

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής 10 Ιουλίου 2004

- Διάρκεια 3 ώρες
- Να απαντηθούν **ΟΛΑ (5)** τα θέματα.
- Δίνεται η περιγραφή των: **ArrayList**, **Iterator**, και **Comparable**.
- Καλή επιτυχία.

Θέμα 1^ο

Να δημιουργηθεί η στατική μέθοδος `sumRowCol(int r, int c)` η οποία δημιουργεί και επιστρέφει ένα δισδιάστατο διάνυσμα (r γραμμών και c στηλών) από ακεραίους. Το στοιχείο σε κάθε θέση να προκύπτει ως το άθροισμα της γραμμής και της στήλης.

Θέμα 2^ο

Να γραφεί κώδικας Java για την στατική μέθοδο `firstDegreeEquation` (πρωτοβάθμια εξίσωση) η οποία λύνει την πρωτοβάθμια εξίσωση $ax+b=0$. Η μέθοδος δέχεται δύο παραμέτρους οι οποίες αντιστοιχούν στις τιμές των συντελεστών a και b και τυπώνει στο παράθυρο εξόδου την τιμή του x που ικανοποιεί την εξίσωση. Στην περίπτωση όπου η λύση της εξίσωσης είναι αδύνατη ($a=0$ και $b\neq 0$) ή αόριστη ($a=0$ και $b=0$) τυπώνονται ανάλογα μηνύματα.

Θέμα 3^ο (Χρήση συλλογών δεδομένων)

Δίνεται η κλάση `Contact` (επαφή) η οποία χρησιμοποιείται στην μοντελοποίηση μίας απλής ηλεκτρονικής ατζέντας. Κάθε αντικείμενο της κλάσης `Contact` υλοποιεί τις μεθόδους:

<code>Contact(String name, String phone)</code>	Κατασκευαστής. Θέτει το όνομα και το τηλέφωνο κάθε επαφής.
<code>void setName(String newName)</code>	Θέτει το όνομα της επαφής.
<code>void setPhone(String newPhone)</code>	Θέτει το τηλέφωνο της επαφής.
<code>String getName()</code>	Επιστρέφει το όνομα της επαφής.
<code>String getPhone()</code>	Επιστρέφει το τηλέφωνο της επαφής.
<code>String toString()</code>	Εκτυπώνει την επαφή (σε μία γραμμή εξόδου)

Να γραφεί κώδικας για την κλάση `PhoneBook` η οποία υλοποιεί την ηλεκτρονική ατζέντα χρησιμοποιώντας ένα αντικείμενο (συλλογής δεδομένων) τύπου `ArrayList`. Η περιγραφή της κλάσης `ArrayList` δίδεται. Η κλάση `PhoneBook` που θα αναπτύξετε έχει τις παρακάτω μεθόδους.

<code>PhoneBook()</code>	Κατασκευαστής. Δημιουργεί μία άδεια ατζέντα.
<code>void insert(Contact c)</code>	Εισάγει την επαφή c στην ατζέντα.
<code>void printContacts()</code>	Τυπώνει τις επαφές στο παράθυρο εξόδου

Για την υλοποίηση της μεθόδου `printContacts()` να γίνει χρήση διαπροσωπελαστών (**Iterators**). Η περιγραφή της διαπροσωπίας `Iterator` δίδεται.

Θέμα 4^ο (Διαπροσωπίες)

Η κλάση `Person` έχει ως σκοπό την μοντελοποίηση της οντότητας «άνθρωπος». Κάθε άνθρωπος (αντικείμενο που δημιουργήθηκε με βάση την κλάση `Person`) έχει δύο πεδία. Το όνομα του (τύπου `String`) και την ηλικία του (τύπου `int`). Κάθε αντικείμενο τύπου `Person` πρέπει να μπορεί να συγκριθεί με άλλο αντικείμενο του ίδιου τύπου, να υλοποιεί δηλαδή την διαπροσωπία `Comparable`. Η σύγκριση δύο αντικειμένων ανθρώπων γίνεται με βάση την ηλικία τους.

Να γραφεί κώδικας για την κλάση `Person` η οποία υλοποιεί την διαπροσωπία `Comparable` και περιλαμβάνει:

- δηλώσεις των πεδίων της κλάσης,
- έναν κατασκευαστή για την κλάση,
- ότι άλλο κρίνεται απαραίτητο ώστε ένα αντικείμενο τύπου `Person` να μπορεί να συγκριθεί με άλλο ομοειδές αντικείμενο.

Η κλάση `Person` δεν περιλαμβάνει `set` και `get` μεθόδους για τα πεδία τους. Η περιγραφή της διαπροσωπίας `Comparable` δίδεται.

Γυρίστε σελίδα...

Θέμα 5^ο (Κληρονομικότητα)

Υποθέστε ότι χρειάζεται να υλοποιήσουμε ένα πρόγραμμα που διαχειρίζεται κύκλους, ορθογώνια παραλληλόγραμμα και κυλίνδρους. 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 (το εμβαδόν του κυλίνδρου θεωρείται ίσο με το εμβαδόν της κυκλικής βάσης του), καθώς και τον όγκο του. Τέλος, για κάθε σχήμα θέλουμε η μέθοδος `toString` (της κλάσης `Object`) να επιστρέφει ένα περιγραφικό μήνυμα με τα χαρακτηριστικά του κάθε σχήματος.

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

Κάθε σχήμα έχει:

- **Τη θέση του.** Όλα τα σχήματα έχουν X και Y συντεταγμένες. (υποθέτουμε ότι ο κύλινδρος είναι πάντοτε τοποθετημένος «όρθιος»)
- **Κίνηση.** Όλα τα σχήματα μπορούν να κινηθούν. Η κίνησή τους μεταβάλλει τις συντεταγμένες τους.
- **Εμβαδόν.** Κάθε σχήμα είναι τοποθετημένο στο επίπεδο και καταλαμβάνει εμβαδόν. (Υποθέτουμε ότι ο κύλινδρος είναι πάντοτε τοποθετημένος όρθιος).
- **Φραστική Περιγραφή.** Κάθε σχήμα αποκρίνεται στο μήνυμα `toString` επιστρέφοντας τα χαρακτηριστικά του γνωρίσματα (τύπος σχήματος, θέση, διαστάσεις)

Επιπρόσθετα οι κύλινδροι έχουν:

- **Όγκο.** Κάθε κύλινδρος καταλαμβάνει όγκο που καθορίζεται από τις διαστάσεις του (ακτίνα βάσης και ύψος)

Να υλοποιηθούν η κλάση `Shape` και οι υποκλάσεις της `Circle`, `Rectangle`, `Cylinder`.