

Εργαστηριακή άσκηση 7: Λίστα Φοιτητών

Δημιουργήστε ένα νέο έργο και ονομάστε το **list_of_students**. Αρχικά ορίστε την κλάση **Foithths**, η οποία θα περιέχει τα πεδία:

```
private String onoma, epvnymo;  
private double average;  
private int age;
```

και τις ακόλουθες μεθόδους:

- (1) **public Foithths(String onoma, String epvnymo, double average, int age)**, (ο κατασκευαστής),
- (2) **public String getOnoma()**,
- (3) **public String getEpvnymo()**,
- (4) **public double getAverage()**, και
- (5) **public void print()**
{
 System.out.println("\n Τα στοιχεία του φοιτητή είναι: \n\t Επώνυμο:: \t" + epvnymo +
 "\n\t Όνομα:: \t" + onoma +
 "\n\t Μέσος Όρος:: \t" + average +
 "\n\t Ηλικία:: \t" + age);
}

Στη συνέχεια ορίστε την κλάση **Node**, η οποία θα περιέχει τα πεδία:

```
private Foithths data;  
private Node next;
```

και τις ακόλουθες μεθόδους:

- (1) **public Node(Foithths data)**, (ο κατασκευαστής),
- (2) **public Foithths getData()**,
- (3) **public Node getNext()**,
- (4) **public void setData(Foithths newData)**,
- (5) **public void setNext(Node newNext)**, και
- (6) **public void print()**.

Ακολουθώντας, ορίστε την κλάση **FoithtesList**, η οποία θα περιέχει το πεδίο:

```
private Node head;
```

και τις μεθόδους:

- (1) **public FoithtesList()**, (ο κατασκευαστής),
- (2) **public void insert(Node newNode)**, η οποία θα εισάγει το νέο κόμβο στην κεφαλή της λίστας,
- (3) **public void insertAverage(Node newNode)**, η οποία θα εισάγει το νέο κόμβο στην κατάλληλη θέση μέσα στη λίστα ώστε αυτή να βρίσκεται σε αύξουσα διάταξη ως προς το μέσο όρο,

Μεθοδολογίες και Γλώσσες Προγραμματισμού II

- (4) **public void insertAlphabetical(Node newNode)**, η οποία θα εισάγει το νέο κόμβο στην κατάλληλη θέση μέσα στη λίστα ώστε αυτή να βρίσκεται αλφαβητικά ταξινομημένη ως προς το επώνυμο, και
- (5) **public void print()**.

Τέλος θα ορίσετε την κλάση **TestSortingAlgorithms** όπως βλέπετε παρακάτω:

```
public class TestList
{
    public static void main(String args[])
    {
        Foithths foit1 = new Foithths("Νίκος", "Νικολάου", 7.34, 20);
        Foithths foit2 = new Foithths("Δημήτρης", "Δημητρίου", 5.99, 21);
        Foithths foit3 = new Foithths("Γιώργος", "Γεωργίου", 7.79, 19);
        Foithths foit4 = new Foithths("Κώστας", "Κωνσταντίνου", 8.75, 22);
        Foithths foit5 = new Foithths("Ανδρέας", "Ανδρέου", 6.35, 21);
        Foithths foit6 = new Foithths("Κώστας", "Γεωργίου", 5.75, 20);

        Node node1 = new Node(foit1);
        Node node2 = new Node(foit2);
        Node node3 = new Node(foit3);
        Node node4 = new Node(foit4);
        Node node5 = new Node(foit5);
        Node node6 = new Node(foit6);
        Node node7 = new Node(foit1);
        Node node8 = new Node(foit2);
        Node node9 = new Node(foit3);
        Node node10 = new Node(foit4);
        Node node11 = new Node(foit5);
        Node node12 = new Node(foit6);
        Node node13 = new Node(foit1);
        Node node14 = new Node(foit2);
        Node node15 = new Node(foit3);
        Node node16 = new Node(foit4);
        Node node17 = new Node(foit5);
        Node node18 = new Node(foit6);

        FoithtesList samos2004 = new FoithtesList();
        FoithtesList samos2004_2 = new FoithtesList();
        FoithtesList samos2004_3 = new FoithtesList();

        samos2004.insert(node1);
        samos2004.insert(node2);
        samos2004.insert(node3);
        samos2004.insert(node4);
        samos2004.insert(node5);
        samos2004.insert(node6);
        samos2004_2.insertAverage(node7);
        samos2004_2.insertAverage(node8);
        samos2004_2.insertAverage(node9);
        samos2004_2.insertAverage(node10);
        samos2004_2.insertAverage(node11);
        samos2004_2.insertAverage(node12);
        samos2004_3.insertAlphabetical(node13);
        samos2004_3.insertAlphabetical(node14);
        samos2004_3.insertAlphabetical(node15);
        samos2004_3.insertAlphabetical(node16);
        samos2004_3.insertAlphabetical(node17);
    }
}
```

Μεθοδολογίες και Γλώσσες Προγραμματισμού II

```
samos2004_3.insertAlphabetical(node18);  
  
System.out.println("\n Η λίστα όταν η εισαγωγή κάθε νέου κόμβου γίνεται στην κεφαλή είναι:  
");  
samos2004.print();  
System.out.println("\n Η λίστα όταν η εισαγωγή κάθε νέου κόμβου γίνεται με αύξοντα μέσο όρο  
είναι: ");  
samos2004_2.print();  
System.out.println("\n Η λίστα όταν η εισαγωγή κάθε νέου κόμβου γίνεται αλφαβητικά είναι: ");  
samos2004_3.print();  
} // main  
  
} // TestList
```

Στη μέθοδο **insertAlphabetical(Node newNode)** μπορείτε να χρησιμοποιήσετε τη μέθοδο **compareTo(string2)** της κλάσης **String**, ως εξής:
string1.compareTo(string2)

Η μέθοδος αυτή δέχεται ένα όρισμα (το **string2**) και επιστρέφει αρνητική τιμή (-1) αν το **string1** είναι μικρότερο λεξικογραφικά από το αντικείμενο **string2**, 0 αν είναι ίσο και θετική τιμή (1) αν το **string1** είναι μεγαλύτερο λεξικογραφικά από το αντικείμενο **string2**.

Να επιδειχθεί στον υπεύθυνο του εργαστηρίου την 7^η εβδομάδα.