Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστήμων
Τομέας Μαθηματικών

Σχεδίαση – Ανάπτυξη Εφαρμογών Πληροφορικής
17 Ιουνίου 2010

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

Ονοματεπώνυμο:	**Α. Μητρώου:**

Θέμα 1ο

Να υλοποιηθεί η στατική μέθοδος **logApprox()** η οποία δέχεται ως παραμέτρους δύο θετικούς ακεραίους ***b*** και ***x*** και «προσεγγίζει» τον λογάριθμό του ***x*** με βάση το ***b***, υπολογίζει και επιστρέφει δηλαδή έναν ακέραιο *y* τέτοιον ώστε να ισχύει ότι $b^y \le x < b^{y+1}$. Να μην χρησιμοποιηθούν μέθοδοι βιβλιοθήκης. Εάν χρησιμοποιήσετε κάποια άλλη μέθοδο (πχ. ύψωση σε δύναμη) **πρέπει δώσετε και τη υλοποίησή** της.

Θέμα 2ο

Να υλοποιηθεί η κλάση **Circle η οποία έχει σκοπό** να μοντελοποιήσει ένα κύκλο. Ο κάθε κύκλος ορίζεται από το κέντρο του (τύπου Point στις δύο διαστάσεις) **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. area()	Επιστρέφει το εμβαδόν του κύκλου.
5. intersectsWith(Circle c)	Επιστρέφει **true** εάν ο κύκλος τέμνει τον κύκλο **c**, **false** διαφορετικά.
6. toString()	**Επιστρέφει τον κύκλο σε εκτυπώσιμη μορφή (ως String)**

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/μετατρεπει την **X-συντεταγμένη** του σημείου
3. setY(int y) Θέτει/μετατρεπει την **Y-συντεταγμένη** του σημείου
4. getX() Επιστρέφει την X-συντεταγμένη του σημείου
5. getY() Επιστρέφει την Y-συντεταγμένη του σημείου
6. distanceFrom(Point p) Επιστρέφει την απόσταση το σημείο από το σημείο που δίνεται ως παράμετρος.
7. toString() Επιστρέφει το σημείο σε εκτυπώσιμη μορφή (ως **string**)

Θέμα 3ο

Να γραφεί η στατική μέθοδος **has2by2PositiveBlock()** η οποία δέχεται **ως παράμετρο ένα** δισδιάστατο διάνυσμα (μεγαλύτερο ή ίσο με 2X2) από ακεραίους και ελέγχει εάν το διάνυσμα περιέχει ως υπο-πίνακα ένα 2X2 πίνακα με *θετικό **άθροισμα στοιχείων***. **Εάν τέτοιος υπο-πίνακας υπάρχει**, η μέθοδος επιστρέφει τις συντεταγμένες του *κάτω-δεξιού* στοιχείου του (ως ένα αντικείμενο τύπου Point – δείτε «**θέμα 2ο**»), αλλιώς επιστρέφει **null**.

Θέμα 4ο (Γραφικά περιβάλλοντα επικοινωνίας)

Ο κώδικας που ακολουθεί υλοποιεί μία παραθυρική εφαρμογή που μετατρέπει δραχμές σε ευρώ και αντίστροφα. **Να γίνουν οι απαραίτητες προσθήκες/τροποποιήσεις** στον κώδικα έτσι ώστε ο μετατροπέας να υποστηρίζει (με δύο επιπλέον πλήκτρα) και την μετατροπή θερμοκρασιών (Κελσίου σε Φαρενάιτ και αντίστροφα). Η μετατροπή μίας θερμοκρασίας από βαθμούς Φαρενάιτ σε βαθμούς Κελσίου δίνεται από την συνάρτηση K(x)=(x-32)/1,8.

```
001 import java.awt.*;
002 import java.awt.event.*;
003 import javax.swing.*;
004 import java.text.*;
005
006 public class EuroConverter extends JFrame {
007    private JTextField inputField;
008    private JTextField outputField;
009    private JLabel errorMessage;
010    private Container cp;
011
012    //Constructor
013    public EuroConverter()
014    {
015       super("EuroConverter");
016       cp=getContentPane();
017       makeConverterGUI();
018
019    }
020
021
022    public static void main(String[] args)
023    {
024     JFrame window=new EuroConverter();
025     window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
026     window.setSize(new Dimension(300,200));
027     window.setVisible(true);
028     window.setResizable(false);
029    }
030
031    public void makeConverterGUI()
032    {
033       JPanel inoutPanel=new JPanel(new GridLayout(2,2));
034       inoutPanel.setPreferredSize(new Dimension(0,60));
035       inoutPanel.setBorder(BorderFactory.createEtchedBorder());
036
037       //GUI elements for data input
038       inoutPanel.add(new JLabel("Input: ", JLabel.RIGHT));
039       inputField=new JTextField("",25);
040       inputField.setHorizontalAlignment(JTextField.RIGHT);
041       inoutPanel.add(inputField);
042
043       //GUI elements for output of results
044       inoutPanel.add(new JLabel("Result: ", JLabel.RIGHT));
045       outputField=new JTextField("",25);
046       outputField.setEditable(false);
047       outputField.setHorizontalAlignment(JTextField.RIGHT);
048       inoutPanel.add(outputField);
049
050       cp.add(inoutPanel,BorderLayout.NORTH);
051
052       JPanel errorPanel=new JPanel();
053       errorMessage=new JLabel("",JLabel.LEFT);
054       errorMessage.setForeground(Color.RED);
055       errorPanel.add(errorMessage);
056       cp.add(errorPanel, BorderLayout.WEST);
057
058       //Setting up panel for conversion buttons
059       JPanel conversionPanel=new JPanel();
060       JButton b=new JButton("Euro To Drachmas");
061       b.addActionListener(new conversionListener());
062       conversionPanel.add(b);
```

```
063
064     b=new JButton("Drachmas To Euro");
065     b.addActionListener(new conversionListener());
066     conversionPanel.add(b);
067
068     cp.add(conversionPanel,BorderLayout.SOUTH);
069   }
070
071   //Internal listener class for conversion buttons
072   class conversionListener implements ActionListener
073   {
074     public void actionPerformed(ActionEvent e)
075     {
076       double numResult=0.0;
077       String strResult="";
078
079       NumberFormat nf = NumberFormat.getNumberInstance();
080       nf.setMaximumFractionDigits(2);
081
082       String operation=e.getActionCommand();
083
084       try
085       {
086         errorMessage.setText("");
087         if (operation.equals("Euro To Drachmas"))
088         {
089           numResult = 340.75 * Double.parseDouble(inputField.getText());
090         }
091         else
092         {
093           numResult = Double.parseDouble(inputField.getText()) / 340.75;
094         }
095
096         outputField.setText(nf.format(numResult));
097       }
098       catch (NumberFormatException ex)
099       {
100         errorMessage.setText("Please insert numeric data!");
101       }
102     }
103   }
104 }
```

Θέμα 5ο

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 επαφής.
`void setName(String newName)`	Θέτει το όνομα του φοιτητή.
`void setID(String newID)`	Θέτει τον αριθμό μητρώου του φοιτητή.
`String getName()`	Επιστρέφει το όνομα του φοιτητή.
`String getID()`	Επιστρέφει τον αριθμό μητρώου του φοιτητή.
`String toString()`	Εκτυπώνει τον φοιτητή (σε μία γραμμή εξόδου)

Δίνεται ο (όχι πλήρης) κώδικάς 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.
`void printStudents()`	Τυπώνει τους φοιτητές στο παράθυρο εξόδου
`int size()`	Επιστρέφει το μέγεθος του τμήματος.
`boolean isFull()`	Ελέγχει εάν το τμήμα είναι πλήρες.
`boolean isInLabGroup(String id)`	Ελέγχει εάν ο φοιτητής ανήκει στο τμήμα.
`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    public void printStudents() {
28       System.out.println("Lab Group:" + groupID);
29       System.out.println();
30       for (int i=0; i<size; i++)
31          System.out.println(data[i]);
32    }
33
34    public boolean isInLabGroup(String id) {
35       for (int i=0; i<size; i++)
36          if (data[i].getID().equals(id))
37             return true;
38       return false;
39    }
40
41 }//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
}
```