνου.
3. <code>translate(Boolean direction, int dist)</code>	Μετατοπίζει το τρίγωνο οριζόντια (εάν <code>direction==true</code> ) ή κάθετα (εάν <code>direction==false</code> ) κατά <code>dist</code> μονάδες μήκους.
4. <code>toString()</code>	Επιστρέφει το τρίγωνο σε εκτυπώσιμη μορφή (ως <code>String</code> )

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

1. `Point(int x, int y)` Κατασκευάζει το σημείο  $(x,y)$
2. `setX(int x)` Θέτει/μετατρέπει την  $X$ -συντεταγμένη του σημείου
3. `setY(int y)` Θέτει/μετατρέπει την  $Y$ -συντεταγμένη του σημείου
4. `getX()` Επιστρέφει την  $X$ -συντεταγμένη του σημείου
5. `getY()` Επιστρέφει την  $Y$ -συντεταγμένη του σημείου
6. `distanceFrom(Point p)` Επιστρέφει την απόσταση το σημείο από το σημείο που δίνεται ως παράμετρος.
7. `toString()` Επιστρέφει το σημείο σε εκτυπώσιμη μορφή (ως `string`)

### Θέμα 3<sup>ο</sup>

Να γραφεί η στατική μέθοδος `has2by2ZeroBlock()` η οποία δέχεται ως παράμετρο ένα δισδιάστατο διάνυσμα (μεγαλύτερο ή ίσο με  $2 \times 2$ ) από ακέραιους και ελέγχει εάν το διάνυσμα περιέχει ως υπο-πίνακα ένα  $2 \times 2$  πίνακα από μηδενικά. Εάν τέτοιος υπο-πίνακας υπάρχει, η μέθοδος εκτυπώνει τις συντεταγμένες του άνω-αριστερου στοιχείου του, αλλιώς εκτυπώνει σχετικό μήνυμα.

### Θέμα 4<sup>ο</sup> (Κληρονομικότητα, διαπρωσωπείες)

Υποθέστε ότι χρειάζεται να υλοποιήσουμε ένα πρόγραμμα που διαχειρίζεται κύκλους και ορθογώνια παραλληλόγραμμα. Το πρόγραμμα εκτελεί βασικές πράξεις για κάθε ένα από τα σχήματα αυτά. Οι πράξεις αυτές περιλαμβάνουν την τοποθέτησή τους στο επίπεδο και την κίνησή τους. Επιπλέον, θέλουμε να μπορούμε να υπολογίσουμε το εμβαδόν κάθε σχήματος. Τέλος, για κάθε σχήμα θέλουμε η μέθοδος `scale(double s)` να αυξομειώνει το μέγεθος κάθε σχήματος (π.χ., εάν  $s=2$  το «μέγεθος» του σχήματος διπλασιάζεται, εάν  $s=0.5$  το «μέγεθος» του σχήματος μειώνεται στο μισό).

Αντικείμενα που καθορίζονται από ένα σημείο αναφοράς και μπορούν να κινηθούν στις δυο διαστάσεις υλοποιούν την διαπρωσωπεία `Movable` η οποία περιλαμβάνει την μέθοδο `moveTo(int x, int y)` η οποία μεταφέρει το σημείο αναφοράς του αντικειμένου στο σημείο  $(x,y)$ .

Ορίζουμε τις κλάσεις `Circle` και `Rectangle`. Για να ελαχιστοποιήσουμε τον παραγόμενο κώδικα, πρώτα υλοποιούμε την κλάση `Shape` και με βάση αυτή ορίζουμε τις `Circle` και `Rectangle` σαν υποκλάσεις της.

Κάθε σχήμα έχει:

- Τη θέση του. Όλα τα σχήματα έχουν  $X$  και  $Y$  συντεταγμένες.
- Τις διαστάσεις του. Ο κύκλος ακτίνα, το ορθογώνιο βάση και ύψος.
- Κίνηση. Όλα τα σχήματα μπορούν να κινηθούν. Η κίνησή τους μεταβάλλει τις συντεταγμένες τους.
- Εμβαδόν. Κάθε σχήμα είναι τοποθετημένο στο επίπεδο και καταλαμβάνει εμβαδόν.

Να γραφεί κώδικας για την διαπροσωπεία `Movable`, την κλάση `Shape` και τις υποκλάσεις της `Circle` και `Rectangle`.

### Θέμα 5<sup>ο</sup>

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>	Επιστρέφει τον φοιτητή (σε μία γραμμή εξόδου) ως <code>String</code>

Να γραφεί κώδικας για την κλάση `LabGroup` η οποία υλοποιεί το εργαστηριακό τμήμα χρησιμοποιώντας ένα μονοδιάστατο διάνυσμα. Δίνεται ότι το μέγεθος ενός εργαστηριακού τμήματος δεν υπερβαίνει τους 25 φοιτητές. Η κλάση `LabGroup` που θα αναπτύξετε 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.