

ΑΣΚΗΣΗ 2

ΣΤΟΙΒΕΣ (Stacks) και ΟΥΡΕΣ (Queues)

Υλοποίηση με πίνακα

(Βλέπε http://www.iee.ihu.gr/~demos/teaching_GR.html)

Άσκηση 2.1

Δίνεται η παρακάτω διασύνδεση **Stack interface** που περιγράφει τη λειτουργία μιας στοίβας. Δώστε μία υλοποίηση της **Stack** με τη βοήθεια της **ArrayStack class** η οποία χρησιμοποιεί πίνακα για την υλοποίηση της στοίβας.

Αρχείο Stack.java

```
public interface Stack
{
    public int size();
    // Returns the size of the Stack

    public boolean isEmpty();
    // Returns true if the Stack is empty

    public boolean isFull();
    // Returns true if the Stack is full

    public Object top() throws StackEmptyException;
    // Returns the top item of the Stack

    public void push(Object item) throws StackFullException;
    // Adds a new item into the Stack

    public Object pop() throws StackEmptyException;
    // Removes the top item of the Stack
}
```

Αρχείο StackEmptyException.java

```
public class StackEmptyException extends RuntimeException
{
    public StackEmptyException (String err)
    {
        super(err);
    }
}
```

```
public class StackFullException extends RuntimeException
{
    public StackFullException (String err)
    {
        super(err);
    }
}
```

Άσκηση 2.2

Δημιουργήστε μια στοίβα από φοιτητές -χρησιμοποιήστε την Student class από το εργαστήριο 2. Εκτελέστε τις βασικές λειτουργίες της στοίβας.

Άσκηση 2.3

Να γραφεί πρόγραμμα Java για τον έλεγχο της σωστής χρήσης των παρενθέσεων (και) σε μία αριθμητική παράσταση χρησιμοποιώντας τη δομή δεδομένων στοίβα. Το πρόγραμμα να επιστρέφει μήνυμα εάν η αριθμητική παράσταση είναι σωστή ή εάν έχει λάθος, να αναφέρει το σημείο που βρήκε το λάθος.

Παραδείγματα

(1+3)*3*(2+2)
Result: No error

1+3)*3*(2+2)
Result: error at position 4 !!

Άσκηση 2.4

Επεκτείνετε το πρόγραμμα της άσκησης 2.4 ώστε να γίνεται έλεγχος για τη σωστή χρήση 3 μορφών παρενθέσεων: (και), { και }, [και].

Άσκηση 2.5

Με βάση την παρακάτω διασύνδεση **Queue interface** που περιγράφει τη λειτουργία μιας ουράς, να υλοποιήσετε την **ArrayQueue class** με χρήση πίνακα. Στον κώδικα που θα δώσετε φροντίστε να αντιμετωπίζεται σωστά η περίπτωση της **εικονικής υπερχείλισης**.

Αρχείο Queue.java

```
public interface Queue
{
    public int size();
    // Returns the size of the queue

    public boolean isEmpty();
    // Returns true if the queue is empty

    public boolean isFull();
    // Returns true if the queue is full

    public Object front() throws QueueEmptyException;
    // Returns the first element of the queue

    public void enqueue(Object item) throws QueueFullException;
    // Adds an element at the end of the queue

    public Object dequeue() throws QueueEmptyException; }
    // Removes and returns the first element of the queue
```

Αρχείο QueueEmptyException.java

```
public class QueueEmptyException extends RuntimeException
{
    public QueueEmptyException(String err)
    {
        super(err);
    }
}
```

Αρχείο QueueFullException.java

```
public class QueueFullException extends RuntimeException
{
    public QueueFullException(String err)
    {
        super(err);
    }
}
```

Άσκηση 2.6

Δημιουργήστε μια ουρά από φοιτητές -χρησιμοποιήστε την Student class από προηγούμενη άσκηση. Ειτελέστε τις βασικές λειτουργίες της ουράς.

Άσκηση 2.7

Να γραφεί πρόγραμμα Java για την εξυπηρέτηση αυτοκινήτων σε διόδια με την χρήση **ουράς**, η οποία θα υλοποιείται με **πίνακα**. Πιο συγκεκριμένα θα εμφανίζεται το παρακάτω μενού:

MENΟΥ

1. Άφιξη αυτοκινήτου
2. Αναχώρηση αυτοκινήτου
3. Κατάσταση ουράς
4. Έξοδος

Επιλογή 1: Θα πληκτρολογούνται τα στοιχεία του αυτοκινήτου π.χ. ο αριθμός αυτοκινήτου και θα τοποθετείται στο τέλος της ουράς.

Επιλογή 2: Το αυτοκίνητο που βρίσκεται πρώτο στην ουρά θα διαγράφεται μαζί με ένα ανάλογο μήνυμα επιβεβαίωσης.

Επιλογή 3: Θα εμφανίζονται με τη σειρά οι αριθμοί των αυτοκινήτων που παραμένουν στην ουρά για να εξυπηρετηθούν.

Επιλογή 4: Το πρόγραμμα θα τερματίζεται.

Πρέπει

- Να δημιουργήσετε την **Car** class –αφαιρετική αναπαράσταση ενός αυτοκινήτου
- Να χρησιμοποιήσετε την **ArrayQueue** class από την άσκηση 3.1
- Να δημιουργήσετε την **CarQueueManagement** class που θα διαχειρίζεται το πρόγραμμά σας (*includes main method*)