

Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Μαθηματικών

## Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής 9 Οκτωβρίου 2014

- Διάρκεια 2:00 ώρες
- Να απαντηθούν **ΟΛΑ (4) τα θέματα.**
- Καλή επιτυχία.

### Θέμα 1<sup>ο</sup>

Να υλοποιηθεί η στατική μέθοδος `factorialGreaterThan()` η οποία δέχεται ως παράμετρο ένα θετικό ακέραιο  $k$  και υπολογίζει τον μικρότερο αριθμό  $n$  για τον οποίον ισχύει ότι  $n! \geq k$ .

Σημείωση: Το  $n!$  διαβάζεται ως " $n$  παραγοντικό". Εξ ορισμού,  $n! = 1 * 2 * \dots * n$ ,  $n \geq 1$ .

### Θέμα 2<sup>ο</sup>

Να δημιουργηθεί η στατική μέθοδος `matrixMultiply()` η οποία δέχεται ως παραμέτρους 2 πίνακες (δυσδιάστατα διανύσματα) από πραγματικούς αριθμούς, έστω  $a$  και  $b$ , και δημιουργεί και επιστρέφει ως αποτέλεσμα ένα νέο πίνακα, ίσο με το γινόμενο τους. Υποθέτουμε ότι οι πίνακες  $a$  και  $b$  είναι συμβατοί μεταξύ τους ως προς τον πολλαπλασιασμό πινάκων, δηλαδή, ο αριθμός των στηλών του πίνακα  $a$  είναι ίσος με τον αριθμό των γραμμών του πίνακα  $b$ .

### Θέμα 3<sup>ο</sup> (Κληρονομικότητα, διαπροσωπείες)

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 διαπροσωπεία `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 4<sup>ο</sup> (Διαπροσωπείες και υλοποιήσεις τους)

Η διαπροσωπεία `FIFOQueue` μοντελοποιεί μία ουρά αντικειμένων, δηλαδή μία δομή δεδομένων που αποτελείται από ένα σύνολο αντικειμένων (επιτρέπονται οι επαναλήψεις) όπου τα αντικείμενα εξέρχονται από την ουρά ακριβώς με την σειρά την οποία εισέρχονται. (Η ονομασία FIFO προκύπτει από τα αρχικά των λέξεων `First-In First-Out`). Η διαπροσωπεία `FIFOQueue` υποστηρίζει τις παρακάτω μεθόδους:

<code>void insertBack(Object elem)</code>	Εισάγει το αντικείμενο <code>elem</code> στο τέλος της FIFO ουράς. Υποθέτει ότι η FIFO ουρά δεν είναι γεμάτη.
<code>Object deleteFront()</code>	Διαγράφει και επιστρέφει το αντικείμενο στην αρχή της FIFO ουράς. Υποθέτει ότι η FIFO ουρά δεν είναι άδεια.
<code>int size()</code>	Επιστρέφει τον αριθμό των αντικειμένων που βρίσκονται στην FIFO ουρά.
<code>boolean isFull()</code>	Ελέγχει εάν η FIFO ουρά είναι γεμάτη.

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