

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής**20 Ιουνίου 2017**

- Διάρκεια 2:15 ώρες
- Να απαντηθούν **ΟΛΑ (4)** τα θέματα.
- Καλή επιτυχία.

Θέμα 1^ο

Να υλοποιηθεί η στατική μέθοδος **PowerLargerThan()** η οποία δέχεται ως παράμετρους δύο θετικούς ακέραιους **k** και **n** και υπολογίζει την μικρότερη δύναμη του **k** η οποία είναι μεγαλύτερη του **n**. Να μην χρησιμοποιηθούν μέθοδοι βιβλιοθήκης. Εάν χρησιμοποιήσετε κάποια άλλη μέθοδο (πχ. ύψωση σε δύναμη) πρέπει δώσετε και την υλοποίησή της.

Θέμα 2^ο

Δίνεται η κλάση **Employee** (Υπάλληλος) η οποία χρησιμοποιείται στην μοντελοποίηση ενός εργαζόμενου. Κάθε αντικείμενο της κλάσης **Employee** υλοποιεί τις μεθόδους:

Employee (String name, int salary)	Κατασκευαστής. Θέτει το όνομα και το μισθό του κάθε εργαζόμενου.
String getName()	Επιστρέφει το όνομα.
int getSalary()	Επιστρέφει το μισθό.
String toString()	Εκτυπώνει τον εργαζόμενο (σε μία γραμμή εξόδου)

Να γραφεί η στατική μέθοδος **SalarySummary** η οποία δέχεται ως παραμέτρους ένα διάνυσμα από αντικείμενα τύπου Employee και **εκτυπώνει** i) το μέσο μισθό των εργαζομένων και ii) το μισθό και το όνομα των καλύτερα αμειβόμενων εργαζομένων (οι εργαζόμενοι με τον μεγαλύτερο μισθό, σε περίπτωση που είναι περισσότεροι του ενός, εκτυπώνονται όλοι).

Θέμα 3^ο (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. **getX()** Επιστρέφει την X-συντεταγμένη του σημείου
3. **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. **getP1()** Επιστρέφει το πρώτο σημείο του ευθύγραμμου τμήματος.
3. **getP2()** Επιστρέφει το δεύτερο σημείο του ευθύγραμμου τμήματος.
4. **length()** Επιστρέφει το μήκος του ευθύγραμμου τμήματος
5. **projection(Point p)** Επιστρέφει την προβολή (ένα σημείο) του σημείου p πάνω στο ευθύγραμμο τμήμα. Εάν η προβολή είναι εκτός του ευθύγραμμου τμήματος, τότε επιστρέφει null.

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

Θέμα 4^ο (Διαπροσωπείες και υλοποιήσεις τους)

Δίνεται η κλάση Student (φοιτητής) η οποία χρησιμοποιείται στην μοντελοποίηση ενός εργαστηριακού τμήματος κάποιου μαθήματος. Κάθε αντικείμενο της κλάσης Student υλοποιεί τις μεθόδους:

Student(String name, String id)	Κατασκευαστής. Θέτει το όνομα και τον αριθμό μητρώου κάθε φοιτητή.
String getName()	Επιστρέφει το όνομα του φοιτητή.
String getID()	Επιστρέφει τον αριθμό μητρώου του φοιτητή.
String toString()	Επιστρέφει τον φοιτητή (ως string)

Δίνεται ο (όχι πλήρης) κώδικας για την κλάση **LabGroup** η οποία υλοποιεί το εργαστηριακό τμήμα (με το πολύ 25 φοιτητές) χρησιμοποιώντας ένα μονοδιάστατο διάνυσμα. Η κλάση LabGroup έχει τις παρακάτω μεθόδους:

LabGroup()	Κατασκευαστής. Δημιουργεί ένα εργαστηριακό τμήμα.
void insert(Student s)	Εισάγει τον φοιτητή s στο εργαστηριακό τμήμα.
int size()	Επιστρέφει το μέγεθος του τμήματος.
boolean isFull()	Ελέγχει εάν το τμήμα είναι πλήρες.
Iterator iter()	Επιστρέφει έναν “προσπελαστή” (iterator) προς τους φοιτητές του εργαστηριακού τμήματος. ΝΑ ΥΛΟΠΟΙΗΘΕΙ.

Να υλοποιηθεί η μέθοδος **iter()**. Να συμπληρωθεί ο υπάρχων κώδικας με ότι επιπλέον στοιχεία χρειάζονται. Δίνεται η διαπροσωπεία **Iterator** (στο τέλος της σελίδας).

```

01 public class LabGroup {
02     private static int groupNumber=0;
03     private String groupID;
04     private Student data[];
05     private int size;
06
07     public LabGroup() {
08         groupNumber++;
09         groupID="Group"+groupNumber;
10         data = new Student[25];
11         size=0;
12     }
13
14     public void insert(Student s) {
15         data[size]=s;
16         size++;
17     }
18
19     public int size() {
20         return size;
21     }
22
23     public boolean isFull() {
24         return (size==25);
25     }
26
27 }//class LabGroup

```

```

interface Iterator
{
    boolean hasNext() //Returns true if the iteration has more elements, false otherwise.
    Object next()    //Returns the next element of the Iteration
}

```