

Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Μαθηματικών

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής  
11 Ιουλίου 2012

- Διάρκεια 2:30 ώρες
- Να απαντηθούν **ΟΛΑ (5)** τα θέματα.
- Καλή επιτυχία.

Όνοματεπώνυμο:	A. Μητρώου:
----------------	-------------

**Θέμα 1<sup>ο</sup>**

Να δημιουργηθεί η στατική μέθοδος `symmetricMatrix(int r)` η οποία δημιουργεί και επιστρέφει ένα δισδιάστατο τετραγωνικό πίνακα ( $r$  γραμμών και  $r$  στηλών) από ακέραιους έτσι ώστε: α) τα στοιχεία της κύριας διαγωνίου να είναι ίσα με το «0», β) τα στοιχεία κάτω-αριστερά από την κύρια διαγώνιο να είναι ίσα με το «-1», και γ) τα στοιχεία πάνω-δεξιά από την κύρια διαγώνιο να είναι ίσα με το «1».

**Θέμα 2<sup>ο</sup>**

Δίνεται 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)</code>	Κατασκευαστής. Θέτει το όνομα και το μισθό του μισθωτού.
<code>void setName(String newName)</code>	Θέτει το όνομα.
<code>void setSalary(int newSalary)</code>	Θέτει το μισθό.
<code>String getName()</code>	Επιστρέφει το όνομα.
<code>int getSalary()</code>	Επιστρέφει το μισθό.
<code>String toString()</code>	Εκτυπώνει το μισθωτό (σε μία γραμμή εξόδου)

Να γραφεί η στατική μέθοδος `maximumSalary()` η οποία δέχεται ως παράμετρο ένα διάνυσμα από αντικείμενα τύπου `Employee` και εκτυπώνει το μισθωτό (όνομα και μισθό) με το μεγαλύτερο μισθό.

**Θέμα 3<sup>ο</sup>**

Το κόσκινο του Ερατοσθένη είναι μία μέθοδος που υπολογίζει τους πρώτους αριθμούς που είναι μικρότεροι ή ίσοι από έναν θετικό ακέραιο  $max$ . Η μέθοδος χρησιμοποιεί ένα σύνολο από ακέραιους που αρχικά περιέχει όλους τους ακέραιους από το 1 έως και το  $max$ , και εκτελεί τα ακόλουθα βήματα:

1. Σβήσε το 1 από το σύνολο
2. Βρες τον επόμενο ακέραιο που ανήκει στο σύνολο και δεν έχει εξεταστεί μέχρι τώρα, έστω  $N$ , και σβήσε όλα τα πολλαπλάσια του ( $2N, 3N, \dots$ ) από το σύνολο. Προσοχή! Ο  $N$  παραμένει στο σύνολο.
3. Συνέχισε με το βήμα 2 έως ότου εξεταστούν όλοι οι αριθμοί που ανήκουν στο σύνολο.

Να υλοποιηθεί η στατική μέθοδος `eratosthenesSieve()` η οποία δέχεται ως παράμετρο ένα θετικό ακέραιο  $max$  και εκτυπώνει όλους τους πρώτους αριθμούς στο διάστημα  $1..max$ .

Το σύνολο των ακεραίων στο διάστημα  $[1..max]$  να υλοποιηθεί ως ένα διάνυσμα, έστω  $A$ , από ακέραιους όπου  $A[i] = i$  εάν ο ακέραιος  $i$  ανήκει στο σύνολο, διαφορετικά  $A[i] = 0$ .

**Θέμα 4<sup>ο</sup>** (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. 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       |

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**).

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

1. Circle(Point a, double radius)	Κατασκευάζει ένα κύκλο με κέντρο το <b>a</b> και ακτίνα <del>radius</del> <i>radius</i>
2. getCenter()	Επιστρέφει το κέντρο του κύκλου (τύπου <b>Point</b> ).
3. getRadius()	Επιστρέφει την ακτίνα του κύκλου.
4. isOnCircle(Point p)	Επιστρέφει <b>true</b> εάν το σημείο <b>p</b> βρίσκεται επάνω στον κύκλο, <b>false</b> διαφορετικά.
4. containsPoint(Point p)	Επιστρέφει <b>true</b> εάν ο κύκλος περιέχει το σημείο <b>p</b> στο εσωτερικό του, <b>false</b> διαφορετικά.
5. intersectsSegment(Segment s)	Επιστρέφει <b>true</b> εάν ο κύκλος τέμνει το ευθύγραμμο τμήμα <b>s</b> , <b>false</b> διαφορετικά.
6. containsSegment(Segment s)	Επιστρέφει <b>true</b> εάν ο κύκλος περικλείει ολόκληρο το ευθύγραμμο τμήμα <b>s</b> , <b>false</b> διαφορετικά.

**Θέμα 5<sup>ο</sup>** (Διαπροσωπείες και υλοποιήσεις τους)

Η διαπροσωπεία **IntPriorityQueue** μοντελοποιεί μία ουρά προτεραιότητας ακεραίων, δηλαδή μία δομή δεδομένων που αποτελείται από ένα σύνολο ακεραίων (επιτρέπονται οι επαναλήψεις) και υποστηρίζει τις παρακάτω μεθόδους:

void insert(int elem)	Εισάγει το στοιχείο-ακέραιο <b>elem</b> στην ουρά προτεραιότητας. Υποθέτει ότι η ουρά προτεραιότητας δεν είναι γεμάτη.
int deleteMax()	Διαγράφει και επιστρέφει το μεγαλύτερο στοιχείο της ουράς.
int size()	Επιστρέφει τον αριθμό των στοιχείων που βρίσκονται στην ουρά προτεραιότητας.
boolean isFull()	Ελέγχει εάν η ουρά προτεραιότητας είναι γεμάτη.

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