

Φροντιστηριακή άσκηση #2-A

Θέμα: Εξοικείωση με την χρήση μεταβλητών τύπου αναφοράς για δημιουργία συνδεδεμένων δομών. (Μία απλή συνδεδεμένη λίστα χρησιμοποιείται στην φροντιστηριακή άσκηση #2-B)

Η φροντιστηριακή αυτή άσκηση αποσκοπεί στην εξοικείωση με τη χρήση των μεταβλητών τύπου αναφοράς για την δημιουργία και τον χειρισμό μιας συνδεδεμένης δομής αποθήκευσης δεδομένων, καθώς και των εντολών επανάληψης. Θα δημιουργήσουμε το έργο **intList** το οποίο περιέχει τις κλάσεις **Cell** και **IntList**.

Η κλάση **IntList** υλοποιεί μια διατεταγμένη λίστα από ακέραιους. Λειτουργίες που παρέχονται από τη λίστα περιλαμβάνουν την προσθήκη αριθμών στο τέλος της λίστας, τη διαγραφή αριθμών από την αρχή της λίστας, την προσθήκη(διαγραφή) αριθμού σε(από) συγκεκριμένη θέση στη λίστα, την εκτύπωση των στοιχείων της λίστας. Για λόγους συμβατότητας με την αρίθμηση των στοιχείων των διανυσμάτων, το πρώτο στοιχείο της λίστας έχει αύξοντα αριθμό μηδέν (0).

Αντικείμενα της κλάσης **Cell** (τα κελιά) αποτελούν τα δομικά υλικά για την κατασκευή την λίστας **IntList**. Κάθε αντικείμενο τύπου **Cell** περιέχει δύο πεδία: ένα για να αποθηκεύσει τον ακέραιο αριθμό (τα δεδομένα) και ένα για να υποδείξει το επόμενο στοιχείο (τύπου **Cell**) στη λίστα.

1. Δημιουργήστε ένα νέο έργο με την ονομασία **intList**. Για οικονομία χρόνου, στην εργαστηριακή αυτή άσκηση δεν θα εισάγουμε σχόλια στον κώδικα που δημιουργούμε.

2. Δημιουργήστε την κλάση **Cell**. Η κλάση **Cell** αποτελείται από τα πεδία

```
private int data
private Cell next
```

και παρέχει τις παρακάτω μεθόδους:

- **Cell()** Κατασκευαστής. Αποθηκεύει τον αριθμό μηδεν(0) και δεν έχει επόμενο κελί (**next=null**).
- **Cell(int value)** Κατασκευαστής. Αποθηκεύει τον αριθμό value και δεν έχει επόμενο κελί (**next=null**).
- **void setData(int newData)** Αλλάζει την τιμή των δεδομένων ενός κελιού
- **void setNext(Cell newNext)** Αλλάζει την τιμή του επόμενου κελιού
- **int getData()** Επιστρέφει την τιμή των δεδομένων ενός κελιού
- **Cell getNext()** Επιστρέφει το επόμενο κελί

3. Υλοποιήστε την κλάση **Cell** και βεβαιωθείτε ότι καταλαβαίνετε τη λειτουργία της. Ιδιαίτερη προσοχή να δοθεί στο ρόλο του πεδίου next και στη χρήση του για τη δημιουργία συνδεδεμένων δομών δεδομένων. Μεταφράστε την κλάση, και δημιουργήστε τουλάχιστον 3 αντικείμενα. Κατόπιν κάνοντας χρήση της μεθόδου **setNext()** συνδέστε τα σε μία λίστα.

4. Δημιουργήστε την κλάση **IntList**. Η κλάση **IntList** αποτελείται από τα πεδία

```
private Cell head; //αναφορά προς το πρώτο στοιχείο της λίστας
private Cell tail; //αναφορά προς το τελευταίο στοιχείο της λίστας
private int noOfElems; //ο αριθμός των στοιχείων της λίστας
```

5. Η κλάση **intList** παρέχει τις παρακάτω μεθόδους:

- **IntList()** Κατασκευάζει μία άδεια λίστα.
- **int size()** Επιστρέφει το πλήθος των στοιχείων της λίστας
- **boolean isEmpty()** Ελέγχει το εάν η λίστα περιέχει στοιχεία
- **void insertBack(int y)** Εισάγει ένα νέο στοιχείο στο τέλος της λίστας
- **int deleteFront()** Διαγράφει ένα στοιχείο από την αρχή της λίστας
- **void print()** Τυπώνει τα στοιχεία της λίστας.
- **int locate(int key)** Επιστρέφει τη θέση του προς αναζήτηση στοιχείου στη λίστα εάν αυτό υπάρχει, διαφορετικά, την τιμή (-1).
- **int dataAt(int location)** Επιστρέφει τα δεδομένα στη συγκεκριμένη θέση της λίστας. Υποθέτει ότι η θέση υπάρχει.
- **int deleteFrom(int location)** Διαγράφει τα δεδομένα από τη συγκεκριμένη θέση της λίστας. Υποθέτει ότι η θέση υπάρχει.

- `void insertAt(int pos, int y)` Εισάγει δεδομένα έτσι ώστε, μετά την εισαγωγή τους, να βρίσκονται στη συγκεκριμένη θέση της λίστας. Υποθέτει ότι η θέση υπάρχει μετά την εισαγωγή.
- `void update(int pos, int newValue)` Ενημερώνει την τιμή των δεδομένων που βρίσκονται στη συγκεκριμένη θέση. Υποθέτει ότι η θέση υπάρχει.

6. Ο κώδικας για τον κατασκευαστή και την μέθοδο `insertBack()` έχει ως εξής:

```
public IntList()
{
    head=null;
    tail=null;
    noOfElems=0;
}

public void insertBack(int y)
{
    Cell cell=new Cell(y);
    if ( noOfElems==0){
        head=cell;
        tail=cell;
    }
    else{
        tail.setNext(cell);
        tail=cell;
    }
    noOfElems++;
} //insetBack
```

Εισάγετε και μεταφράστε τον κώδικα. Μελετήστε και βεβαιωθείτε ότι τον κατανοείτε. Παρατηρήστε ότι ιδιαίτερη σημασία δίδεται στην περίπτωση εισαγωγής δεδομένων σε άδεια λίστα

7. Ο κώδικας για την μεθοδο `print()` έχει ως εξής:

```
public void print()
{
    System.out.print("(");

    Cell item=head;
    while (item!=null) {
        System.out.print(item.getData());
        if (item.getNext()!=null)
            System.out.print(", ");
        item=item.getNext();
    }
    System.out.println(")");
} //print
```

Εισάγετε και μεταφράστε τον κώδικα. Μελετήστε και βεβαιωθείτε ότι τον κατανοείτε. Πειραματιστείτε δημιουργώντας μια λίστα και κατόπιν τυπώνοντας την με τη μέθοδο `print()`.

8. Υλοποιήστε τις υπόλοιπες μεθόδους της κλάσης `intList`. Να δώσετε ιδιαίτερη προσοχή στις οριακές περιπτώσεις, δηλ, όταν εισάγουμε σε άδεια λίστα, όταν διαγράφουμε το τελευταίο στοιχείο της, όταν επιχειρούμε να προσπελάσουμε στοιχεία στην αρχή ή το τέλος της λίστας. Μετά από την υλοποίηση κάθε μεθόδου να πειραματίζεστε δημιουργώντας μια νέα λίστα και εκτελώντας τη μέθοδο. Να ελέγχετε πάντα τις οριακές περιπτώσεις.
9. Να εμπλουτιστεί ο κώδικας των μεθόδων του έργου `intList` με σχόλια.