

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής 5 Ιουλίου 2013

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

Όνοματεπώνυμο:

A. Μητρώου:

Θέμα 1°

Έστω ένας μονοδιάστατος πίνακας από λογικές μεταβλητές. Ένα «plateau» είναι ένα σύνολο από συνεχόμενες θέσεις του πίνακα που περιέχουν την τιμή «true». Να δημιουργηθεί η στατική μέθοδος `findLargestPlateau(boolean m[])` η οποία δέχεται ως παράμετρο ένα μονοδιάστατο πίνακα από boolean και εκτυπώνει το μήκος του μεγαλύτερου plateau, τη θέση που αρχίζει και τη θέση στην οποία τελειώνει.

Θέμα 2°

Αίεται η κλάση `Employee` (μισθωτός) η οποία χρησιμοποιείται στην μοντελοποίηση ενός μισθωτού. Κάθε αντικείμενο της κλάσης `Employee` υλοποιεί τις μεθόδους:

<code>Employee(String name, int salary, boolean gender)</code>	Κατασκευαστής. Θέτει το όνομα, το μισθό και το φύλλο (TRUE για αρσενικό, FALSE για θηλυκό) του/της μισθωτού.
<code>void setName(String newName)</code>	Θέτει το όνομα.
<code>void setSalary(int newSalary)</code>	Θέτει το μισθό.
<code>void setGender(boolean isMale)</code>	Θέτει το φύλλο.
<code>String getName()</code>	Επιστρέφει το όνομα.
<code>int getSalary()</code>	Επιστρέφει το μισθό.
<code>boolean getGender()</code>	Επιστρέφει το φύλλο.
<code>String toString()</code>	Εκτυπώνει το μισθωτό (σε μία γραμμή εξόδου)

Να γραφεί η στατική μέθοδος `averageMaleSalary()` η οποία δέχεται ως παράμετρο ένα μονοδιάστατο πίνακα από αντικείμενα τύπου `Employee` και επιστρέφει το μέσο όρο των μισθών των ανδρών του πίνακα.

Θέμα 3°

Έστω ένας διδιάστατος τετραγωνικός πίνακας `m` από ακέραιους κάθε στοιχείο του οποίου αρχικά περιέχει τους αριθμούς «0» ή «-1». Να δημιουργηθεί η στατική μέθοδος `countMines(int m[][])` η οποία αντικαθιστά τα μηδενικά με το πλήθος των `-1` που βρίσκονται στις γειτονικές θέσεις του, δηλαδή έναν ακέραιο στο διάστημα `[0..8]`.

Σημείωση: Ένα στοιχείο του πίνακα που αρχικά περιέχει «0» και δεν έχει γείτονες που περιέχουν «-1», μετά την επεξεργασία θα εξακολουθεί να περιέχει το στοιχείο «0».

Θέμα 4^ο (Κλάσεις και υλοποιήσεις τους)

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

- | | |
|-------------------------|---|
| 1. Point(int x, int y). | Κατασκευάζει το σημείο (x,y) |
| 2. setX(int x) | Θέτει/μετατρέπει την Χ-συντεταγμένη του σημείου |
| 3. setY(int y) | Θέτει/μετατρέπει την Υ-συντεταγμένη του σημείου |
| 4. getX() | Επιστρέφει την Χ-συντεταγμένη του σημείου |
| 5. getY() | Επιστρέφει την Υ-συντεταγμένη του σημείου |

Δίνεται η κλάση **Segment** η οποία μοντελοποιεί ένα ευθύγραμμο τμήμα στο επίπεδο (καθορισμένο από δύο σημεία (Point)) και υποστηρίζει τις μεθόδους:

- | | |
|---------------------------------|--|
| 1. Segment(Point p1, Point p2). | Κατασκευάζει το ευθύγραμμο τμήμα με άκρα τα p1 και p2. |
| 2. setP1(Point newPoint) | Θέτει/μετατρέπει το πρώτο σημείο του ευθύγραμμου τμήματος. |
| 3. setP2(Point newPoint) | Θέτει/μετατρέπει το δεύτερο σημείο του ευθύγραμμου τμήματος. |
| 4. getP1() | Επιστρέφει το πρώτο σημείο του ευθύγραμμου τμήματος. |
| 5. getP2() | Επιστρέφει το δεύτερο σημείο του ευθύγραμμου τμήματος. |
| 6. length() | Επιστρέφει το μήκος του ευθύγραμμου τμήματος |

Να υλοποιηθεί η κλάση **Circle** η οποία έχει σκοπό να μοντελοποιήσει ένα κύκλο. Ο κάθε κύκλος ορίζεται από το κέντρο του (τύπου **Point**) και την ακτίνα του (τύπου **double**). Η κλάση **Circle** υλοποιεί τη διαπροσωπεία Comparable (δίνεται στο τέλος της άσκησης). Έναν κύκλο θεωρείται μεγαλύτερος από έναν άλλο εάν έχει μεγαλύτερο εμβαδόν.

Αντικείμενα της κλάσης **Circle** υποστηρίζουν τις παρακάτω μεθόδους:

1. Circle(Segment s)	Κατασκευάζει ένα κύκλο ακτίνα το ευθύγραμμο τμήμα s. Κέντρο του κύκλου είναι το πρώτο άκρο του τμήματος s.
2. getCenter()	Επιστρέφει το κεντρο του κύκλου (τύπου Point).
3. getRadius()	Επιστρέφει την ακτίνα του κύκλου (τύπου double).

```
interface Comparable
{
    int compareTo(Object o) // Επιστρέφει -1, 0 ή 1 εάν το αντικείμενο μικρότερο, ίσο ή μεγαλύτερο
                          // από αυτό που δίνεται ως παράμετρος.
}
```

Θέμα 5^ο (Διαπροσωπείες και υλοποιήσεις τους)

Η διαπροσωπεία **ObjectQueue** μοντελοποιεί μία **ουρά αναμονής αντικειμένων**, δηλαδή μία δομή δεδομένων που αποτελείται από μία ακολουθία αντικειμένων (το ένα πίσω από το άλλο) και υποστηρίζει τις παρακάτω μεθόδους:

void add(Object elem)	Εισάγει το αντικείμενο elem στο <u>τέλος</u> της ουράς αναμονής. Υποθέτει ότι η ουρά αναμονής δεν είναι γεμάτη.
Object remove()	Διαγράφει και επιστρέφει το πρώτο στοιχείο, δηλαδή αυτό που βρίσκεται στην <u>αρχή</u> , της ουράς αναμονής. Υποθέτει ότι η ουρά αναμονής δεν είναι άδεια.
boolean isFull()	Ελέγχει εάν η ουρά αναμονής είναι γεμάτη.
boolean isEmpty()	Ελέγχει εάν η ουρά αναμονής είναι άδεια.

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